A Content-Based Recommendation System for Leisure Activities

Marcelo Darío Rodas Brítez

Advisor: Prof. Maurizio Marchese

Department of Information Engineering and Computer Science
University of Trento

This dissertation is submitted for the degree of
Doctor of Philosophy

October 2019

To God, the one that invited me to follow this amazing challenge.

Acknowledgements

Thank all my friends that give me support for my life in Trento. Especially my family that, from the long distance, gave me the force to go through this big personal and professional challenge.

I want to thank my advisor, Prof. Maurizio Marchese, for the opportunity to work and learn during these last three years of the Ph.D. Programme.

I am thankful to the European Union that supported and funded the project which this research work has been a part of. This work has received funding from the European Union's Horizon 2020 Research and Innovation Programme - Societal Challenge 1 (DG CONNECT/H) under grant agreement n. 643644 "ACANTO - A CyberphysicAl social NeTwOrk using robot friends".

Abstract

People's selection of leisure activities is a complex choice because of implicit human factors and explicit environmental factors. Satisfactory participation in leisure activities is an important task since keeping a regular active lifestyle can help to maintain and improve the wellbeing of people. Technology could help in selecting the most appropriate activities by designing and implementing activities, collecting people profiles and their preferences relations. In fact, recommendation systems, have been successfully used in the last years in similar tasks with different types of recommendation systems. This thesis aims at the design, implementation, and evaluation of recommendation systems that could help us to better understand the complex choice of selecting leisure activities. In this work, we first define an evaluation framework for different recommendations systems. Then we compare their performances using different evaluation metrics. Thus, we explore and try to better understand the user's preferences over leisure activities. After, having a comprehensive analysis of modelling recommended items and leisure activities, we also design and implement a content-based leisure activity recommendation system to make use of a taxonomy of activities. Moreover, in the course of our research, we have collected and evaluated two datasets obtained one from the Meetup social network and the other from crowd-workers and made them available as open data sources for further evaluation in the recommendation system research community.

Keywords: Leisure Activity Analysis, Recommendation System, Activity Recommendation System, System Modeling, Clustering System

Table of contents

List of figures

List of tables

Chapter 1

Introduction

The beginning is the most important part of the work.
Plato

Recommending an activity to a person is not an easy task because of several behavioural factors like external motivations (monetary, power, prestige, pleasure, fear of punishment, etc.) or internal motivations (enjoyment, learning, etc.) [17]. From a social and human perspective, we could argue that the importance of this type of recommendation is based on the need for improvement or maintenance of people's health and wellbeing. From a technological perspective, we could argue that it is a big challenge to design a feasible behavioural model, and also to effectively help people in the selection of the most appropriate and relevant activities to do.

Also, health and wellbeing studies [89, 34, 45] highlight the importance of engaging in favourite activities, especially in later life. However, proposing suitable leisure activities to an individual is mainly done ad hoc, typically from a medical perspective by a professional. It results in pushing the user towards mainly physical activities while neglecting other aspects of wellbeing, and user preferences towards such activities. This approach can lead to user rejection of the proposed activities.

Considering that the improvement of the user's wellbeing is important, we choose to analyze leisure activities. Also, in part of the analysis of some chapters, we focus on older adults population, which are a good target for recommendations of leisure activities because of the declining wellbeing that this population deal with. The problem context is shown in the Figure 1.1, shows both sides of the problem, considering the real-life interactions of peoples (User) with Activities (Items), and the technological perspective that we are considering to approach.

The technological perspective, shown in Figure 1.1, includes social network sites (SNS) where users could interact as managers of the information. The social interactions that happen in such SNS could be assessed by recommendation systems that provide useful and contextual information to the users, keeping their attention and interest, while satisfying the conditions (preferences and needs) and the physical or cognitive limitations.

Social networks sites and other technological means of communications can help people in the selection of activities and more importantly in socializing these activities with their community of friends, family, doctors, and caregivers. These technology has been implemented and tested to facilitate socialization and sharing of information [9]. For example, one of the chapters takes as a dataset example from a social network focus on the organization of meetups, the Meetup social network[1].

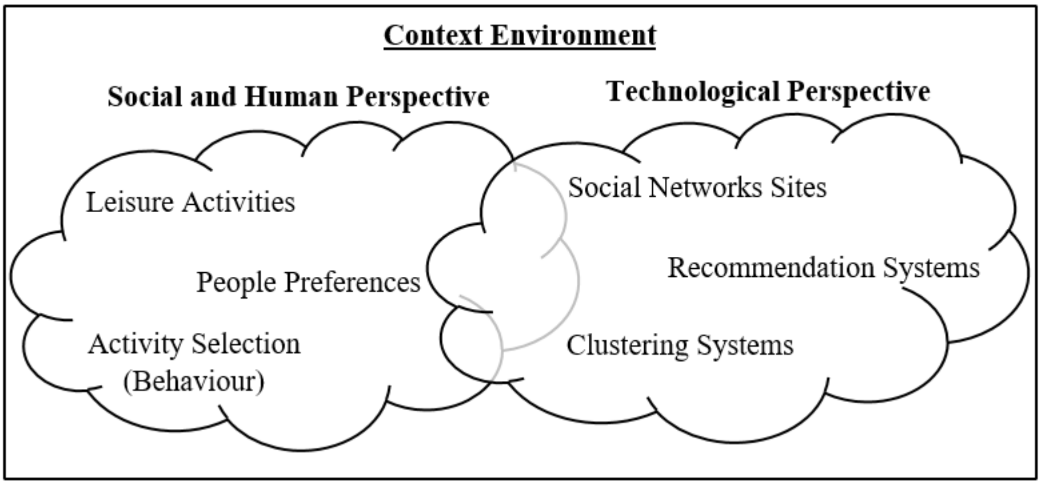


Fig. 1.1 Problem Context.

Within the research context defined above, this thesis focuses on the analysis of leisure activities and recommendation systems, by analyzing, designing and implementing a leisure activity recommendation system. For doing so, we first present the technological background by providing a general description of recommendation systems and the possible evaluation metrics. Base on this background, we design and build an evaluation framework for comparing different recommendation systems approaches. Then, a novel content-based recommendation system using existing and novel activities models will be presented. Finally, we will present and analyze in detail three specific studies relevant to the recommendation of leisure activities and clustering of users.

[1] www.meetup.com

1.1 The Niche of the Problem

The context of our problem is **thus** within recommending **leisure activities** to people, **typically** technologically implemented in recommendation systems embedded in social network sites. Our problem is positioned between the social problem of leisure activity selection and the technological problem of using a recommendation system for recommending relevant items to users.

The social aspect of the problem is understanding the behavioural perspective of people into selecting activities. The technological perspective opens to the possibility of modeling the environment (people, activities, and preferences over activities), together with the possibility of implementation and experimentation of recommendations systems.

In order to analyze, understand, design and implement an information system to support this problem, we pose the following research questions (RQ):

RQ1. Can a feasible data model be developed for a Leisure Activities Recommendation System that represents users, leisure activities, and users' preferences?

RQ2. What are the most appropriate metrics to compare recommendations of leisure activities?

RQ3. What are the most performing recommendations system approaches to meet end users preferences?

1.2 The Proposal

The objective of this research is to analyze technological support alternatives for a sustainable active lifestyle for users. We propose to support this overall objective by recommending to users the participation to social groups with common interests, and thus provide the base for recommending appropriate leisure activities. We expect that the use of such technological support will provide improvements of users' overall wellbeing, as a consequence of users' continuous active involvement into activities.

We propose to analyze the problem from two approaches. Firstly, using a recommendation system of leisure activities, starting from the premise that preferences ratings are likely to determine if the user will actually engage in an activity. Secondly, using clustering algorithms and neighbourhood approaches (like collaborative filtering recommendation), which are based on the idea that the community could help to understand the individual preferences and preference tendencies. These approaches depends on the modelling of the elements of the system (users, activities and preferences).

In the particular problem context of this thesis and considering the technological elements involved, we will be focusing on three main research areas:

- Leisure Activity Modelling.
- Clustering Systems.
- Recommendation Systems.

It is important to understand that any recommendation system is basically composed by the model (users, items, preferences) and the algorithm. In our case, the model corresponds to the Leisure Activity Modelling area and the algorithm corresponds to the Clustering Systems and the Recommendation Systems areas. In the following subsections, we briefly present the three main research areas and our specific approach to them.

1.2.1 Leisure Activities Modelling

We define leisure activity as a voluntary action done by one or many people for a period of time, typically during their free time. It is intrinsically rewarding for the person (e.g. fun) and it is a goal in itself. To work on clustering leisure activities we first need to find good descriptors of leisure activities. **Therefore we focused in the first part of our work** on understanding the leisure activities by analyzing the taxonomies used in the literature, and design an appropriate model to represent activities in clustering and recommendation algorithms. The definition of leisure activity involves many domain factors like energy expenditure, context (sleeping, working, leisure), the intensity, exercises, physical control, etc [15, 56].

We also analyze a dataset of the Meetup social network site because it is oriented to build relationships to support face to face meet-ups of the users. By studying the characteristics of such systems, we can analyze the way users describe their preferences and how activities are implemented and described.

Researchers of this area define limited domains (from 1 to 13 domains, with mean 3,8) showing the diversity of categorical analysis of activities [2]. Also, there are studies about the intrinsic or extrinsic motivations of the users for doing leisure activities [17]. Our model of leisure activities takes into consideration these domains to go together with current research and to define a more comprehensive design of an activity model.

Additionally, we focus on leisure activities because they are considered important for the wellbeing of people [48] especially for older adults [23, 2].

1.2.2 Clustering Systems

Clustering entities by features is a well-known problem in computer science [11]. Many algorithms have been proposed to perform this operation. The usefulness of these systems is that allows us to understand the groups of relationships between entities, having a deeper understanding of how entities are related. We focus on defining some clustering algorithms so we can understand the relationships between users and activities using their descriptors (e.g. tags, dimensions et al.).

It is important to understand that different clustering algorithms give different results because they analyze the data space in different ways, providing advantages and disadvantages. This is the reason why we find important to implement clustering algorithms in an evaluation framework. The evaluation of clustering algorithms will help us to understand the information used to describe the users and the activities, and eventually, could help understand and compare other possible models of our recommendation system.

1.2.3 Recommendation Systems

An important corpus of research has been dedicated to understand and improve recommendation algorithms over the general population of users [99, 19, 3]. If we consider the specific population of older adults, we found research that is not focused on general leisure activities, rather, they focus on other specifics aspect like alimentation recommendations [10, 21], which is also fundamental for physical development, or physical activity recommendation [21, 65, 78, 42, 83].

Considering implementations of a context-based recommendation system of leisure activities, we only found one proposal that tilts to treat the activity as an event (with time and place) and with a generic and static classification of the activity [6]. This example implements a hybrid recommendation system in which the activity was modeled using a combination of patterns observed across the user's demographic population and individual behavior pattern, where the activities were classified in 5 modes: Eating, Shopping, Seeing, Doing, Reading. The problem with this approach is that the activity classification is fixed and the classification is a very high-level abstraction.

This thesis searches a more comprehensive analysis of leisure activities into recommendations systems giving a novel description of the items we are recommending, developing a particular content-based recommendation algorithm based on a clustering model of activities using dimensions of activities.

1.2.4 Contributions

The main contributions (MC) of this thesis, in the aforementioned research areas, are described in relation with the research areas in Table 1.1, with an extended explanation in the following list.

- **MC1 - Evaluation Framework**: We **designed, developed and tested** a Framework for Evaluation of Recommendations (**FER**) to provide pattern analysis of clustering and recommendation algorithms (Clustering Systems and Recommendation Systems areas) in order to compare their results in a systematic way, facilitating developers in the selection of the most appropriate clustering algorithms that best fit their specific requirements and use cases. The proposed framework is based on Java patterns and is extensible: new algorithms, statistics and quality metrics can be easily added. As a first approach, three different cluster algorithms were implemented and initially analyzed: K-means [33], Fuzzy K-means [5], and Affinity Propagation [25]. Also, five types of recommendation algorithms were implemented: item-based collaborative filtering, user-based collaborative filtering, SVD collaborative filtering, content-based filtering, and hybrid-based collaborative filtering. The impact of this contribution to the RQs are corresponding to the feasibility of designing and implementing leisure activity models for recommendations systems.

- **MC2 - Clustering Analysis**: We used the **developed** evaluation framework (FER) for the analysis of clustering algorithms using some data on user's groups from an online social network (Clustering Systems area). We performed older adult group clustering based on affinity to create social groups. The evaluation is based on existing groups in a Meetup dataset. A priori, evaluation of new groups created by different clustering algorithms can then lead social researchers to analyze the relations and distribution of data generated by the social interactions in other datasets. The clustering algorithms are evaluated with both internal and external assessment criteria and using tags as the descriptors of user preferences and as descriptors of the groups. This clustering analysis impact the RQs in the feasibility of using clustering algorithms for recommendation systems for obtaining a better profile of users (groups) or better understanding of activities (activities' groups).

- **MC3 - Leisure Activity Recommendation**: We **developed** a novel Leisure Activity Recommendation System (LAR System) using a model to describe user preferences for leisure activities with predefined dimensions (Leisure Activities, Clustering Systems and Recommendation Systems areas). We evaluate the model showing that a dimension-based leisure activity model outperforms a tag-based model using statistical analysis. Our leisure activity recommendation implementation uses a more uniform representation of the activities, which could simplify the effort from users to express their preferences. This is the major contribution of the thesis, considering that was build using the previous two contributions.

Table 1.1 Degree of contributions in the research areas of the problem space.

Contributions	**Research Areas**		
	Leisure Activities Modelling	**Clustering System**	**Recommendation System**
MC1	low	low	low
MC2	low	high	medium
MC3	high	medium	high

1.3 ACANTO Project

The present research has been developed in the context of the European Project ACANTO, within the Horizon 2020 Framework [1]. The project started at the beginning of 2015 and is coordinated by the University of Trento. The main objective of ACANTO is to increase the number of older adults who engage in regular and sustained physical activity, targeting older adults with mobility problems.

The main reason this project is described here, is that part of the inspiration and studies done in this thesis are linked to the ideas proposed and developed in the ACANTO project.

The main components of the ACANTO project are a walker device (FriWalk), a tablet (FriTab) and a cyber-physical social network described as follows:

- The FriWalk is a device with sensors to help with the movement of older people and to obtain real-time information.
- The FriTab is a tablet and acts as the main interface of interaction with the users and the social network.
- The cyber-physical social network is the technological framework where the social network will be implemented, integrating the components of the Cyber Physiscal Network (CPN).

The social network will be an interface for input and output information related to the recommendation system and the expected social interaction. User profiles, health profiles and environmental data will be available for processing the activity recommendations.

One of the key ideas of ACANTO is to learn as much as possible about the user of the FriWalk/FriTab without the necessity to enter actively this information by the user since we want to ease the burden for our target group. In other words, this means continuous observation and perception of the user's state. Some of the observations will be relevant only in the time of being measured, some will be meaningful by aggregation over a longer period, some of them indicate physiological conditions with medical relevance (e.g. with respect to therapeutic goals) while others address the motivational level or mood of the person.

In any case, the lever to gather all the information is mainly sensors. The sensors are deployed on the FriWalk/FriTab or alternatively also as wearables on the user. The latter option is considered very carefully since the focus was to acquire data in an utmost non-obtrusive way whenever possible.

Furthermore, the sensors could be use to perform relative localization, i.e. with respect to other FriWalk units by the introduction of a novel concept of collaborative localization that reflects an aim of ACANTO in a very natural manner: to fostering group activities and social contacts among older adults.

This thesis contributes to the development of this project mainly in tree parts:

- The requirement analysis of leisure activities and user's preferences.
- The development of an leisure activity model.
- The analysis of clusters of users in social networks.

Finally, the methodology of ACANTO developed an ambitious combination of a careful recognition of user needs and market opportunities, high-quality research and technology integration into a fully functional prototype.

1.4 Methodology and Thesis Structure

For the general methodology of our research, we followed a software engineering approach as shown in Figure 1.2. This diagram shows the following level of analysis: analysis of the problem (requirements), modelling a solution (design), developing software for the purpose (implementation), and evaluating the proposal (testing). This general approach is entangled with the study of the described activity-users problem shown in Figure 1.1, mainly relying

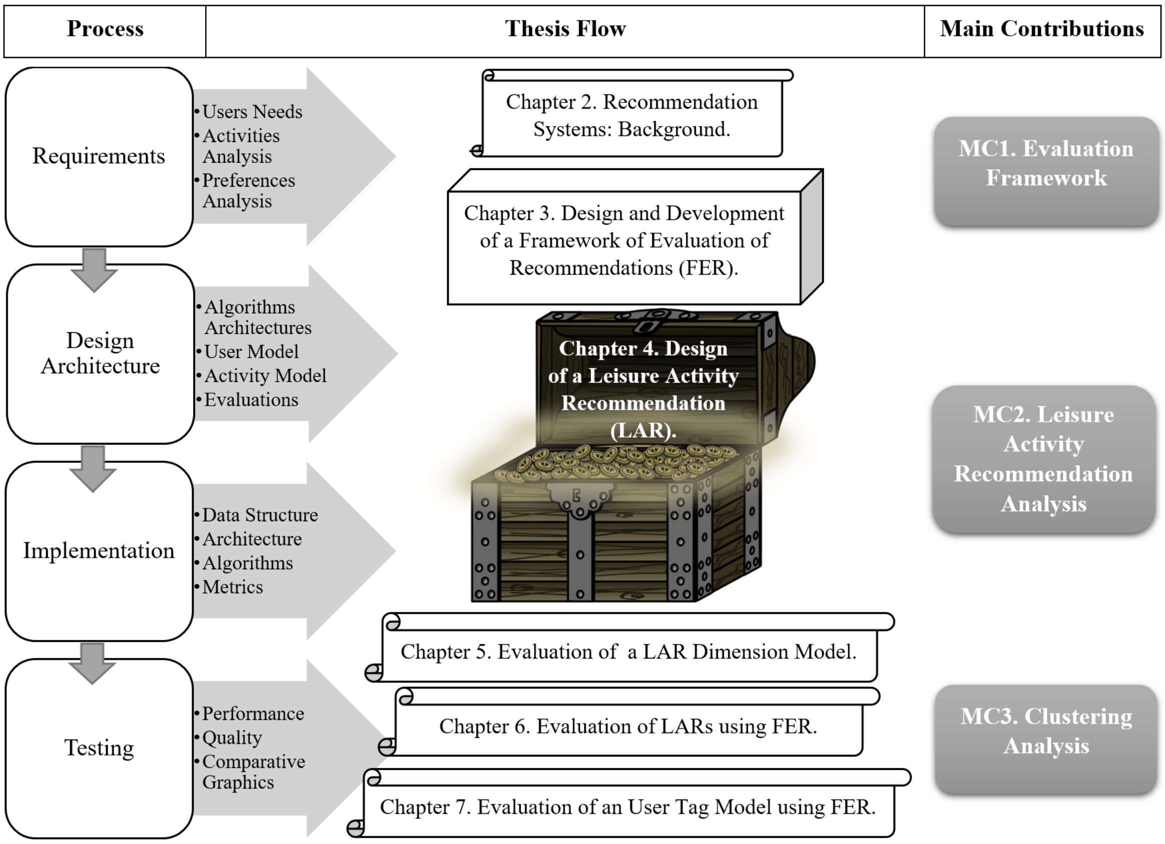

Process
Thesis Flow
Main Contributions
Requirements
•Users Needs
•Activities Analysis
•Preferences Analysis
Chapter 2. Recommendation Systems: Background.
Chapter 3. Design and Development of a Framework of Evaluation of Recommendations (FER).
MC1. Evaluation Framework
Design Architecture
•Algorithms Architectures
•User Model
•Activity Model
•Evaluations
Chapter 4. Design of a Leisure Activity Recommendation (LAR).
MC2. Leisure Activity Recommendation Analysis
Implementation
•Data Structure
•Architecture
•Algorithms
•Metrics
Chapter 5. Evaluation of a LAR Dimension Model.
Testing
•Performance
•Quality
•Comparative Graphics
Chapter 6. Evaluation of LARs using FER.
Chapter 7. Evaluation of an User Tag Model using FER.
MC3. Clustering Analysis

Fig. 1.2 The thesis development process and its main elements

on evaluations of recommendations, user studies, and surveys. Also, the main contributions in Figure 1.2 are leveled relative to the process and flow of this thesis.

The chapters of the thesis follow different methodologies since they analyze different aspects of the leisure activities recommendation systems. Chapter 2 methodology is oriented to understand the requirement analysis for leisure activity recommendations and the feasibility for implementing it in recommendations systems, by using existing literature and technical reports. Chapter 3 methodology is oriented to organize the requirements and technological opportunities into a common workplace for implementing and testing possible recommendations alternatives. Chapter 4 methodology is oriented to the design and implementation of leisure activity recommendation, including the analysis of state of the art of recommendation systems that work with leisure activities. Chapters 5, 6, and 7 are focus on to the evaluation of the leisure activity recommendation, so the methodologies are related to the definition of metrics that allows comparison of different aspects of the recommendation problem (dimension model, performance, and clustering).

The work presented here is based on research publications conducted during the years of doctoral studies, and on technical reports (deliverables) and development done as part of the ACANTO project [1]. For clarity, we will include the citations to these publications in the following description of the structure of the thesis.

Chapter 2.Recommendation Systems: Background.

This chapter provides an in-depth analysis of the state of the art of modern Recommendation Systems. In the literature, there are numerous studies on recommendation systems, focused on specific domains. We are interested in recommendations on groups and physical and social activities. Additionally, we have performed some analysis on commercial social network sites where recommendation systems are used or could be used.

Most of the analysis of the state of the art of recommendation systems have been part of the ACANTO project, specifically in the following public deliverables of the Work Package 4 (WP4), Conception of Social Activities:

Ramos, I., Mediavilla, C., Marchese, M., and Rodas, M. (2016). Deliverable 4.5. User communities creations based on user's profile matching (static profile): social network creation and evolution in older adults communities. ACANTO Project Deliverable, ATOS and University of Trento. [75]

Marchese, M., Rodas Britez, M. D., Ramos, I., and Brauchoff, I. (2017). Deliverable 4.2. User Profile Repository (Final). ACANTO Project Deliverable, University of Trento and ATOS. [51]

Ramos, I., Brauchoff, I., and Marchese, M. (2017). Deliverable 4.4. Social Activity Repository (Final). ACANTO Project Deliverable, ATOS and University of Trento. [74]

Marchese, M., Rodas Britez, M. D., Ramos, I., and Brauchoff, I. (2017). D4.6. User communities' creations based on user's profile matching (dynamic and adaptive profile). ACANTO Project Deliverable, University of Trento and ATOS. [50]

Chapter 3. Design and Development of a Framework of Evaluation of Recommendations (FER).

This chapter describes the design and development of our proposed evaluation framework for the analysis and evaluation of different types of recommendations systems, mainly clustering algorithms and recommendation systems. The framework focuses on putting together the following main aspects of recommendation systems: activity recommendation, group creation, and algorithms evaluations. This is an open source technological tool that will allow us to compare different algorithms and models.

Part of the content of this chapter is described in the study published in:
Rodas Britez, M., Marchese, M., and Cernuzzi, L. (2017). Towards a social and physical Activities recommendation system for active ageing. IX Congreso Iberoamericano de Tecnologías de Apoyo a la Discapacidad - Iberdiscap 2017, ISSN 2619-6433, pages 452–459. [77]

Chapter 4. Design of a Leisure Activity Recommendation (LAR).

This chapter describes a model and a recommendation algorithm design for recommending leisure activities. The model is a Dimension Model of the activity, the recommendation algorithm approach is a content-based recommendation, and the preferences are 7-scale ratings. We describe and compare tags and dimensions as possible representation of the activities. Finally, we elaborate on a discussion of the advantages and disadvantages of designing a leisure activity recommendation.

Chapter 5. Evaluation of a LAR Dimension Model.

In this chapter presents the results of evaluating the LAR dimension model proposed as a content-based recommendation. We describe the analysis of a dataset of leisure activities, clustered using either dimensions or tags, and compare the ability of dimension-based clusters against tag-based clusters to predict user preference for the activities. Clustering and performance evaluation required collecting three pieces of data: per-activity scores on dimensions, tags related to activities, and user preferences for activities. All three data collections relied on crowd-sourcing. We describe the issues and features of the implementation of an activity recommendation system.

Part of the content of this chapter is described in the study accepted in:
Miniukovich, A., Rodas, M, Jovanovic M., Marchese, M. (2019). Towards Engineering Leisure

Recommendation. Proceedings of the 5th International Conference on Fuzzy Systems and Data Mining, FSDM 2019, Kitakyushu City, Japan.

Chapter 6. Evaluation of LARs using FER.

This chapter describes the performance analysis for the rating-based activity recommendation system. We chose to use the following metrics: Precision, Recall, F1 Measure, Normalized Discounted cumulative gain (nDCG), accuracy, coverage, and transparency. We discuss the advantages and disadvantages of the recommendation algorithms in terms of the evaluation metrics.

Part of the content of this chapter would be considered for the submission in the ACM Conference on Recommender Systems, showing the different comparisons of the implemented recommenders.

Chapter 7. Evaluation of an User Tag Model using FER.

This chapter describes the clustering analysis of the results obtained by collecting information about users, preferences, and groups from the Meetup social network. The main idea is to understand how Tags could be useful to describe user preferences in relations to groups that in the social network has the intention to incentive face-to-face meetups between users. This test case includes a group of older adults. We also show how clustering algorithms allow us to better understand the relations between activities and which algorithms perform better with our data collection. The clustering analysis initially has been part of an Master's thesis [76] at the University of Trento that we supervised.

The content of this chapter integrates the study published in:

Rodas Britez, M., Lissoni, D., and Marchese, M. (2018). An evaluation framework for group's clustering algorithms in social networks - the use case of a meetup dataset of older adults. In Tallón-Ballesteros, A. J. and Li, K., editors, Fuzzy Systems and Data Mining IV - Proceedings of FSDM 2018, Bangkok, Thailand, 16-19 November 2018., volume 309 of Frontiers in Artificial Intelligence and Applications, pages 417–427. IOS Press.

Lissoni, D. (2017). Clustering Algorithms and Recommender Systems Analysis. A Comparative Java Oriented Framework. Master's thesis, Department of Information Engineering and Computer Science. University of Trento, Trento, Italy. [76]

Chapter 8. Conclusions and Future Work.

This chapter summarizes the contributions of the thesis. The contributions are described in relation to the research questions (RQs) of this thesis. Finally, we comment on the limitations and future work.

Chapter 2

Recommendation Systems: Background

The whole is more than the sum of its parts.
Aristotle

Social networks and cyber-physical networks are current technological sources of information and social sharing of information. On one side, this describes the complexity of the interactive system of information, on the other side, it opens the opportunity to build and process information using innovative information systems like recommendation systems.

Recommendation systems are important tools for social networks to provide useful information to the users, keeping their attention and interest. In the literature, there are numerous studies on recommendation systems, but the large majority is using just one field of evaluation for the recommendations [69].

The main components of recommendation systems are users and items. The users have personal characteristics and opinions about items and their features. The interactions between components in a recommendation system are given by the user's opinions over the items.

In our work, the main example, the user domain for our activity recommendations will be for older adults. From the technical point of view (Clustering and Recommendation Systems) this Chapter will present the different types of techniques for recommendations systems, with some examples of implementations in different domains. From the social point of view (Leisure Activities) the emphasis is on the item and user model, analyzing content-based approaches for understanding the activities, and analyzing model-based approaches for understanding the users.

This Chapter focuses on giving the technological insides for understanding the feasibility for recommendation systems search on the **RQ1**. The evaluation section of this Chapter contributes to the **RQ2** by explaining possible evaluations metrics for recommendation systems. Finally, this Chapter is the needed base for eventually answer the **RQ3**.

2.1 Recommendation Systems General Definition

Recommendation systems emerged around the mid-1990's when researchers started focusing on recommendation problems that explicitly rely on the rating structure [3]. This tendency also followed by the industry that starts recognizing the opportunities of using this novel technology, especially in the e-commerce business [7].

Initially, the recommendation problem is reduced to the problem of estimating ratings for the items that have not been seen by a user. Intuitively, this estimation is usually based on the ratings given by this user to other items and on some other information. Once we can estimate ratings for the yet unrated items, we can recommend to the user the items with the highest estimated rating.

The utility of an item is usually represented by a rating, which indicates how a particular user liked a particular item, e.g., Maria gave the activity "playing football" the rating of 4 (out of 7) [3].

Ratings are grounded on the idea that we have an explicit value that defines the preferences of the users over the items, called rating. In this case, we do not need additional information about the users and the item for the process of the recommendations. Ratings could be represented in different scales, e.g.: 5-liker scale or 7-liker scale.

To estimate the unknown ratings using the known ratings could generally classify into these two techniques [3]:

- Heuristics: specifying heuristics that define the utility function and empirically validating its performance.

- Statistics: estimating the utility function that optimizes certain performance criterion, such as the mean square error.

In the literature, there are numerous studies on recommendation systems, focus on specific domains. We are interested in recommendations of leisure activities considering the social factor of doing activities in groups. Important aspects of this context are that it happens in a social environment, and typically using social networks.

2.2 Recommendation System Techniques

The two more general types of recommendation systems are described as their main elements, users, and items, as shown in Figure 2.1. Basically, content-based filtering refers to the recommendation that uses additional information (content) of the recommended item. On the other hand, collaborative filtering refers to the recommendation that uses additional information related to the user.

Collaborative filtering has more sub-classifications shown in Figure 2.1, and generally refers to techniques that rely on the user community information. The model-based filtering refers to techniques that define a model that represent the similarity of users to use the information of similar users as collaboration. Then the memory-based filtering refers to techniques that use past information of the user and the item for recommending. Generally speaking, item-based recommendations are recommendation systems that use past information of the preferences over items to process the recommendations. Then, user-based recommendations are recommendation systems that use past information of the users to process the recommendations.

Hybrid filtering describes different approaches for mixing together the other recommendation system techniques.

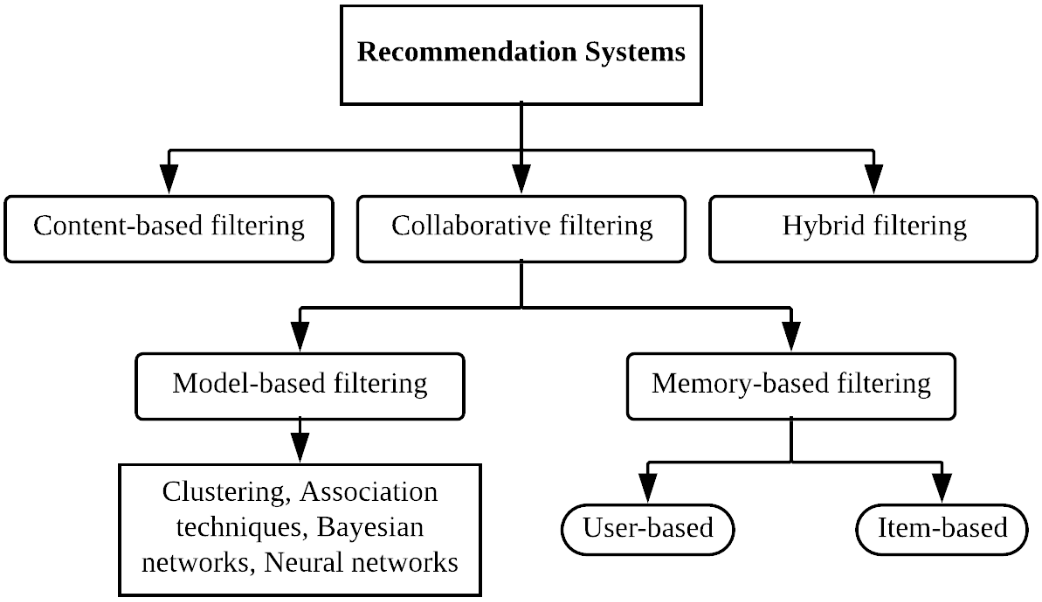


Fig. 2.1 Recommendation System techniques.

2.2.1 Content-based Filtering

The typical recommendation systems recommend items basing their recommendations strictly on users' feature such as users' likes, preferences, friends, objects they bought or they searched and so on and so forth. But also the environment in which users live, friends' features, advertising seen, news red, life quality, and lifestyle, are all factors that indirectly affect the users' way of thinking and therefore their choices. Moreover, people operate in groups not individually, they like to share news, talk in groups and do group activities. For example, it is much more pleasant to exchange opinions among friends on news that everyone knows that talking about an event that only a single person has seen. Furthermore, some studies mentioned in [39] proved that items, events, and activities are used/performed at least as often by groups as by individuals. Groups recommendation systems try then to fill the gaps of personalized recommendation systems brought by the lack of social parameters by creating a batch personalized environment. In addition, grouping recommendation can also bring benefits to some personalized recommendation systems problems, brought to the fast growth of data made available on the Internet, such as algorithms running time and resources required for their execution.

2.2.2 Collaborative Filtering

Collaborative Filtering is the information filtering systems that deal with the problem of information overload by filtering vital information fragment out of a large amount of dynamically generated information according to user's preferences, interest, or observed behaviour about item [38]. Basically, recommendation systems create users' personalized environments. This concept introduces the idea of finding inter-related information from a group of users or a group of items.

Recommendation systems approaches are used by almost every big company, especially in the e-commerce area, and therefore they are constantly evolving [7]. New social information starts to be considered as recommendation features like the influence of society on users.

2.2.3 Hybrid Filtering

The combination of collaborative filtering and content-based approaches are called hybrid methods. There exist a huge number of possible combinations between collaborative filtering and content-based systems but the most utilized are:

- Implement more than one recommendation and then combine predictions in some ways, for example through a linear model.

- Mix together content-based recommendation with collaborative filtering.

2.2.4 Clustering Systems

Clustering is the process of assignment of items to groups so that items in the same group are more similar than the items in other groups. As mentioned before, clustering could be a technique to implement or improve a recommendation system giving the potential advantage of reducing the size of the dataset used to process the recommendations [49]. Additionally, clustering could help with the cold start problem, that basically describes the problem that certain algorithms have at the start of putting into production the algorithm, and there is lack of information or low amount of information.

Clustering is particularly important for the recommendation systems of our domain because of the possibility of building a model-based recommendation system using clustering, or as a hybrid filtering approach. Also, clustering is important to study the opportunities that some activities are better done in groups and that doing social groups could support the sociability part of the wellbeing of people.

One clustering approach is the affinity propagation, that is a feature-based algorithm, and it is a potentially useful example of a classification approach [25, 30]. The affinity propagation algorithm groups data by finding a set of prototypes or exemplars for each cluster. This algorithm could be used to solve different types of problems: pattern recognition on images [11], and text clustering with few labelled objects [30]. The Affinity Propagation algorithm can also be extended with a seed-based initialization process to improve precision and efficiency [30].

Recommendations have been also studied in groups context, helping with items selection [8, 90, 13, 43]. It is found that a number of activities are done better in social groups, so it is important to leverage this aspect in the technological approach.

Another analysis of the characteristics of the group recommendation system motivates the group creation perspective with the existence of social processes like conformity and emotional cognition [13]. One of the last surveys on group recommendations classifies using three generalizations: user profile aggregation, user recommendation aggregation, and group model [43]. This research also discusses the social influence on groups recommendation, including the personality analysis approach.

There are various clustering methods and they are currently widely used [41]. One characteristic of clustering algorithms is the level of belonging of the element with the group, defining two main clustering types:

- Hard clustering: clustering where each data-point or belongs entirely to a cluster or does not. This means that different clusters cannot overlap.

- Soft clustering: clustering where each data-point belongs to every cluster with a certain probability. This means that the output of this clustering methodology does not consist in a set of well-defined groups but rather to a set of data-points with probabilities that describe how much a data-point belongs to a certain group.

2.3 Recommendation Algorithms Evaluations

Evaluation is the ability of the system to evaluate the results to estimate the strength of the relationship of the recommendations. The literature on recommendation algorithms distinguishes typically between two broad categories of measuring recommendation accuracy: *rating prediction*, often quantified in terms of the root mean square error (RMSE), and *ranking*, measured in terms of metrics like precision and recall, among others [88].

There is an additional complexity of the evaluation considering that the clustering algorithms are part of some implementations of recommendations algorithms.

2.3.1 Clustering Evaluation

For the evaluation of clustering algorithms, there are two general evaluation areas we implement: internal criterion, and external criterion. These evaluations areas are described as follow:

- Internal Criterion Evaluation: the evaluation that takes into consideration information of the generated clusters for the estimation of the strength of the relations of the clusters.

- External Criterion Evaluation: the evaluation that uses an external source of validation to compare with the generated clusters searching to be as similar as possible.

The evaluation with an external criterion consists in obtaining a real dataset of users and groups that represent the external criterion. Usually, this dataset is called the *ground-truth* dataset and the related evaluation technique the *ground-truth* approach. The idea then is to test various algorithms and identify the one that produces the most similar results with respect to the "ground truth" dataset. This technique introduces the problem of defining a similarity metric. One type of similarity metric is based on the distances between cluster centroid [87]. Also, one could instead calculate clusters similarity by using the Jaccard similarity measure between cluster instances pairs [94]. Another classic way to calculate clusters similarity is to count how many instances two clusters have in common.

Another measurement technique is to use the external criterion without using a similarity measure by calculating the purity measure[82]. The purity measure is evaluated on how well clusters matches with a predefined set of classes.

Another standard way to measure the overall quality of a clustering algorithm performs is to measure the performance in terms of the Precision, Recall, and F-measure as described in [82], by identifying the following variables of the classification: true positive (*True-Positive*) as the correctly assigned clusters, false positive (*False-Positive*) as the incorrectly assigned clusters, and False Negative (*False-Negative*) as the not assigned clusters when it should be assigned. The formulas are described as the following:

- $Precision = \frac{True-Positive}{True-Positive+False-Positive}$
- $Recall = \frac{True-Positive}{True-Positive+False-Negative}$
- $F\text{-}measure_\beta = \frac{(1+\beta^2)(Precision*Recall)}{\beta^2 Precision+Recall}$, where β weighs the importance of Precision and Recall (β higher than 1 weighs more the Recall).

Precision, Recall, and F-measure are used to evaluate classification supervised learning algorithms, and they are built over the confusion matrix, which is a matrix that shows how many users were incorrectly assigned [82]. Since some clustering algorithms use also unsupervised learning approaches, in these cases to generate the confusion matrix there is the need to use ground-truth groups as the corrected prediction. Ground-truth is then used to define the "predefined classes" (groups), while the clusters generated represent the "actual classes". The confusion matrix is created through the comparison between the predefined cluster instances and the actual clusters. Precision is calculated as the fraction of pairs correctly put in the same cluster, Recall is the fraction of actual pairs that were identified, and F-measure is the harmonic mean of the precision and the recall [82].

Another way of comparing two clusters, based on information theory, is called "variation of information" [55]. This measurement defines a distance between two partitions of the same data set, by measuring the amount of information lost and gained in changing from one cluster to another one.

Evaluation of the recommendation system is defined as the ability of the system to make correct recommendations. The literature on recommendation systems distinguishes typically between two broad categories of measuring recommendation accuracy: rating prediction, often quantified in terms of the root mean square error (RMSE), and ranking, measured in terms of metrics like precision and recall, among others [88].

2.3.2 Recommendation Evaluation

Recommendation systems are implemented for recommending clusters of users and activities. The evaluations of the recommendation systems are based on their ability to generate the right recommendations.

There are two main ways in order to evaluate a recommendation system: rating predictions or ranking.

The rating prediction is based on the comparison between the predicted values of the recommendation system recommendations with the ground truth users' rating of the items. This methodology is usable, therefore, only when recommendation systems are executed on a dataset which provides users the possibility to rate items. The formula typically used in order to evaluate recommendation systems through rating prediction is the root mean square error (see Formula 2.7 where X is the vectors containing the predictions while Y stays for the vector composed by the ground truth values).

Ranking instead, is a recommendation system evaluation methodology based on the creation of the confusion matrix and its attached formulas (precision, recall, accuracy, etc.). Also, this evaluation is therefore computed through the comparison between the recommendations generated with the ground truth data.

For the evaluation of the recommendations having a ground truth we could calculate the following metrics:

The precision measures the amount of relevant selected elements over the total amount of retrieval elements, as shown in the Formula 2.1.

$$Precision = \frac{True-Positive}{True-Positive+False-Positive} \tag{2.1}$$

The recall measures the amount of relevant selected elements over the total relevant elements, as shown in the Formula 2.2.

$$Recall = \frac{True-Positive}{True-Positive+False-Negative} \tag{2.2}$$

The F-measure measures the relation between the Precision and the Recall, as shown in the Formula 2.3.

$$F-measure_{\beta} = \frac{(1+\beta^2)(Precision*Recall)}{\beta^2 Precision+Recall}. \tag{2.3}$$

$$Discounted\ CG_p = DCG_p = \sum_{i=1}^{p} \frac{rating(i)}{log_2(i+1)} \tag{2.4}$$

$$Ideal\ DCG_p = IDCG_p = \sum_{i=1}^{|REL|} \frac{rating(i)}{log_2(i+1)} \tag{2.5}$$

Normalized Discounted Cumulative Gain (nDCG) measure the ranking quality of a list of recommendations, as shown in the Formula 2.6.

$$Normalized\ DCG_p = \frac{DiscountedCG_p}{IdealDCG_p} \tag{2.6}$$

Root Mean Square Error (RMSE) measure the standard deviation between the measured values from the actual values (prediction errors).

$$RMSE(X,Y) = \frac{\sum_{i=0}^{n}(x_i - y_i)^2}{n} \tag{2.7}$$

These evaluation metrics are usually evaluated with different size of the training selection (10% to 100%) for the information retrieval evaluations. The idea behind is to measure the performance of the recommendations from the small size of information to the big size of information.

2.4 Implications of Social Network Sites for Recommendation Systems

Social network sites (SNS) are web-based services that allow individuals to (1) construct a public or semi-public profile within a bounded system, (2) articulate a list of other users with whom they share a connection, and (3) view and traverse their list of connections and communications, and those made by others within the system. The nature and nomenclature of these connections may vary from site to site [9]. These sites are ways of supporting real-life social networks, giving new public visibility, and giving the virtual opportunity to analyze and organize relationships.

One interesting aspect of social networks is that they are useful tools to obtain information regarding user profile, preferences and activities. In addition, these sites are one of the main interfaces of communication between users and the virtual world.

One of our case studies focus on activity preferences of older adults, so, we did an analysis of social networks sites for older adults. This SNS are becoming more and more relevant in today's society. In the European health policy framework [67], from the World Health Organization (WHO), are listed several solutions suitable for older adults in the section referred to the evidence-based strategies to be applied and the key stakeholders. Among them, an important category of solutions focuses on health, social services and support for informal care and social networks.

The aim of the mentioned international initiative is to reduce health inequities and promote the empowerment of older adults through health literacy and disease self-management. This also shows the current increased interest in dealing with the active living of older adults.

For the older adult's case study, we performed an analysis of the current social networks, comparing the features between popular general social networks (Facebook and Google+) and older adult's social networks (Blom, Sentab, GrandPad, Ownphone, Grandcare, Stich, Older is Wiser, Eldr, Breezie, Connect Living and Max 50plusnet). We found little differences in the coverage of the features. The main differences lay on the ways they display the information to the users and the procedures put in place to use the existing features.

We have seen that current SNS functionalities are very similar to each other. However, older-adult oriented SNS focus on (1) easy configuration, (2) a good support system and, (3) easy to use functionality (few and easy steps). In general, SNS for older adults focuses on facilitating configuration options and usability, e.g, connectivity already set up, or the support service could do it.

Moreover, with this analysis of virtual communities available in our technological reality, we have analyzed the social, regulatory, institutional and market context from a non-technical perspective. This analysis shows the following opportunities:

- Active Aging is an issue addressed currently by governments and organizations. A number of European and national projects, like the ACANTO project related to the Horizon2020 Program [1], are a clear example of this intention.
- The number of older adults (from the United States) using social networks sites increased from 2% in 2005 to 35% in 2015, indicating a strong interest in such tools and technologies [71].
- In our study, we could not find SNSs that provide recommendations using activities, similar friends, health advises, etc.

Thus, the identified threats are the following:

- Seniors articulate many concerns with online social media, including the time required for legitimate participation, the loss of deeper communication, content irrelevance, and privacy [36].

- The purpose/benefits of SNS are not obvious for older participants [28]. Educators developing ICT programs for older adults need to take into consideration these populations' characteristics, attitudes, and beliefs about ICT, and in doing so they are more likely to accommodate their needs and interests. It is necessary to address concerns, lack of confidence with technology, and present ICT as personally relevant, user-friendly, while building knowledge and skills that equip them to navigate successfully with ICT tools and its on-going developments [98].

Finally, the success of social networks is clear in a younger population, but for older adults, it seems to be not so attractive because of existing weaknesses, namely:

- There is a gap between young people and older adults regarding the SNS and the aim for which they use them. This could generate more distance between young people and older adults, not allowing inter-generational sharing and communication.
- Technical difficulties and the fact that current Web design does not take the needs of older users into account [64].
- Mobility is a specific problem for older adult [26], so special care should be taken in the way events and activities are proposed to this user group.
- Decreasing cognitive abilities is a reality for older adults. In addition, where special care needs to be taken in the development of proper and adaptable user interfaces.

2.5 Implementation Examples

Considering the domain of appliance, the literature focuses on activity recommendation systems for older adults narrowing the concept of activity to physical activities [21, 65, 78, 42, 83], alimentary recommendations [10, 21], or news recommendations [52].

In regard to activities more closely related to health, clinical recommendations, we found that supporting tools should consider the following aspects [63]:

- Acquisition and validation of patient data.
- Modelling of medical knowledge.
- Elicitation of medical knowledge.
- Representation of and reasoning about medical knowledge.

- Validation of the system performance.
- Integration of different part of the recommendation system.

Considering the user groups creation and selection, recommendations were focusing on groups context [8, 90, 13, 43], including personality analysis approaches for the participants. Many activities are done better in social groups, so it is important to leverage this aspect in the technological approach.

2.6 Contributions

This chapter main contribution goes to the technical analysis of recommendation systems and their evaluation (**RQ1**, **RQ2**), but also gives the ground for explaining possible contributions of recommendation systems (**RQ3**). We need this analysis to understand the possibilities that the technological context provide to us. This chapter helps us to develop the requirements and the design of the Thesis.

The summary list of contributions is the following:

- **Recommendation systems**: This chapter gives the theoretical background for understanding the implementation and evaluation of recommendation systems. Basically presents the four typical techniques for recommendations systems: content-based filtering, collaborative filtering, hybrid filtering, and clustering systems. The clustering systems approach is proposed to eventually be used as part of a model-based or content-based approach. Also, some evaluation metrics are described. Finally, some examples of implementation are presented.
- **User Requirements**: the social understanding of the activities is an important factor of this thesis, that is why this chapter presents the analysis of perceptions of a specific domain of users: older adults. This will allow a better understanding of the user, and eventually, critique better the benefits and problems of possible recommendation systems implementations.

Chapter 3

Design and Development of a Framework of Evaluation of Recommendations (FER)

Great things are done by a series of small things brought together.
Vincent Van Gogh

Recommendations systems normally work towards selecting accurate items based on users preferences from an overwhelming amount of information. To achieve the best recommendations the different techniques trade off between been accurate with respect to previous preferences and been novel to recommend new items. Even tho, we focus on content-based recommendation systems, to understand the possibilities of recommendations system we need to compare the different types of algorithms and techniques. This is why we thought it was a good idea to build a framework from which we could start developing and comparing some approaches, and eventually improved the development time, maintenance and integration of new algorithms and techniques. Also, at the begging of this Thesis, the existing Java libraries were in early development time and without a consolidated framework.

This chapter presents a framework designed to provide analysis of groups' clustering algorithms and recommendation algorithms, giving the possibility to compare the results in a systematic way, and facilitating the developers in the selection of the best algorithms that fit better their specific requirements and use cases.

The proposed Framework of Evaluation of Recommendation (FER) is based on Java patterns and it is extensible: algorithms, statistics and evaluation metrics can be easily added. The initial development of the FER has been developed and described in the Master Thesis of Lissoni, 2017 [46], under our supervision, where the analysis has been concentrated in the clustering analysis.

Three different cluster algorithms were implemented and initially analyzed: K-means [33], Fuzzy K-means [5], and Affinity propagation [25].

Also, four recommendation algorithms were implemented using the Mahout libraries [20]: a user-based recommendation algorithm, an item-based recommendation algorithm, a Singular Value Decomposition (SVD) algorithm, a content-based recommendation algorithm, and a Hybrid recommendation algorithm.

We also have implemented some standard information retrieval metrics for comparing the algorithms: Precision, Recall, F1-Measure, Normalized Discounted Cumulative Gain (nDCG), and Root Mean Square Error (RMSE).

For the evaluation of clustering algorithms, we used the following classification: internal evaluation, and external evaluation. Internal evaluation means that we describe the clusters based on internal characteristics of the clusters (i.e. overlapping of certain characteristics). External evaluation means that we describe the clusters based on comparative with some ground truth dataset, where available.

Although the major focus of this Thesis is not the development of a software Framework, we chose to do this development because of two reasons: i) Existing libraries of recommendation systems are in a relatively early stage of development and with limited documentation and none of them are using existing datasets or examples related to leisure activities, and ii) Because of the early stage of development together with the emerging development of recommendations systems we could reasonably expect a technological evolution within short and medium terms.

The main focus of this chapter is to build the playground for incremental and systematic development and testing of different user's models, item's models, algorithms, and evaluation metrics. This, eventually, will facilitate the extension of the research to other machine learning techniques (e.g.: new hybrid algorithms) or other technological environments (e.g.: big data).

This framework will support mainly the **RQ1** by the implementation of the algorithms, and the **RQ2** by the implementation of the evaluation metrics. Finally, this chapter will also build the ground for addressing the **RQ3**.

3.1 Goals

The main goal of this framework is to develop an evaluation playground for recommendations systems. We aimed to reach the following three characteristics: i) To use the most stable libraries in Java. ii) Consider the possibility of parallel processing. iii) To obtain examples of recommendations for leisure activities.

The main features we want are the following:

- Diverse evaluations that allow fair comparisons among different algorithms.
- At least a base model of recommendation items (leisure activities).
- An architecture that allows high flexibility and expandability.

With these goals and features, we want to ensure that the research done with the framework could be easily verified and reproduce.

3.2 FER Main Features

A framework is designed as a reusable and extensible architecture for various application domains [72]. Usually, developers want higher productivity and shorter time-to-market for the development of object-oriented applications, and these goals are achieved through good design and reusable architectures.

The FER is built on top of the Apache Mahout library version 0.12.2, using Java 8 programming language. We use design patterns to structure the implementation.

To achieve good design and reusable architecture, in our overall design, we choose to adhere to the following principles:

- Extensibility: the framework should be extensible. This means that a user can add functionalities to the framework without changing the existing core code.
- Inversion of control: the framework maintains the control of the application life-cycle.
- Interfaces and class segregation: the framework should separate different functionalities into different interfaces and different entities in different classes.
- Dependency inversion: high-level framework components shall perform their functions using lower-level framework components, through the interfaces exposed by the latter.

3.2.1 Initial Algorithms

The initial algorithms are only a first set of algorithms we have started to use in our research. They have been selected searching for popular, and simple implementations of clustering and recommendations, trying to cover as many current techniques as possible. Our proposed FER allows adding a novel and more recent algorithms and evaluations when needed.

The initial clustering algorithms are the following:

- K-means.
- Fuzzy K-means.
- Affinity Propagation.

The initial recommendation algorithms are the following:

- User-based recommendation algorithm using ratings.
- Item-based recommendation algorithm using ratings.
- SVD recommendation algorithm.
- Content-based recommendation algorithm.

Additionally, the users-relation-item model of the framework is composed of people, activities, ratings, tags, and activities' dimensions.

3.2.2 The Architecture

The architecture in Figure 3.1 shows the different layers of implementation chosen in the design of the proposed framework. The use of data access objects (DAO), factories, abstract classes, and interfaces, are the main design patterns used to achieve the principles mentioned above.

One of the goals of the FER is to build a structure that can be used for the implementation and evaluation of most clustering algorithms and recommendation algorithms and, at the same time, build a structure that is extensible and adaptable for more high-level implementations. Object-oriented and in particular Java pattern allowed us to meet all these requirements. In particular, the patterns we selected are the following:

- Data access object (DAO): DAOs are classes with relative methods that are used to isolate the access to database entity using queries. Basically, DAOs are used in order to separate the logic for accessing the database from the operations we want to do after obtaining the data.

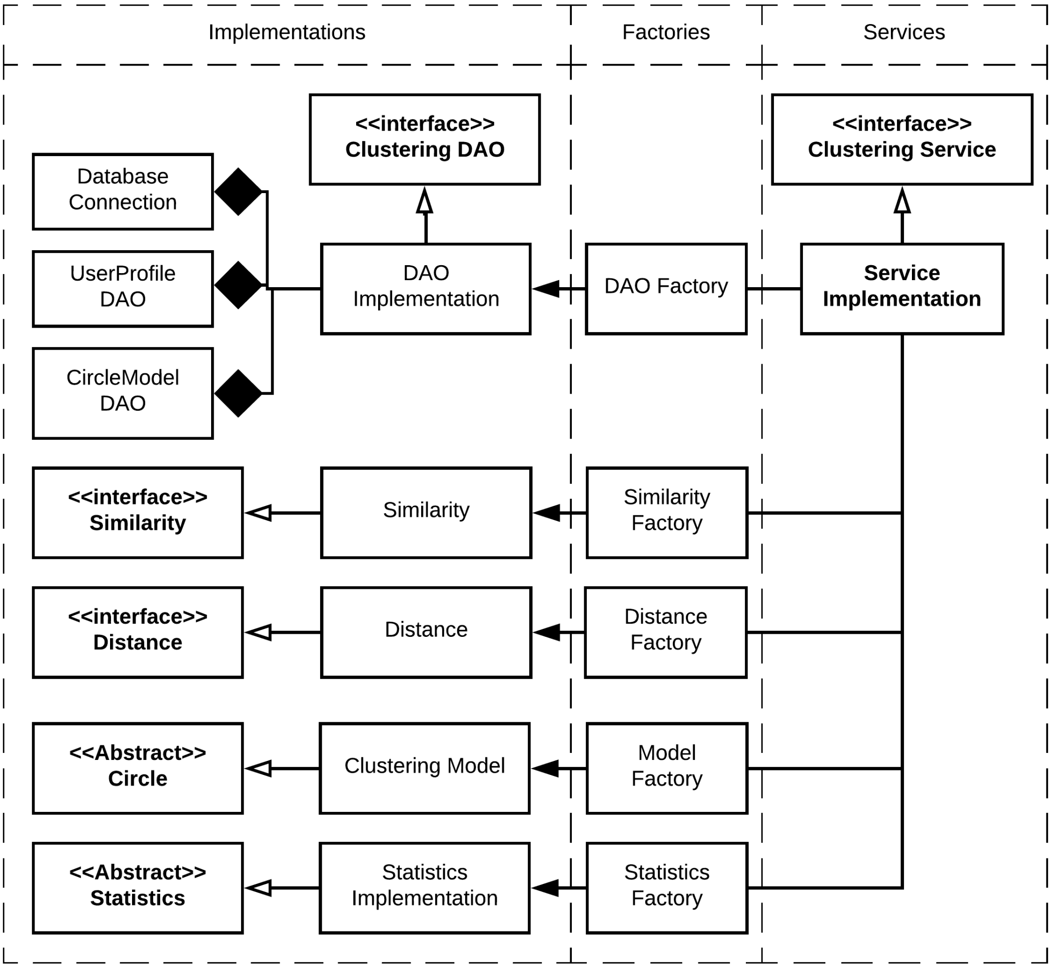


Fig. 3.1 Framework architecture

- Factory: Factories are used to encapsulate instances. The Factory is an inversion of control technique with the purpose of controlling the instantiation of particular classes of the framework. The factory works by allowing the developers to request for specific class instantiation.

- Abstract classes and interfaces: Abstract classes are classes that require the implementation of their abstractions and interfaces are definitions of classes without the actual implementation. The FER defines as abstract classes the plain old Java object (POJO) classes to have minimal entity classes. It will then be the task of the developer to implement the required classes. The structure of the business logic classes uses interfaces.

3.2.3 Abstract classes

For the implementation of the low-level implementation that access to the databases as abstract classes (POJOs and DAOs). The abstract classes implemented in the framework are described as following:

- *Circle*: POJO class that reflects the dataset Circle node. This class is used to store the clusters generated by the framework clustering algorithms. It contains a CircleStatistics that is a class used to generate and store statistics about the circle.
- *Comparison*: This abstract class does not refer to any node or edge in the database. The class has been designed to store data about the comparison between a circle generated by the algorithm and a group coming from the ground truth. This class contains methods used to calculate the percentage of the correctness of circle respect to a ground truth group.
- *ComparisonPairwiseService*: This abstract class defines the structure used to calculate the circle evaluation measures chosen: Precision, Recall, and F-measure.
- *Statistics*: This abstract class is used to generate statistics about the results of the clustering algorithm and its execution specifications. Statistics include the circle's sizes and circles preferences specifications.
- *ActivityGroup*: This abstract class contains the common functions for processing the Activity Groups for the different algorithms that create groups of activities.
- *MyRecommendationService*: This abstract service class contains the common functions for processing recommendations of activities to the users.
- *MyRecommenderIRStatsEvaluator*: This abstract service class contains the common functions for processing the information retrieval statistical evaluation.
- *MyMahoutTest*: This abstract class defines the structure for developing a MahoutTest.

Abstracts classes are the implementation of generic operations that can be executed with a generic representation of classes and they have common processes for all related classes. In the framework's architecture shown in Figure 3.1, the abstracts classes are classes with common process within a group of classes that are not needed to be repeated in every related class.

3.2.4 Interfaces

The business logic has been implemented using services. The list of implemented services are described as following:

- *ClusteringInterface*: All the classes in charge to execute a clustering algorithm must implement this interface. In particular, the interface contains the following methods:
  - *run()*: run all needed functions in their correct order;
  - *preProcessing (void)*: data initializer method;
  - *postProcessing (void)*: method used in order to refine the cluster output;
  - *process(void)*: method used to orderly call all the other methods;
  - *saveCircles (void)*;
  - *printStatistics (void)*: method used in order to print out all the statistics generated;
  - *deletePrecedentResults (void)*: method designed in order to delete circles and their relative connection of past algorithm execution;
  - *setCluster_Criterium(String criterium)*: save the criterium used for doing the clustering.
- *ComparisonServiceInterface*: This interface define the structure for comparison services. The interface define the following methods:
  - *run (void)*: method used in order to call the interface methods neatly;
  - *preProcessing (void)*;
  - *postProcessing (void)*;
  - *process(void)*;
  - *printComparison (void)*:
- *Similarity/Distance*: Developers can define the metric used by the clustering algorithms in order to compare users as input argument of clustering algorithms. The interface contain the method: double compare(v1[],v2[]) which is in charge to compare two vectors and return a numerical value.
- *CircleDAO* : DAO interface that defined functions used to perform the most common database operation.
- *RecommendationMahoutService*: This interface designs the structure for recommendation systems services made available by Mahout. The methods proposed are:

  - *preProcessing (void)*: used to create the data model. The data model basically consists of the user-item matrix on which the recommendations are based. The matrix should be created in a format that satisfies the Mahout recommendation system requirements;
  - *buildRecommender (Recommender)*: this method build the Mahout Recommender object. In the initialization of the recommender, the developer must define the recommendation system to use and its specification arguments. The most commons specifications are the similarity measure to use and the Data model;
  - *evaluateRecommendations (void)*: the method used in order to start the recommender system evaluation based on the recommendations generated by the system;
  - *printRecommendations (void)*: the method used to print out all the recommendations generated by the system;
  - *runRecommendation(void)*: method used to call the service methods in the correct sequence.

- *UserCircleRankDAO*: This DAO interface refers to the connection between Users and Circles. It is used to manage the persistence of the Is-Member database edge. This DAO is used by recommender services in order to create the user-item (score) matrix on which the recommendations are based.

- *RecommenderEvaluation*: This interface defines the structure for the classes used in order to evaluate the recommender systems. The predefined methods are:

  - *preProcessing(void)*;
  - *process(void)*;
  - *print (void)*;
  - *run (void)*;

3.2.5 The Database

Regarding the database choice, relational databases were not designed to cope with the scale and agility challenges that face modern application [61]. A dataset graph structure, on the other hand, allows to perform efficient, constant-time operation, and allow to traverse a big amount of connections per second per core [79].

Since, for our research purpose, we had to handle social information, our database choice has been oriented on graph structure databases and in particular to OrientDB database version

2.2.18. OrientDB is the first Multi-Model Open Source NoSQL DBMS that combines the power of graphs with documents, key/value, reactive, object-oriented and geo-spatial models into one scalable, high-performance operational database [68]. Our choice of OrientDB database is motivated not only by its graph structure but also by the presence of Java API since we choose Java as our programming language for developing the FER.

However, since the framework is extensible, other databases could be used by simply adding additional connections and data access objects classes.

OrientDB use link attributes to build the relations between vertexes, allowing the graph structure of vertexes (Table 3.1) and edges (Table 3.2). We design a database structure minimal and simple shown at the Figure 3.1.

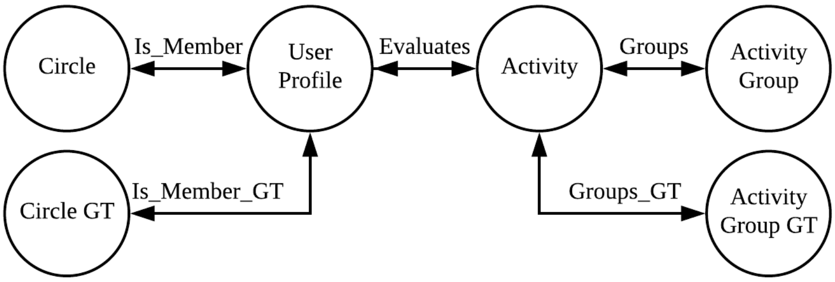


Fig. 3.2 Dataset Structure

The next list is a description of the Vertexes of our dataset:

- **User Profile**: is the node representing the User.
- **Activity**: is the node representing the Leisure Activity.
- **Tag**: is the descriptor for user's preferences and Activities.
- **Circle**: is the node representing the Group of Users.
- **Activity Group**: is the node representing the Group of Activities.
- **Circle Ground-Truth**: is the node representing the Group of Users of the Ground-Truth.
- **Activity Group Ground-Truth**: is the node representing the Group of the Activities of the Ground-Truth.

The next list is a description of the Edges of our dataset:

Table 3.1 Database Vertexes

Node	**Attributes**
User Profile	*rid (String)*: user id. *screenname (String)*: user name. *tags (Linkset)*: link to Tags.
Activity	*rid (String)*: activity id. *name (String)*: activity name. *tags (Linkset)*: link to Tags. *dimensions (EmbeddedMap)*: link to dimension's values.
Tag	*rid (String)*: tag id. *label(String)*:tag nominative.
Circle	*rid (String)*: circle id. *name(String)*: circle name. *tags(Linkset)*: link to Tags.
Activity Group	*rid (String)*: activity group id. *name(String)*: activity group name. *tags(Linkset)*: link to Tags.
Circle Ground-Truth	*rid (String)*: Circle id. *name(String)*: circle name. *tags(Linkset)*: link to Tags.
Activity Group Ground-Truth	*rid (String)*: Circle id. *name(String)*: circle name. *tags(Linkset)*: link to Tags.

Table 3.2 Database Edges

Edge	**Attributes**
Is-Member	*rid (String)*: edge id. *in (Link)*: Link to a Circle. *out (Link)*: Link to a User Profile. *timestamp (Date-time)*: creation date-time. *rank (Double)*: relation value of user-circle.
Evaluates	*rid (String)*: edge id. *in (Link)*: Link to an Activity. *out (Link)*: Link to a User Profile. *timestamp (Date-time)*: creation date-time. *score (Double)*: relation value of user-circle.
Groups	*rid (String)*: edge id. *in (Link)*: Link to an Activity Group. *out (Link)*: Link to an Activity. *timestamp (Date-time)*: creation date-time. *rank (Double)*: relation value of activity-activityGroup.
Is-Member Ground-Truth	*rid (String)*: edge id. *in (Link)*: Link to a Circle Ground-Truth. *out (Link)*: Link to a User Profile.
Groups Ground-Truth	*rid (String)*: edge id. *in (Link)*: Link to an Activity Group Ground-Truth. *out (Link)*: Link to an Activity.

- **Is-Member**: is the edge representing the relation between Circles and User Profiles.
- **Evaluates**: is the edge representing the relation between Activities and User Profiles.
- **Groups**: is the edge representing the relation between Activity Groups and Activities.
- **Is-Member Ground-Truth**: is the Is-Member edge for the Ground-Truth.
- **Groups Ground-Truth**: is the Groups edge for the Ground-Truth.

The elements of the database that are indicated as "Ground-Truth" are replicates of the user-evaluation-item model for the recommendation system, so we have it organized separately.

3.3 Clustering Implementations

The approach of recommending activities to a group is based on the idea that people usually prefer to do activities in groups: they like to share news, discuss in groups, and perform activities in a social environment. For example, it is much more pleasant to exchange opinions among friends on news that everyone knows, than talking about an event that only a single person has seen. In fact, these types of recommendation algorithms are interesting because events and activities are selected at least as often by groups as by individuals [39].

Clustering entities by features is a well-known problem in computer science [11]. Many algorithms have been proposed to perform this operation. However, typically the use of different clustering algorithms leads to different results i.e. different clusters/groups. Each clustering algorithm has some predefined parameters strictly based on the implementation of the algorithm. For example, the k-means algorithms are centroid cluster using the mean distance for the evaluation of their instances. Another example is the affinity propagation algorithm that calculates examples of the clusters, call exemplars.

This section is made to explore the clustering alternative in model-based filtering type of recommendation systems and possible techniques for implementing hybrid filtering with clustering. So we implemented some types of clustering algorithms identifying the common process to avoid repeating these common processes.

We found common operations inside clustering algorithms that we generalized defining the following operations abstractions:

- Run: execute the specific clustering algorithm.
- Generate-statistics: execute the statistics about the created clusters and the execution of the algorithm.

- Evaluation: execute the comparison of the resulting clusters with the ground-truth. The evaluations are the cohesion of the clusters and the pairwise comparison.

Eventually, every new clustering implementation should implement these common operations to work with our framework.

3.3.1 K-means Algorithm

The k-means algorithm defines the number of groups they want to generate, by assigning the k value. Defining the best k is a common problem in centroid models cluster algorithms, and it is often ambiguous. This is because the choice of the number of clusters strictly depends on the ratio between clusters dimension and clusters quality the users want to achieve. In fact, the bigger is the cluster, the clusters tend to be more heterogeneous. However, there are methodologies to approximate the best k according to the user requirements[92] [70].

The initialization of the k-means consists of the translation of the users as data-point in a feature space. Each data-point has to be represented as a point in an “n” dimensional space, where n represents the number of possible users’ preferences (tags).

The algorithm starts by randomly choosing k points (in the n-dimensional space) as initial centroids and assign each data-point to the nearest centroid. The distance is calculated through a metric predefined. The default distance measure is the Euclidean distance but also other metrics are provided by the framework.

Once all the data-points have been assigned to a cluster, the algorithm regenerates cluster centroids by calculating the center point between cluster data-points instances. The metric used to calculate the center point is the same one used in order to assign data-points to clusters. Finally, the k means re-assign every data-point to the nearest centroid. K-means runs these two last steps until convergence or until a predefined number of iterations has been reached up.

There are several Java libraries that implement the k-means algorithm, and we decided to use the Apache Mahout framework. Apache Mahout is a distributed linear algebra framework and mathematically expressive Scala DSL designed to let mathematicians, statisticians, and data scientists quickly implement their own algorithms [20]. We decided to use Mahout because of the following reasons:

- map-reduce K-means implementation: this gives the possibility to execute the algorithm on huge datasets using distributed systems.

- Comprehensive Java Framework: Mahout provide simple and intuitive framework Java structures and documentation.

As mentioned, users, in order to be processed by the algorithm should be translated in an n-dimensional numerical array. For the tag representation of the user preferences, the user's tags are treated as user's preferences, that a user may have or not have (1 or 0). The dimension of the user's array is the number of tags present in the dataset and each element of the user's array correspond to a tag.

The pre-processing method is the one in charge to execute this operation. It collects the data from the dataset and prepares them for the k-means execution. Furthermore, the pre-processing data execute operations in order to pass the input arguments required by Mahout. The arguments for the algorithms are the following: delta-convergence, classification threshold, and distance measure.

- Delta convergence: defines the value that determines when to stop the algorithm. The algorithm will stop when all the cluster centroids have not moved more than the delta-convergence value.

- Classification threshold: value defines instead when a data-points must create a new cluster or will become part of an already existing one. Basically, if the distance from a data-point to the nearest centroid is greater than the threshold, then, the object will create a new cluster.

- Distance Measure: is the specific distance formula that calculates the distances between the points of the dataset.

The k-means service uses then the pre-processing output data for the process method. The process method calls the Mahout function run-K-means that starts the algorithm. Once finished the execution, the service uses the process output for the post-processing function.

The post-processing saves and prints the cluster generated by the algorithm. The run-K-means output in-fact should be translated in order to have more understandable output data.

Finally, we generate the statistics (execution time and arguments setting) and the cluster generated (clusters size and quality specifications).

3.3.2 Fuzzy K-means Algorithm

Fuzzy k-means is the soft clustering extension of the k-means algorithm. The major difference between k-means and fuzzy k-means is that in the fuzzy k-means each data-point can belong to more than one cluster with a certain probability. Since users can belong to more than one group at the same time and for different motivation, fuzzy k-means seems to us an algorithm that potentially fits better our domain with respect to k means. In fact, users usually belong to more than one group.

Design, structure, and implementation of the fuzzy k means are pretty the same as the k-means ones. The only difference is the required "fuzzyness" input argument, a positive value which controls the extent of sharing among fuzzy clusters.

3.3.3 Affinity Propagation Algorithm

The Affinity propagation is an agglomerate hierarchical clustering technique. We chose affinity propagation essentially for three main reasons:

- Research Purpose: In order to compare the results on two completely different clustering techniques.
- User Representation: Cluster centroids (exemplars) in the affinity propagation are data-points (users). In the k-means implementations instead, centroids are just points in a certain vector space. This can be useful in order to find the most representative users.
- Less initial parameters: Affinity propagation does not require a number of clusters to be determined before the algorithm execution, avoiding the problem of the selection of a k value.

Affinity propagation starts with the creation of a similarity matrix between data-points to be grouped. This means that, in affinity propagation, data-points (users) are compared with each other using a similarity measure and not a distance metric as in the k-means. The similarity matrix is a *M x M* dimensional matrix where M represents the number of data-points. Each row and column element correspond to a data-point. The cells contain the similarity between the row data-point and the column data-point in question. The main diagonal of the similarity matrix represents the object preferences, a value that defines how likely a particular data-points is to become an exemplar.

The algorithm, once initialized the similarity matrix, executes the following operation until convergence:

$$resp(i,k) = similarity(i,k) - max_{k' \neq k}\left\{availability(i,k') + similarity(i,k')\right\} \tag{3.1}$$

$$availability(i,k) = min(0, resp(k,k) + \sum_{i' \notin (i,k)} max(0, resp(i',k))) \tag{3.2}$$

1. Calculate responsibilities (Equation 3.1): resp(i, k) reflects the accumulated evidence for how well-suited point k is to serve as the exemplar for point i, taking into account other potential exemplars for point i. Responsibility is sent from data point i to candidate exemplar point k.

2. Calculate availability (Equation 3.2): availability(i, k) reflects the accumulated evidence for how appropriate it would be for point i to choose point k as its exemplar, taking into account the support from other points that point k should be an exemplar. Availability is sent from candidate exemplar point k to point i.

The stopping criteria are the convergence achievement or the execution of a maximum number of iterations predefined.

We implemented the affinity propagation algorithm by using the APRO Java library. APRO is a Java implementation of the Affinity Propagation clustering algorithm. It is efficiently parallelized for use on multi-core processors and NUMA architectures (using "libnuma" native library), with a simple API [37].

First of all, the algorithm parameters that require to be set up before the algorithm execution are:

- Maximum number of iterations.

- Similarity measure to use.

- Data-points preferences value (if there are some).

Data points (users) are translated in numerical vectors through the “one shot” approach regard data-point features(tags), as explained in the k-means description. The pre-processing service function is the method in charge to perform this job. Furthermore the pre-processing initialize and populates the similarity matrix calculating the similarity between users. Finally, the pre-processing method creates the builder for the APRO library by passing it all the necessary data.

Once the pre-processing finished its execution, the process method starts. The process method calls, through the builder, created the APRO run() function. The run() function requires, as an input argument, only the number of maximum iterations.

The APRO run() function returns as output users and exemplars. Each output user is represented by an integer number, where the value of the number corresponds to the position of the initial similarity matrix created in the pre-processing.

The post-processing method is in charge to create the clusters by using the APRO output and translating the output in a user-friendly format.

The saveCircle() and generateStatistics() methods are the last business logic functions called in the service and their functionalities and implementation are the same mentioned in the k-means algorithm explanation.

3.4 Recommendation Implementations

The recommendation algorithms available in the framework were built using Apache Mahout. Structure and behavior of the recommendation algorithms reflect therefore the Apache Mahout requirements. The main processes for the recommendation systems are the following:

- Build Recommendation.
- Run Recommendation.
- Evaluate Recommendation.

The structure of the recommendation system service in the framework follows the same logic used in the framework for the service used for clustering algorithm execution. The recommendation system service has been divided into three main parts: pre-Processing, build-Algorithm, and run-Algorithm. The main components of the recommendation system are the following:

1. *Pre-Processing*: Read Dataset, and initialize variables.
2. *Build-Algorithm*: initialize the algorithm's objects.
3. *Run-Algorithm*: execute the algorithm.
4. *Post-Processing*: Evaluation of the algorithms.

In the framework, we implemented three different recommendation system techniques: user-based, item-based and SVD collaborative filtering. All the recommendation systems operate on the same users-circles (rank) matrix. Rank defines the affinity ratio between the user and the circle in question and is calculated in the clustering algorithm service when the clusters created are saved into the dataset. The rank is a ratio between the user's tags and the circle tags, described by the formula $rank = \frac{TotalUserTags - UsersTagsMissing}{5}$, where UsersTagsMissing is the number of users tags that do not equals to the best 5 cluster tags.

All the recommendation systems have been implemented on the top of the Apache Mahout framework. Mahout recommendation systems operate on a matrix of user-item ratings. In our case, items are groups and the rating is the rank.

3.4.1 User-based approach

The user-based approach predicts item-users rating by combining the ratings of other users that are similar to the user in question. The items (groups or activities) recommended to users are the ones for which have been estimated with the highest rating.

The Similarity between users is calculated based on the user's ratings. The similarity is calculated on the evaluation of items, where ideally, users with same ratings receive the same recommendation. There are many metrics to calculate the similarity and can be modified by developers.

For the implementation, we need to implement the *Recommender* object in charge of performing the recommendations. This object should contain:

- Similarity measure: the similarity measure that the algorithm will use in order to calculate the similarity between users. The similarity measure must implement the *UserSimilarity* class of Mahout.

- Clustering technique: for user-user recommendation systems Mahout only provides nearest neighbor approaches. There are multiple implementations of nearest neighbor made available by Mahout. The only constraint in order to choose an approach is that it must implement the neighborhood class.

- Data model: *GenericDataModel* object is used to contain and manage the data. The Data model should contain the data used in order to create the user-item (score) matrix i.e. groups, users and rank. Once collected the data, we can generate the matrix automatically. Data could be collected at running time easily from a CSV file or from a database.

The creation and instantiation of the *Recommender* object are performed in the framework in the *pre-Processing* recommendation service method.

Then, we need to instantiate the *Recommender* object, and request the recommendations with the *recommend(user, number of recommendation)* method. This method will return a list of *RecommendedItem* for the user passed as an argument. A *RecommendedItem* object contains the item recommended and the value of the recommendation for the user in question. In the case that there are not recommendations for a determined user, the list will return as a null object.

3.4.2 Item based approach

The item-based approach, to predict the rating of a certain item B by a user, makes a weighted average of the user's rating of items similar to B. The weight used is the similarity between B and the item in question. The similarity between items is based on the ratings they received by users. The similarity metrics are similar to the user-based approach.

For the implementation, we also need a similarity object and the Data model. Yet the clustering technique is not necessary since the weights for the predictions are obtained by the user own ratings.

The item-based recommendation system implementation has the same overall structure and behavior of the user-based one. The only thing that change is the builder instantiation. In fact, in the item based approach, the similarity must implement an Item-Similarity object. In our case we used the *GenericItemBasedRecommender* approach.

3.4.3 Singular value decomposition (SVD)

SVD is a latent factor based recommendation system technique. SVD approach works in the same way as the item-item collaborative filtering approach but, instead to operate on the whole user-item rating matrix, it operates on the user-item rating matrix transposed in a factor space (factorization matrix). The matrix factorization is in charge to reduce the number of variables used by merging some original matrix variables that vary together. The element that is created through the merge of more matrix variables is called a factor. This technique allows discovering possible correlations between features, creating also in this way a less sparse matrix. The matrix is reduced through matrix factorization and in particular by using the singular value decomposition. For more details about this technique please refer to the paper [29].

The real problem of matrix factorization is to find the right factor space of the matrix, that is the number of factors that will produce better results in the recommendation system. In fact, in the factorization matrix, each factor explains a percent of the total variance of the original matrix. Factors that do not explain much variance might not be worth including in the final model. So, ideally, the factorization matrix should contain the factors that explain the most part of the total original matrix variance.

The technique used in order to find the right number of factors is called factor analysis. Factor analysis is a technique that allows to highlight the existence of one or more factors, not measurable directly, within a set of directly observable variables.

In the framework, we implemented a factor analysis using the Principal component analysis (PCA) statistical procedure. PCA is a mathematical procedure that transforms a number of (possibly) correlated variables into a (smaller) number of uncorrelated variables called principal components. The steps we used in order to perform PCA are the following:

1. Initialization: Collect the data and create the user-group rank matrix;

2. Correlations: Calculate the group-group correlation matrix. As correlation measure has been used the Pearson correlation. The Pearson correlation is calculated by comparing the user's ranks respect the group in question.

3. Eigen Values: Compute the Eigen values of the correlation matrix. The Eigen value for a given factor measures the variance in all the variables which are accounted for by that factor.

4. Sort factors: Sort the factors in a descendant order respect to their Eigen value ratio and get neatly, starting from the first one, the factors until the factors collected will have explained the most part of the total matrix variance. This is because the ratio of Eigen values is the ratio of explanatory importance of the factors with respect to the variables. If a factor has a low Eigen value, then it is contributing little to the explanation of variances in the variables and may be ignored as redundant with more important factors.

In the framework, we implemented the factor analysis that takes in as input a matrix and will return the number of factors to use in the SVD recommendation. The class has been implemented in order to improve the quality of the recommendation system and also in order to avoid that the framework users will lose time in randomly finding the number of factors to use in the recommendation system.

For the execution of the algorithms, we need to specify the percentage of the variance of the matrix that we want to explain with the factors. The framework gets automatically the data from the OrientDB dataset and will return the number of factors that explain the percentage of the variance of the original matrix required.

The SVD recommendation system works in the same way as the item-based and user-based approaches. Then the framework procedure and structure are identical to them. The only thing that change is the instantiation of the *Recommender* object. In fact, in order to perform the SVD Mahout recommendation system the *Recommender* must be instantiate a *SVDRecommender* object. Its constructor requires as arguments the data model, the number of factors to use and the maximum number of iterations to perform.

3.5 Evaluations

For the evaluation, the class responsible for this process follows the design used for every service of the framework, using the following processes:

1. Pre-Processing: the collection of data need for the evaluations.

2. Evaluate: process the evaluation statistics.

3. Printing: print out the results.

An additional complexity of the evaluation of the clustering algorithms is the mixture and use within recommendations algorithms.

3.5.1 Clustering Evaluation

We approach the clustering evaluation with an internal criterion and an external criterion. These evaluations are as follow:

- Internal Criterion Evaluation: calculating internal quality (cluster cohesion) using the tags related to the users or the activities.

- External Criterion Evaluation: the evaluation that uses an external source of validation to compare with the generated clusters searching to be as similar as possible.

In the framework, we add three different analysis of the clusters: clusters size, clusters tags size and cluster cohesion.

The clusters size specifications, we calculate and print the size (number of instances) of each cluster and then we report, the biggest cluster, the smallest one, and the average size. Furthermore, we calculate the size distribution. The size distribution consists, for each different cluster size, in the percentage of the number of clusters having the size in question.

The clusters tags size specifications follow the same logic of the clusters size specifications. We basically calculate the cluster size using cluster features (tags) instead of that cluster instances. For each cluster generated, we calculate and report the cluster tag size (the number of different tags present in the cluster). Then we calculate the cluster containing more different tags, the one that contains the smallest number of tags and the average.

The clusters cohesion is a metric independent of the clustering algorithm and the domain of execution. We have indeed considered important to implement the cohesion metric in order to simplify the comparison between the results deriving from different algorithms. We decided to calculate the cohesion of a cluster basing the quality measure on the cluster instances' features (tags) for which instances have been grouped. We define the quality of a tag in the cluster as the percentage of the number of cluster instances that has the tag ($TagQualityInCluster = \frac{NumClusterInstancesWithTheTag}{NumClusterInstances} * 100$). The whole cluster quality is defined as the average of the cluster tags qualities. We report the cluster quality for each cluster and, as in the other evaluations, we highlight the cluster having the highest quality, the lowest one, the qualities average and the qualities distribution.

It is important to understand that the internal criterion evaluation by itself only gives an insight over the distribution of the information and relations of the clusters. These metrics provide useful insights into the behaviour of the algorithms with a specific dataset and usually cannot demonstrate that a clustering algorithm is better than others.

The evaluation with an external criterion consists in obtaining a real dataset of users and groups that represent the external criterion. Usually, this dataset is called the *ground-truth* dataset and the related evaluation technique the *ground-truth* approach.

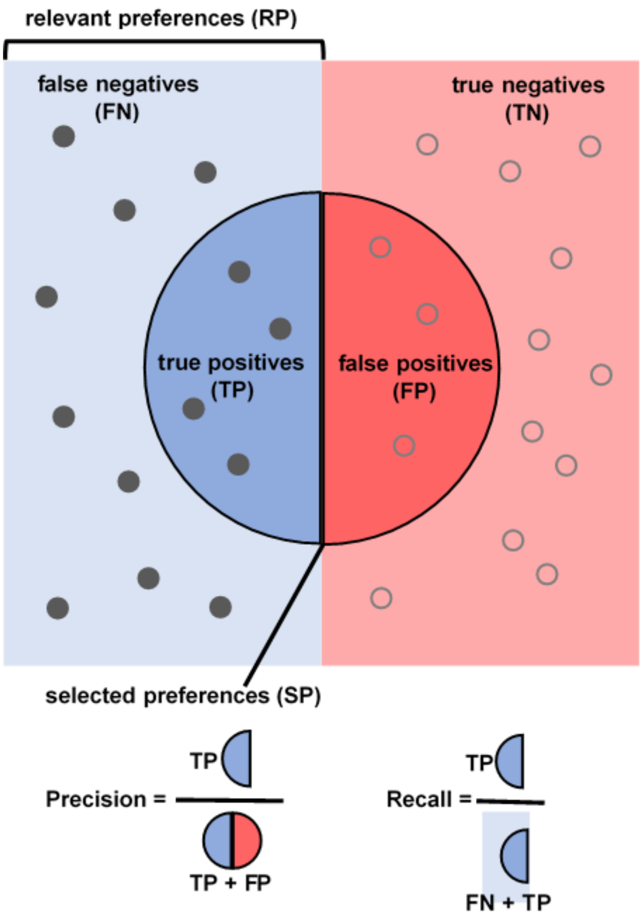


Fig. 3.3 Precision and Recall representation

We measure the performance in terms of the Precision, Recall, and F-measure as described in [82]. The formulas are described in and are shown in the following list:

- $Precision = \frac{True-Positive}{True-Positive+False-Positive}$
- $Recall = \frac{True-Positive}{True-Positive+False-Negative}$
- $F\text{-}measure_{\beta} = \frac{(1+\beta^2)(Precision*Recall)}{\beta^2 Precision+Recall}$.

Precision, Recall, and F-measure are used to evaluate classification supervised learning algorithms as shown in Figure 3.3, and they are built over the confusion matrix, which is a matrix that shows how many users were incorrectly assigned [82]. Ground-truth is then used to define the "relevant elements", while the clusters generated represent the "selected elements". The confusion matrix is created through the comparison between the selected elements and relevant elements.

For the clustering of users, we calculate precision, recall, and f-measure. The precision is calculated as the fraction of pairs correctly put in the same cluster. The recall is the fraction of actual pairs that were identified. The *F-measure*$_\beta$ is the harmonic mean of the precision and the recall, where β weighs the importance of Precision and Recall (β higher than 1 weighs more the Recall).

3.5.2 Recommendation Evaluation

The evaluations of the recommendation systems are based on their ability to generate the right recommendations. The general operations for the evaluation of recommendation systems are the following:

1. Initialization: Create a new dataset by removing randomly, from the original labeled dataset, some users preferences about items;

2. Run-Recommendation: Executing the recommendation systems and generate recommendations on the new dataset just created;

3. Evaluation: Compare the recommendations generated with the original labeled dataset.

There are two main ways in order to evaluate a recommendation system: rating predictions or ranking.

The rating prediction is based on the comparison between the predicted values of the recommendation system recommendations with the ground truth users' rating of the items. This methodology is usable, therefore, only when recommendation systems are executed on a dataset which provides users the possibility to rate items. The formula typically used in order to evaluate recommendation systems through rating prediction is the root mean square error ($RMSE(X,Y) = \frac{\sum_{i=0}^{n}(x_i - y_i)^2}{n}$ where X is the vectors containing the predictions while Y stays for the vector composed by the ground truth values).

Ranking instead, is a leisure activity recommendation evaluation based on the creation of the confusion matrix and its related metrics (precision, recall, f1-measure, nDCG), where two of them are shown in Figure 3.3.

When recommending groups to users, we found the problem that by recommending groups that the framework generates, we were not having ground truth data on which to compare the recommendations generated. Then we had to define a recommendation quality measure in order to evaluate the recommendation systems.

As recommendation quality measure we simply decided to use the user-group rank. In particular for every recommendation produced by the system the framework calculates the rank between the user that received the recommendation and the group recommended. Higher is the rank, higher will be the recommendation value. Then, in order to evaluate the system, the formula used is the average of the ranks respects every recommendation generated.

This recommendation quality measure used is also coherent with the way the systems produce recommendations. In fact, all the recommended systems implemented operate on the same users-circles ranks matrix.

The functions used by the framework in order to compute evaluations and generations of the statistics about recommendation systems are implemented in the EvaluationRecommendations class. This class is called by the recommendation service once the recommendation system finished its execution. Statistics calculated includes:

- Execution time.
- Number of generated recommendations.
- Number of users that do and do not receive recommendations.
- Average of the number of groups recommended per user.

Evaluations of the system instead are based on the recommendations quality measure and in particular the *EvaluationRecommendations* class reports:

- The average of the recommendations ranks (the value used to evaluate the system);
- The recommendation which achieved the highest rank value;
- The recommendation which achieved the lowest rank value.

3.6 Extending FER

The framework structure has been constructed in such a way that, interested developers, can easily incorporate new features. With the extension, we mean possible implementation and upgrade of the framework aimed at improving and extending the usefulness of the framework itself.

The framework extension allowed the addition of the following elements used in a recommendation system:

- Clustering algorithms.
- Recommendation systems.
- Cluster statistics.
- Similarity metrics.
- Cluster evaluation techniques.
- Recommendation systems evaluation techniques.

Currently, the requirement to extend the framework are the following:

- New customize classes should be implemented as abstract classes. This technique allows to framework users with easy application customization.
- Each service, DAO, or model class should implement its appropriate interface. Interface implementation is necessary to maintain a consistent framework structure and logic. If the class in question has a completely new logic and structure, a new interface must be defined.
- The instantiation of each class should be reachable by the appropriate factory class. Therefore, factories are modifiable. New factories can also be implemented in order to return new instances groups.
- The classes required by Mahout algorithms, if you want to extend the initial group of algorithms.

We suggest following the standard Java-Doc documentation of the code for a better understanding of the new improvements and additions.

3.6.1 Example: Create a New Recommendation Algorithm

For the implementation of a new recommendation algorithm you need to do the following steps:

1. Create a new Recommender class extending *AbstractRecommender* class. Implement if necessary the following functions: *recommend(long userID, int howMany, IDRescorer rescorer, boolean includeKnownItems)*; *estimatePreference(long userID, long itemID)*; *refresh(Collection<Refreshable> alreadyRefreshed)*.

2. Create a new Recommendation Builder class extending *RecommenderBuilder* class. Implement a *buildRecommender()* using the new Recommender.

3. Create a new Recommendation Service class extending *MyRecommendationService* class. Implement the *buildRecommender()* using the new Builder.

3.6.2 Example: Create a New Test Algorithm for Recommendations

For the implementation of a new test algorithm you need to do the following steps:

1. Create a new Test class extending *MyMahoutTest* class. Implement if necessary the following functions: *getService(int percentage, int at, DataModel fullDataModel, DataModel dataModel)* and *obtainTrainingDataModels(int min)*.

2. Create a new Main Test class using the new Test class, and defining the amount of recommendations required together with the evaluator class *MyRecommenderIRStatsEvaluator*.

3.7 Existing libraries

A number of free and open source machine learning Java projects focused on data mining algorithms, exists and are used to help developers during the implementations of such algorithms. Apache Mahout [4], for example, is a free Apache Software Foundation platform that makes available different scalable machine learning algorithms focused primarily in the areas of collaborative filtering built on the top of Apache Hadoop. Another popular Java framework is WEKA: WEKA is a collection of machine learning algorithms for data mining tasks [22]. Finally, LibRec is a Java Library for Recommender Systems focused on the study of rating prediction and item recommendation [32]. Regarding algorithms evaluations, all three libraries contain implementations of some evaluation measures.

Namely, Mahout provides only limited cluster quality evaluation, while WEKA has three different ways to measure the quality of a cluster:

1. The percentage of instances contained in each cluster.

2. The possibility to evaluate clustering on separate test data if the cluster representation is probabilistic (e.g. for Expectation Maximization).

3. Classes to clusters evaluation: in this mode WEKA first ignores the class attribute and generates the clustering. Then during the test phase, it assigns classes to the clusters,

based on the majority value of the class attribute within each cluster. Then it computes the classification error, based on this assignment and also shows the corresponding confusion matrix.

LibRec evaluation are related to the rating prediction (Mean Absolute Error, Root Mean Square Error, etc.) and item recommendation (Precision, Recall, nDCG, etc.) [32].

At the beginnings of the thesis, all existing libraries were in an early stage of development and with limited documentation, with no warranty for long support and maintenance. Even tho the distributed processing of large datasets was not the main requirement, it was also considered for the analysis and only Mahout was oriented to work in these terms using Hadoop.

Analyzing Python's libraries that implemented machine learning, although they contain several powerful libraries for machine learning, unfortunately, they do not always scale well to large datasets. This gives the advantage to Java alternatives when there is a need for process data that can not fit on a single machine.

Finally, the main goal of the FER is to help the developers in the evaluation and comparison of clusters and recommendations by providing an extensible organization of the algorithms and evaluation metrics, and by allowing the distributed execution of the algorithms. The addition of new libraries, algorithms and evaluation metrics is expected.

3.8 Contributions

This chapter addresses the design and implementation phases of the thesis and gives the foundations for new designs and developments of recommendation systems. The proposed Framework is the base for the development of the following chapters and provides opportunities for the expansion of the research with other datasets or techniques.

The main contribution of this chapter is the base implementation of recommendation algorithms and the technical analysis of recommendation systems with their evaluation (**RQ1**, **RQ2**), giving the ground for explaining possible contributions to the end users of the recommendation systems (**RQ3**).

The summary list of contributions is the following:

- **Evaluation Framework**: the main contribution of this chapter is to provide a playground for comparing recommendations algorithms. Basically, we present the three implemented clustering algorithms: K-Means, Fuzzy K-Means, and Affinity Propagation. Also, presents the four implemented recommendation algorithms: item-based recommendation, user-based-recommendation, SVD recommendation (model-based), and content-based recommendation. For the evaluation of the clustering we present: internal criteria implementation, we present some evaluation metrics: precision, recall, f1-measure, nDCG, RMSE.

- **Framework Extensibility**: the different characteristics of the framework (interfaces, abstract classes, design patterns) allow the possibility to extend the framework to other algorithms, metrics, or even domains.
- **Framework Scalability**: The implemented algorithms have the possibility for scalability to perform big-data executions because of the use of mahout libraries.

Chapter 4

Design of a Leisure Activity Recommendation (LAR)

Human behavior flows from three main sources: desire, emotion, and knowledge.
Plato

Health and wellbeing studies [89, 34, 45] highlight the importance of engaging in favorite activities, especially in later life. However, proposing suitable leisure activities to an individual is mainly done ad hoc, from a medical perspective by a professional. It results in pushing the user towards mainly physical activities while neglecting other aspects of well-being, and user preferences towards such activities. This approach can lead to user rejection of proposed activities.

As described in previous chapters, the recommendations systems could be useful technologies to assess recommendations to the users. So, we need to consider the benefits and problems of the different algorithms, adding the analysis of leisure activities.

We define leisure activity as a voluntary action done by one or many people for a period of time and it is intrinsically rewarding for the person (aka, fun) and is a goal in itself. Usually, researchers define a domain for the study of activities [2] and also search for intrinsic or extrinsic motivations for doing leisure activities [17]. We focus on leisure activities because they are considered important for the wellbeing of people [48] especially for older adults [23, 2].

When designing the leisure activity recommendation system we are dealing with two areas: building a data model, and selecting the best recommendation algorithms.

Regarding data modeling our approach is to model activities characteristics so we could process our evaluation of information on the activities, having a content-based approach. Another approach could be the clustering of users based on their preferences over the dimensions, and using this first step to reduce the size of users analyzed to receive a recommendation.

Developing a leisure activity model for a recommendation system may offer a better approach to the recommendation of activities. We start from the assumption that preference and liking are likely to determine if the user will actually engage in an activity. To describe both user preferences and recommended items, item-based collaborative filtering systems rely on location, free-form keywords (tags) [57, 14, 58, 44]. Another analysis will focus on using clustering algorithms to understand user groups preferences and classifications of leisure activities.

A tag-based approach is designed to be flexible by not limiting tags numbers and can be applied to different domains. However, this flexibility exposes limitations concerning recommendation performance and user experience. On one hand, algorithms struggle to generate meaningful recommendations until a large mass of tags has been accumulated, known as cold start problem [38].

To work on clustering leisure activities we want to find good descriptors of leisure activities, so we worked on understanding the leisure activities by running some clustering algorithms based on two descriptors (tags and dimensions). The definition of leisure activity involves many domain factors like energy expenditure, a portion of life that belong (sleeping, working, leisure), the intensity, exercises, physical control, etc [15, 56].

This chapter takes together all three research areas that this Thesis is working on (Leisure Activities, Clustering Algorithms, Recommendation Systems). This chapter presents the leisure activity model, uses clustering algorithms for implementing a content-based recommendation algorithm, and discuss the implications of the proposed design.

The main focus of this chapter is the discussion on the design of a leisure activity recommendation based on a content-based algorithm, and this contributes to mainly answering the **RQ1** and **RQ2**. The implemented model of leisure activities contributes to answering the **RQ1** and gives the basis for answering the **RQ3**.

4.1 State of the Art of LAR

The initial phase for recommendation systems is information collection [38] needed to build a model of the users, items and user-preference-item relations. One simple and common approach is to use tags to describe items to be recommended. The other approach includes developing a set of dimensions relevant for and descriptive of all items. For example, leisure activities can be characterized by their sociability, duration, and intensity of physical effort. Tags are, however, much more commonly used in the industry for recommendation, with such notable examples as SoundCloud[1] recommending music, IMDb[2] and Movielens[3] recommending movies, and Meetup[4] recommending events.

After the initial phase of data acquisition, a recommendation system learns the items a user likes and finds the items similar to the liked ones. A common approach to finding such similar activities is collaborative filter [38], which uses a model-based approach based on clustering items in coherent groups. This process could be further improved by constructing a set of user groups or item groups and then predicting a user rating for an item using the mean rating across group members [95].

An example of leisure activities analysis looking for exploiting recommendation of leisure activities can be seen at [66]. This research presents a user study for collecting psychological information and leisure activity preferences to analyze possible data relations.

We found research that proposed clustering of leisure activities based on participation ratings or the perceived needs that the activities satisfy, using factor analysis of some factors related to the perceived needs of the users regarding activities [47]. Also, another research developed an user-based leisure activity recommendation system based on the rating of the activities [97]. The User-oriented approaches are the typical approach for the analysis of leisure activities, when, on the other side, the content-based approaches main focus is on using only the place and time. Basically, researchers did not focus on other specific characteristics of the leisure activities that people usually consider when selecting a leisure activity.

We found an implementation of a context-based recommendation system of leisure activities tilt to treat the activity as an event (with time and place) and a with a generic classification of the activities [6]. This example implements a hybrid recommendation system in which the activity was modeled using a combination of patterns observed across the user's demographic population and individual behavior pattern, where the activities were classified in 5 modes: Eating, Shopping, Seeing, Doing, Reading. The problem with this approach is that the activity classification is fixed and the classification is a very high-level abstraction.

Similarly, some researchers studying activities in recommendation systems are focusing on user-based events recommendations systems [62], in which also a specific place and time are defined. Also, these approaches define leisure activities as a whole, without breaking down the particularities of leisure activities.

[1] https://soundcloud.com/

[2] https://www.imdb.com/

[3] https://movielens.org/

[4] https://www.meetup.com/

4.2 LAR Model Analysis

One popular approach for developing the content of an item within social networks sites is folksonomy [12]. This approach basically constructs the content of items by relying on an unstructured collaborative classification scheme obtained by the same users of the system. The users can annotate items with freely chosen words (tags).

An alternative to Tags is to actually build a taxonomy of activities and use it to describe the content of the items, which in our case is the dimensions of the activities. A dimension-based recommendation method would rely on each item (a leisure activity in our case) being represented as a short fixed-length vector of numeric values instead of a potentially unlimited-length list of item-related tags, which tag-based methods rely on. Each value in the vector characterizes a single dimension. Dimensions (e.g., socializing) are features of the recommendation domain (e.g., leisure), which all domain items possess to an extent (e.g., 'chatting with friends' presumes a lot of socializing, whereas 'swimming' presumes little of it). Dimensions should be descriptive of the domain, i.e., a good set of dimensions should allow for effectively differentiating two dissimilar items by showing a large difference in their dimension vectors. This property makes dimensions well suited for grouping items in clusters based on item similarity.

We relied on existing work to develop an initial set of dimensions for leisure activities. Mokhtarian et al. [60, 59] describe the aspects of leisure from the perspective of technology use and organize them as an activity-centred taxonomy combining task at hand, time, location, cost, planning, arrangement, and general person factors such as motivation and effort. We choose to rely on this taxonomy over other alternatives [93, 96] due to it including more aspects of leisure and focusing on the quasi-objective features of leisure. Less objective features – such as the needs that a leisure activity satisfies [93], e.g, the needs for self-expression or novelty – would require more observations aggregated per feature per activity than quasi-objective features to be measured reliably. The less-objective features would also be less effective in the recommendation: since it is done per individual and the scores that individual assigns to the features strongly depend on unique personal characteristics, the resulting recommendation would suit well the average user, but not the individual.

We have used the taxonomy and reorganizing the describing attributes of activities in different dimensions, translating them in a new group of activities dimensions, and developed semantic-differential measurement scales for each dimension. The resulting set contained 17 dimensions, as shown in Table 4.1 and in Figure 4.1. These dimensions are rated in 7-scale ratings for every activity, using the low anchor (1) to the high anchor (7) described in Table 4.1. All dimensions can describe any activity.

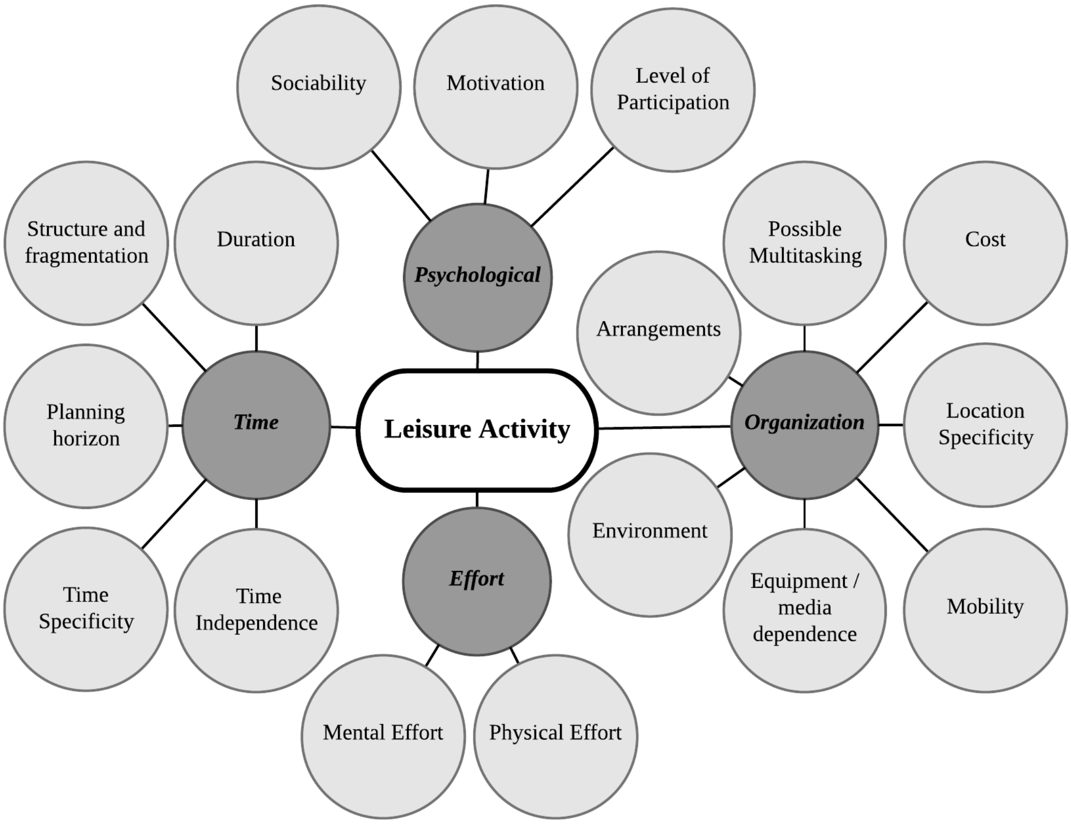


Fig. 4.1 Dimensions for the Leisure Activity Model

The dimensions in Figure 4.1 have been classified into 4 general groups: Time, Psychological, Organization, Effort. These general groups of dimensions allow us to abstract the idea that when selecting an activity we generally ask our-self four questions: 1) if we have time for it (Time), 2) if we can do it (Efford), 3) what I need to do it (Organization), and 4) what I have in return for doing it (Psychological).

Table 4.1 Dimensions of leisure activities.

Title	Description	Low Anchor	High Anchor
Physical effort	The amount of physical effort that an activity presumes per unit of time	Low	High
Mental effort	The amount of mental effort that an activity presumes per unit of time	Low	High

Environment	How much the environment where an activity takes place has been artificially transformed, ranging from low (outdoor) to high (indoor)	Low	High
Duration	An activity duration, ranging from short (minutes) to long (days)	Short	Long
Time independence	The level of temporal constraints and flexibility in the timing of activities, ranging from flexible (a person decides on timing) to fixed in time (advance time boundaries)	Flexible	Fixed in Time
Planning horizon	How much in advance the activity needs to be planned, ranging from short (planning horizon in minutes or hours) to long (planned ahead in weeks).	Short	Long
Time specificity	How much the activity depends on the time (part of a day, week, month or year), ranging from low (done at any time) to high (specific time of a day, week, month or year)	Low	High
Temporal structure and fragmentation	Whether an activity can be fragmented in several blocks of time or must be uninterrupted	Fragmented	Uninterrupted
Possible multitasking	How much of multitasking is possible during an activity, ranging from little (few activity-unrelated tasks are possible) to a lot (Many tasks are possible without stopping)	Little	Much
Sociability	The intensity of social interaction that an activity presumes, ranging from low (solitary activities) to high (activities are centred around socializing)	Low	High

Level of participation	The level of engagement in a way that affects the outcome, ranging from viewing (one doesn't influence the outcome) to doing (one entirely determines the outcome).	Viewer	Doing
Equipment/media dependence	The amount of training needed to participate in and enjoy an activity, ranging from low (no training needed) to high (is not possible without prior training).	Low	High
Arrangements	The amount of preparation and prior arrangement an activity requires, ranging from low (no arrangements needed) to high (a lot of effortful arrangement needed).	Low	High
Motivation	The type of benefits someone gets from an activity, ranging from hedonic enjoyment (solely enjoyable experience) to utilitarian gain (the activity results in material, social status, learning, health or some other personal gain).	Hedonic	Utilitarian
Cost	The total direct costs - including the cost of renting equipment and/or buying tickets, subscriptions and consumables - of engaging in an activity for one time, ranging from low (free of charge) to high (high monetary cost)	Low	High
Mobility	How much one's geographical location changes during an activity, ranging from low (stationary activities) to high (changing location is key)	Low	High

Location Specificity	How much an activity depends on the geographic location and climatic conditions, ranging from low (activity can be done everywhere) to high (an activity is restricted to a specific geographic location)	Low	High

4.3 Recommendation Preferences: Ratings

As explained at the Recommendation Systems Chapter, rating structure was a popular implementation of preferences in recommendation systems. So, to have an initial baseline, we choose to implement a rating structure as described in the FER Chapter.

Essentially, since is one of the baselines for recommendations we need to understand the benefits and the drawbacks of such implementation. The popularity of the rating structure for the preferences in recommendation systems basically was based on the simplicity of implementation, since you do not need to understand and model the item to recommend. Additionally, the Internet revolution gives support for considering recommendation systems as interesting topics of research and development.

A clear advantage of rating structure for recommendation systems is that currently is a good study solution for recommendation systems, and probably a trending topic in the current machine learning community. This situation comes together with the current increase of digital information using Social Network Sites.

The drawbacks of rating structures in recommendation systems are related to the current context of the technological environment. As explained in the Recommendation Systems Chapter current concerns on the adoption of technologies are privacy and security, so the fact the algorithms based on rating needs the ratings from the community is a problem. Additionally, a known problem for these recommendation systems is the cold start problem.

4.4 Recommendation Performance

A good recommendation system has multiple characteristics besides high accuracy, including good item-space coverage, ability to earn user's trust, preservation of user privacy and other positive characteristics [31]. User privacy has been a persistent concern for social networks and their increasing use of recommendation systems creates new possibilities for privacy to be compromised, e.g., if preferences of a user can be deduced from reviewing the recommendation generated for somebody from the social circle of the user [53]. Several recent approaches attempted to mitigate privacy risks in recommendation systems [27, 81].

The propensity to recommend the same, highly popular items - which corresponds to low recommendation coverage of item space - may not only hurt commercial systems, like sales platforms, [24], but be also perceived as boring. Users value new, unexpected recommendations [86], and some recommendation accuracy may well be sacrificed for this effect.

User trust leads to system adoption and continued use [73, 16]. While the majority of research meant trust as user confidence in recommendation, some research focused on recommendation transparency [85], which referred to the user understanding of why a particular item was recommended. A common approach to improving transparency included explaining the inner workings of recommendation algorithms to the user [35].

4.5 Content-based approach

Alternatives for solving the issues with rating structure in recommendations systems have been developed in the research community, like content-based algorithms, model-based algorithms, or hybrid systems. These approaches are described in the Recommendation System Chapter, and in this section, we want to describe one particular content-based approach for recommending leisure activities.

Contextual information on recommendation items has a significant impact on the process of decision-making. Additional contextual information like time, location, budget, weather, or social position are information currently been considered in recommendation system research [40].

A line of research for leisure activity recommendation is focused on the Tourism domain. Usually, researchers tend to implement hybrid approaches to increase the possibilities of the recommendation algorithms [62]. Tourism domain approaches take into account demographic data, details that define the context of the travel (e.g., the composition of the travel group), geographical aspects, the information provided explicitly by the user (e.g., main travel motivations), and implicit feedback deduced from the interaction of the user with the system.

We developed a content-based activity recommendation system by using the content information (dimensions) of the activities to generate clusters of activities, so we could recommend other activities with them.

Our content-based recommendation algorithms take advantage of the description of activities using the defined 17 dimensions. These dimensions are used to measure the similarity between activities using a specific clustering algorithm. After having these clusters we can use them to recommend leisure activities. This algorithm is described with the following list of operations:

- Initialization: Obtaining information and initializing need classes.
- Clustering algorithm of activities: Execute the clustering algorithms of activities base on the dimensions' values.
- Recommendation based on content: weight the dimensions of activities to recommend with previous ratings of the user and the dimension values of not rated activities.

The recommendation based on the content operations is better described in the Algorithm 1. This algorithms process the leisure activity recommendation using the information of the activities' dimensions and the personal rating of each user.

```
Data: String userID, Integer howMany, List<ActivityGroup> activityGroups
Result: List<Recommendations> results
baseGroup = userID rated activities;
adjustment = meanDistance(baseGroup, activityGroups);
foreach activityGroup in activityGroups do
    $\rho$ = Compare(basedGroup, activityGroup, adjustment);
    topActivitiesGroups = save(activityGroup, $\rho$);
end
min = minimumDistance(topActivitiesGroups);
max = maximumDistance(topActivitiesGroups);
foreach activityGroup in topActivitiesGroups do
    foreach activity in activityGroup do
        if activity was not been rated by userID then
            rating = getRating(activityGroup);
            finalRating = ((rating - min) / (max - min) *6) + 1;
            response.add(activity, finalRating);
        end
        if response has howMany recommendations then
            break foreach cicles;
        end
    end
end
```

Algorithm 1: Pseudo-code of the content-based recommendation algorithm

The function $Compare(baseGroup, activityGroup, adjustment)$ compares two groups of activities, show the distance between them and adjust the returned value with the value of $adjustment$. Inter-activity distances were calculated as Pearsons' correlations for dimensions (numeric data) and cosine distances for tags (binary data). The *baseGroup* is the group of activities that the user has rated and this is the only personal information used to predict the recommendations. The *activityGroups* is the group of groups of activities that could be simply one group of all the available activities to recommend or could, eventually, be generated using a clustering method. This comparison function internally calculates the pairwise distances between activities of the *baseGroup* $(size = m)$ and activities of the *activityGroups* using the following formulas:

$$\rho = \frac{\sum_{i=1}^{n}((d(activity_i, exemplar) - \delta) \times (rating_i - \bar{r}))}{n} \tag{4.1}$$

where *exemplar* is the best representation of the *activityGroup*, δ is the mean pairwise distance among activities, and $\bar{r}$ is the mean activity-preference ratings of one user. These representations are described as following:

$$\delta = \frac{\sum_{i=1}^{m}\sum_{j=1}^{p}(d(activity_i, activity_j))}{m \times p} \tag{4.2}$$

$$\bar{r} = \frac{\sum_{i=1}^{q}(rating_i)}{q} \tag{4.3}$$

The dimension-based algorithm was a particular case of the general content-based algorithm, with activities groups (*activityGroups*) having only 1 activity.

Finally, for the evaluation of this content-based recommendation of leisure activities, Chapter 5 and Chapter 6 present several evaluations from the model perspective and from the algorithm perspective.

4.5.1 Clustering

Regarding the clustering algorithm, we choose the Affinity Propagation algorithms because in one of our studies we found better performance. Yet, other clustering algorithms (k-means, fuzzy k-means, etc.) could be tested and compared. The advantages of the Affinity Propagation algorithms could be seen in the evaluations chapters of this Thesis.

4.5.2 Distances

The distance (d) is the mathematical measure of separation between two points. For comparing the distance of two activities in terms of the dimension's values, we implement the Pearson correlation coefficient (P). This coefficient is probably the most widely used measure for linear relationships between two normally distributed variables and thus often just called "correlation coefficient". Usually, the Pearson coefficient is obtained via a Least-Squares fit and a value of 1 represents a perfect positive relationship, -1 a perfect negative relationship, and 0 indicates the absence of a relationship between variables. The formula is described as following:

$$P = \frac{\sum_{i=1}^{s}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{s}(x_i - \bar{x})^2(y_i - \bar{y})^2}} \tag{4.4}$$

The variables x and y are two numeric vectors of length s; and $\bar{y}$ and $\bar{y}$ are the means of x and y, respectively. In our case, x and y represent the dimensions' vectors for two activities.

4.6 Hybrid-based approach

We have developed an implementation of an Hybrid-base approach using the user-based algorithm with a particular similarity measure that consider additionally the information of dimensions of Activities.

We implement a new similarity measure to develop an hybrid-based algorithms that uses the information of the Activities (Dimensions) and the preferences of the Users.

4.7 Discussion

The understanding of the main elements of the recommendation system (users and items) and the possible relationships (ratings, preferences, classifications, etc.) will help the development of better recommendation systems. We define the domain of the recommendation system in leisure activities and we model some characteristics of the activities to eventually use them.

The current state of the art of leisure activity recommendation systems tend to go towards the Tourism domain, giving more importance to the place, time and cost of the activity than actually understanding or modeling of the leisure activity. This could be seen at the systematic review [80], where context-aware recommendations system from 2000 till 2016 were analyzed and there were three implementations of activity-based recommendations systems in the areas of events, music, and traveling.

The examples of recommendation in the Tourism domain use demographic data and travel information to characterize better the item [62]. These examples could be added to the overall system but it would be shifting the problem to events recommendations where time and place are constraints of the system.

Our content-based leisure activity approach takes contextual information about the activities to recommend users activities. Practically speaking this means that no collaborative information is need from other users, but only some initial rating preferences of the users over some activities. This also means that this proposal is less affected by the cold-start problem compared with other collaborative-based approaches.

The state of the art of leisure activity taxonomies provide classifications of activities that are not standard, so a better and standardized understanding and classification of leisure activities will be helpful to implement in the industry. Our dimensions could be a starting point for standardizing the descriptions of leisure activities.

Comparing with other more popular recommendation systems, like SVD recommendation, the amount of calculus done is less. So theoretically speaking, our proposed algorithm could obtain results faster.

Considering that a hybrid method approach helps to avoid certain limitations of using only content-based or collaborative systems [3], we propose the content-based approach to eventually combining it with collaborative approaches.

Finally, for evaluation purpose, it is important to approach from two perspectives: performance evaluations, and user's perceptions. For time reasons the user's perceptions evaluations have not been done in this thesis, but it could add a lot of insights to the human interaction part of the recommendations, and eventually for implementations in the industry.

4.8 Contributions

This chapter main contribution is the design and implementation of a novel content-based recommendation systems addressing the **RQ1**. Then the discussion of the proposed recommendation gives an interesting explanation of the implication of such algorithms (**RQ3**). Also gives the ground for explaining possible metric evaluations of recommendation systems (**RQ2**).

The following list summarizes the contributions of this chapter:

- Activity dimension Model: the dimension based activity model that describe the activities from characteristics closely related to leisure activities like physical effort, mental effort, environment, duration, time independence, planning horizon, time specificity, temporal structure, possible multitasking, sociability, level of participation.

- Standardization: The proposed activity dimension model could be an initial idea for the standardization of leisure activities.

- Content-based Leisure Activity Recommendation: a novel content-based recommendation system has been proposed in this Chapter. The theoretical advantages are the following: a better understanding of leisure activities, low influence on the cold-start problem, and could have a good performance time-wise (if clustering is performed in batch).

- Recommendation system analysis: based on the current state of the art of recommendation systems we discuss the alternatives and analysis of techniques and approaches to implement leisure activity recommendations systems.

Chapter 5

Evaluation of a Dimensional Model for Leisure Activity Recommendations

What's measured improves.
Peter Drucker

This chapter follows the line of the analysis of the recommendation item, so evaluating a content-based approach for representing the activities in recommendations systems. We are collaborating in understanding the benefits and drawbacks of designing the LAR Model with tags and Dimensions.

This Chapter samples a dataset of leisure activities, clusters the activities using either dimensions or tags, and compares the ability of dimension-based clusters against tag-based clusters to predict user preference for the activities. Clustering and performance evaluation required collecting three pieces of data: per-activity scores on dimensions, tags related to activities, and user preferences for activities. All three data collections relied on crowdsourcing. The same set of 135 activities was used in all three data collections.

The objectives of the study were twofold: i) validating the capacity of our dimensions to describe leisure, and thus, to differentiate similar from dissimilar leisure activities; and, ii) comparing the performance of the dimension-based approach against the classical, tag-based approach of recommendation. The study relied on clustering as a research tool and one key assumption that most recommendation systems rely on: the user would like activities similar to the few activities that they had explicitly indicated as liked. If dimensions allow for producing clusters with all in-cluster activities either strongly liked or disliked, a dimension-based recommendation will perform well.

The evaluation of the LAR model contributes to the understanding of alternatives for designing leisure activity models into recommendation systems, using clustering algorithms as an evaluation tool. The analysis is done by doing some statistical analysis of groups of activities generated by clustering algorithms and using the tags and dimensions characterization of the content of the recommended item: activities.

This chapter focus on the evaluation of the recommendation model for understanding the LAR model related to **RQ2**. These evaluations allow us to understand how this model could contribute to the recommendation of leisure activities **RQ3**. Also, this chapter is an example of the implementation of leisure activities in a recommendation system, so, in a way collaborate to answer the **RQ1**.

5.1 Activities Dataset

Leisure activities are intertwined with the well-being of older adults [89, 34, 45]. However, a handful of studies concerned collecting preferred activities of older adults. NHATS (National Health and Aging Trends Study) is a longitudinal, ongoing study [34] with the community-living healthy older adults (N=5247) that collects data on the social and environmental living conditions, as well as their daily activities. A related study focused on favourite activities of older adults [89]. It showed that contrary to the stereotype older adults chose a physical activity as their favourite activity (walking, jogging, gardening, or playing sports).

For selecting our dataset of 135 unique activities (see Appendix A) we did the following analysis and selection:

- We extracted 58 activities from the original NHATS dataset and filtered out the activities that were either not clear from the data (such as 'no favorite activity' or 'no activity'), not specific enough (starting with 'other'), not considered as a leisure (such as 'work' or 'sleep'), or not ethical ('smoking' or 'gambling').

- Next, we took the activities from the interview study on the motivations of older adults (N=18) to engage in their preferred activities [18].

- Then, we conducted aggregations of the activities above and added leisure activities of the younger life to counteract the aging stereotype.

For building the context of the activities we use the dimensions described in the LAR Chapter. The activities were scores on the 17 dimensions and the scores were aggregated across participants for further analyses.

5.2 Dimensions in Activities: A Data Collection

5.2.1 Participants

English-speaking crowd workers (n = 386, 208 men, m = 33.88 years, from 18 to 72 years old)[1] received 80 US dollar cents for a 7-minute data collection session, which was comparable with the US minimum wage. Each crowd-worker could participate only once.

5.2.2 Procedure

Crowd-workers were redirected from the crowd-sourcing platform to the web-page introducing the study. After accepting the conditions of the study, they filled out a brief demographic questionnaire and rated randomly selected 62 activities on a randomly selected dimension using a 7-point semantic differential item. Seven activities out of 62 were duplicates and were used to access the consistency of ratings and detect dishonest participation. Crowd-workers could leave optional feedback in the end.

5.2.3 Results

For each participant, we reviewed score histograms and correlations among the scores of the 7 activities that were rated twice. The review suggested that 141 participants did not take the task seriously, e.g., because the correlations were very low or all scores were the same (e.g., 1s). Their data were omitted from later analyses and they were banned from participating in other data collections. We further reviewed the correlations between the scores of individual participants and the average across participants for each dimension. The review showed moderate to very strong correlations for all but 15 participants, whose scores did not correlate with the average. The data of these 15 participants were also omitted from analyses. When aggregated across participants per dimension, the correlations between individual and mean scores ranged from r = .49 for Level of Participation to r = .86 for Physical Effort, suggesting an acceptable level of consistency among participants for all dimensions. We did not use interclass correlation coefficients to describe inter-rater consistency as it is not recommended for datasets with a lot of missing data (each participant rated 55 out of 135 activities, and the rest were missing values). Cross-correlations among dimensions did not exceed r = .70, with an exception of Physical Effort and Mobility ($r(133) = .77, p < .001$). A series of regression models, with one dimension as output and all other dimensions as predictors, showed that no more than 75% of the variance in one dimension scores can be explained by other dimensions,

[1] www.microworkers.com

(R-squared = .75 for Physical Effort). This suggested that no dimension was redundant and they all were retained for further analyses.

5.3 Tagging Activities: A Data Collection

Each activity was described with a list of free-form tags, imitating many of the current recommendation systems.

5.3.1 Participants

English-speaking crowd workers (n = 185, 94 male, M age = 34.84 years, from 18 to 71 years old) were paid 85 US dollar cents for a 10-minute data collection session. Each crowd-worker could participate only once.

5.3.2 Procedure

After agreeing to the terms of the study and filling out a demographic questionnaire, crowd-workers described 10 randomly selected activities with 4 free-form keywords or short phrases. We instructed crowd-workers to “think what makes someone interested in an activity and what distinguishes it from other activities” when suggesting keywords. Optional feedback could be left at the end of the session.

5.3.3 Results

Reviewing the data showed that eleven crowd-workers did not take the task seriously. Their tags were repetitive (e.g., ’I like it’ copy-pasted for all activities), random words (e.g., ’snow’ and ’river’ to characterize playing chess), or sequences of random characters. We further reviewed individual tags, removing tags with more than three words and non-words, fixing spelling errors, and removing unnecessary words (e.g., removing ’smelling’ from ’smelling fresh air’ for Camping). This left us with 2330 tags, 79% of which were linked with only one activity. We further automatically increased this number by adding tag synonyms that Wordnet [58] indicated as similar ($sim > .75$) to the tag-linked activity. The resulting dataset contained 4223 tags, and the proportion of non-shared tags decreased down to 69%, which should have improved the performance of clustering algorithms relative to the original tag set. Obtaining 4223 tags was equivalent to the work of 212 crowd-workers.

5.4 Preference Activities: A Data Collection

Preference scores were the ground truth to evaluate the performance of dimension-based and tag-based methods against.

5.4.1 Participants

English-speaking crowd workers (n = 160, 84 male, M age = 34.52, from 18 to 72 years old) received 65 US dollar cents for a 10-minute data collection session. A crowd-worker could participate only once.

5.4.2 Procedure

After agreeing to participate in the study, a crowd-worker filled out a demographic questionnaire and proceeded to rate all 135 activities using a 7-point Liker-type item *'How much would you enjoy engaging in this leisure activity?'*. A tenth of the activities was rated twice to later assess worker trustworthiness.

5.4.3 Results

We reviewed score histograms and correlations among twice-rated activities for each participant. Overall, 19 participants were not as trustworthy as they either did not rate activities consistently with themselves or used a single score for all items (e.g., 1s for all items). We excluded their data from further analyses. Per-participants histograms also highlighted large inter-participant differences in preference patterns: some participants disliked most activities, some others liked most activities, the rest liked and disliked an equal number of activities. When averaged across participants, preference scores varied from 1.88 (Collecting Bottles) to 5.86 (Watching Movies), Figure 5.1, suggesting our sample of activities was diverse and contained both liked and disliked activities. However, no activity was universally liked or disliked, with even Collecting Bottles receiving a 7 from a participant, Figure 5.2. This emphasized that preference for an activity was a highly subjective judgment.

The mean rating distribution is described in Figure 5.3. We see that the mean ratings of the participants have a normal distribution.

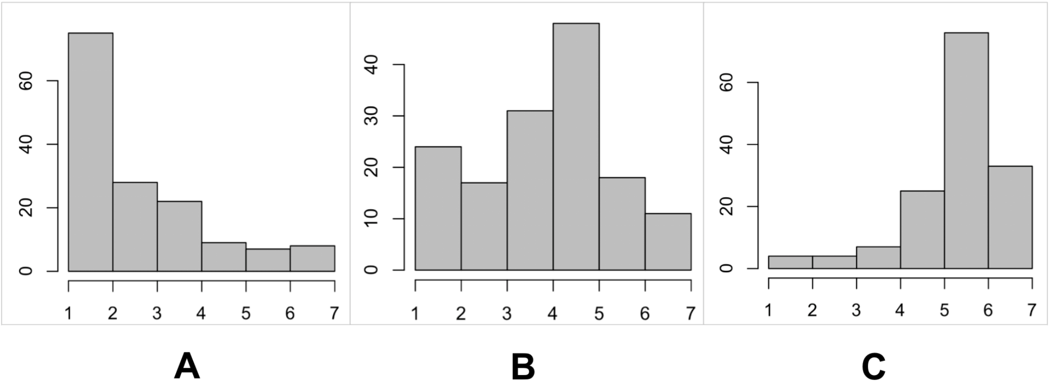


Fig. 5.1 Patterns of user preference for leisure activities, with some users either disliking (A) or liking (C) most of the activities, and others (B) liking and disliking an equal number of activities.

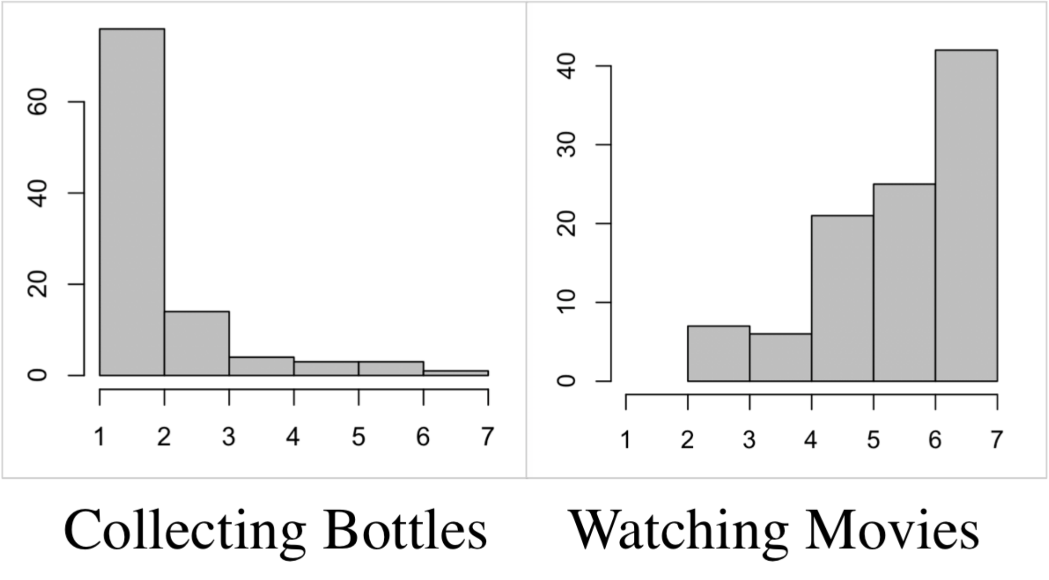


Fig. 5.2 Histograms of scores for the most disliked (Collecting Bottles) and most liked (Watching Movies) activities, showing that even these two activities were not universally liked or disliked.

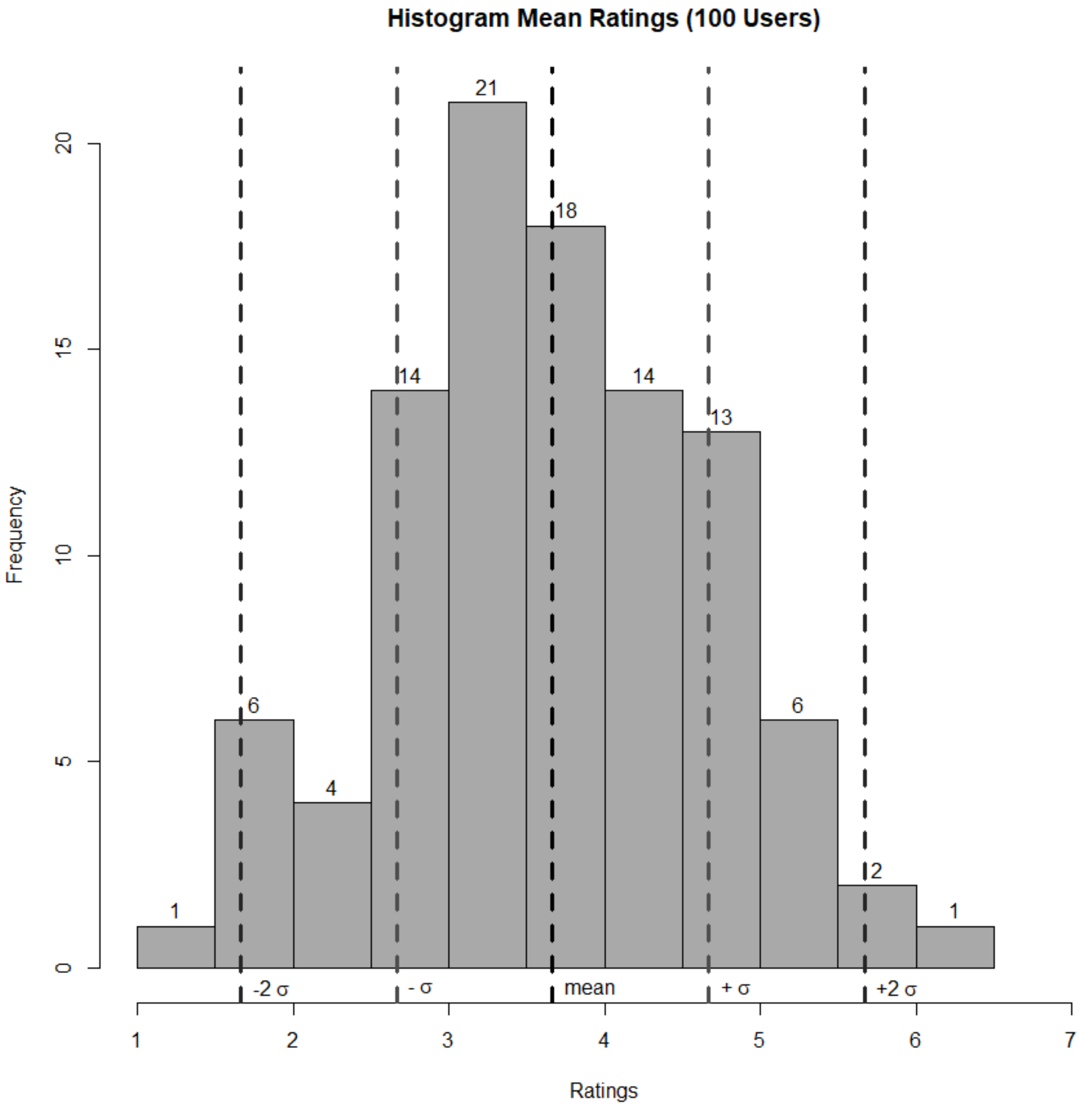


Fig. 5.3 Mean rating obtain from the data collection

5.5 Clustering of Activities

We chose to rely on Affinity Propagation [25] to automatically cluster activities. More common and well-known clustering algorithms – such as K-means [33] and agglomerate hierarchical clustering [91] – seemed less suitable for a fair comparison of tag-based and dimension-based approaches. K-means produces more meaningful results when running on the continuous numerical type of data, which would put the tag-based approach at a disadvantage since it produces sparse binary data (tag present for an activity - 1; tag absent - 0). Hierarchical clustering does not aim at ensuring all items within a cluster are similar, which the study needed, and we opted for Affinity Propagation.

Affinity Propagation required specifying several parameters. For dimensions, we chose the Pearson correlation coefficient as a similarity metric as it was well suited for comparing 17-dimension numeric vectors that represented each activity. For tags, we chose the cosine similarity metric as it was better suited than the Pearson correlation for comparing 4223 - long binary vectors that represented each activity. The input preference parameter was kept as the median similarity value for both datasets. Dimension-based clustering resulted in 16 clusters; tag-based clustering in 18 clusters. (appendix B)

5.5.1 Performance Metrics

We defined three metrics to compare the performance of dimension-based clustering against tag-based clustering: maximal spread, maximal consistency, and adjusted r-squared. The maximal spread metric described the ability of a clustering approach for finding a group of very liked and a group of very disliked activities. The larger the difference in liking – aka, spread on the liking continuum – between these two groups, the better recommendation could be, since a real-world system would have the user explicitly specify one or very few liked and disliked activities to determine which cluster is liked or disliked, and recommend or avoid all other activities from those clusters. To estimate the spread, we averaged activity-preference scores within each cluster separately for each user and measured the distance in preference between most liked and most disliked clusters.

The maximal consistency metric described the variance in preference scores within clusters. The smaller the variance, the better recommendation could be: high variance would imply a cluster contained both liked and disliked activities, which would increase the chance of the user specifying an activity as liked while the other activities in the cluster are not liked, which would lead to the system recommending those, generally-disliked activities. To estimate consistency, we calculated the standard deviation of preference scores for each cluster, and averaged it across all clusters, separately for each user.

The adjusted r-squared metric described the amount of variance in preference ratings of a user, which clusters as categorical variables could explain in a linear regression model. The higher the r-squared, the better an approach is at describing user preferences and the more precise the recommendation would be. Adjusted r-squared only increases if each additional variable – additional cluster in our case – increases the fit of the linear model by more than would be expected by chance. This allowed for a balanced comparison between the tag-based and dimension-based approach since the tag-based method produced two more clusters, which would results in a higher non-adjusted r-squared by chance.

5.6 Results

The study explored whether our set of dimensions was good at describing activities and whether the dimension-based approach could result in a better recommendation than a tag-based approach.

5.6.1 Dimensions

We correlated dimensions against activity preference, separately for each of 101 activity-preference participants. If any of the 17 dimensions had no connection to preference, it would make them irrelevant in a recommendation and they could be omitted from further analyses. Preferences for leisure was highly subjective, as expected, and some dimensions correlated with preferences for some participants, but not others. However, a visual inspection of correlations suggested that all dimensions, but Duration, defined activity preferences for at least some users, Figure 5.4. When preference scores were averaged across participants and correlated with dimensions (Table 5.1), Possible Multitasking, Temporal Structure and Fragmentation, and Level of Participation correlated the strongest with preference. However, one should not rely on such correlations with the average preference to judge a dimension as preference-unrelated due to high interpersonal differences: Figure 5.4 shows higher Sociability was significantly liked by some participants, but disliked by others, which resulted in near-zero correlation on average. We excluded Duration from further analyses and re-run clustering without it.

To test if dimension-based clustering allowed for grouping similar activities together, which was the first study objective, we calculated the three performance metrics and compared them against the same metrics for random clusters. The random clusters were of the same size as dimension-based clusters, but activities were randomly reshuffled across clusters. The random reshuffling and performance estimation was repeated 1000 times, and the

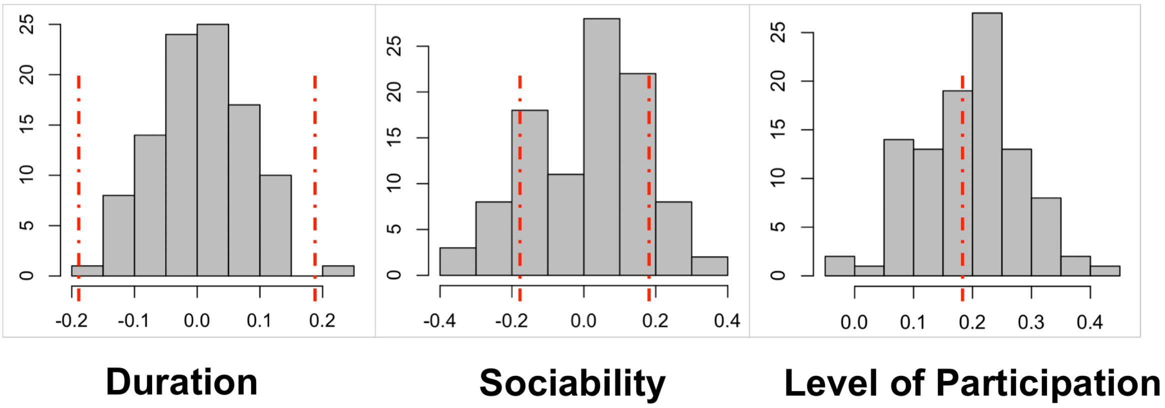


Fig. 5.4 Distributions of per-participant correlations between preference and dimensions. Dot-dashed lines show where significant correlations begin (all correlations with the magnitude $> .18$).

Table 5.1 Correlations between preference scores averaged across participants and activity dimension scores.

Dimensions	r(133)
Arrangements	-0.12
Cost	0.18*
Duration	0.01
Environment	0.34***
Level_of_participation	0.41***
Location_Specificity	0.06
Mental_effort	-0.28***
Mobility	-0.11
Motivation	0.16
Physical_effort	-0.21*
Planning_horizon	-0.22**
Possible_multitasking	0.43***
Temporal_structure_and_fragmentation	0.43***
Time_independence	-0.17*
Time_specificity	0.08
Equipment_media_dependence	-0.40***
Sociability	0.02

* $p < .05$; ** $p < .01$; *** $p < .001$.

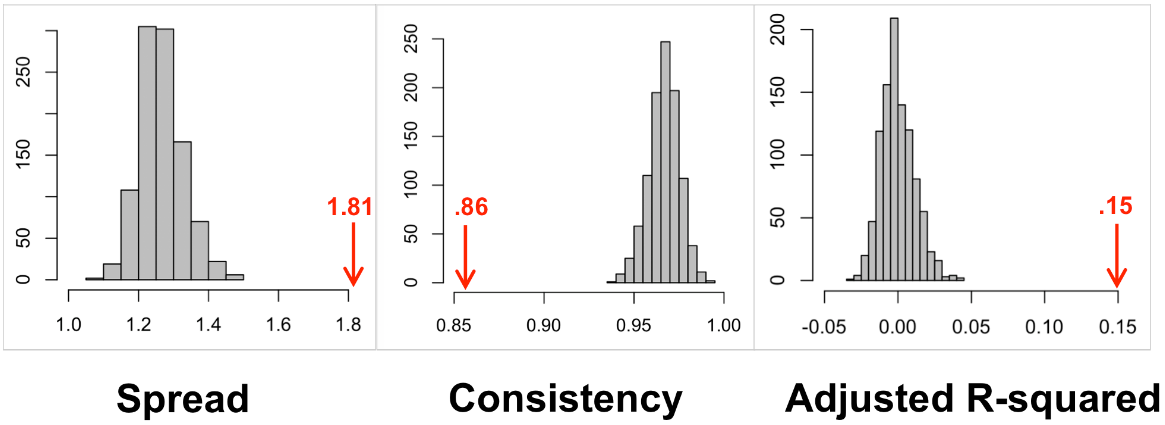


Fig. 5.5 Histograms of three performance metrics for clusters bootstrapped 1000 times. The red arrows and values show performance achieved by the actual dimension-based clusters.

2.5th and 97.5th percentiles of the resulting 1000 performance values were used as the thresholds beyond which any performance was counted as too good to have happened due to chance.Figure 5.5 shows that dimension-based clustering performed much better than chance. The last of the three metrics – adjusted R-squared – was distributed around zero since it could be negative, unlike regular R-squared.

5.6.2 Dimensions VS Tags

We could not directly compare the three performance metric estimates for the dimension-based against tag-based cluster configurations, because these cluster confirmations differed in the number of clusters and number of small clusters. Having more clusters would inadvertently improve performance: when we tried K-means clustering on dimension data with k from 2 to 30, all three performance metrics strongly correlated with k, from r(27) = 0.84 for adjusted R-squared to r(27) = 0.95 for both spread and consistency (all $p < 0.001$). Having more of smaller clusters would also inadvertently improve performance: smaller clusters tended to be further away from the middle of the 7-point scale than larger clusters ($rs(1598) = -0.15, p < 0.001$, affinity propagation clusters on dimension data for all participants), which would affect the spread metric; smaller clusters also tended to have smaller in-cluster standard deviation in preference than larger clusters ($rs(1598) = 0.13, p < 0.001$), which would affect the consistency metric.

To directly compare tag-based and dimension-based methods, we rescaled performance estimates using the 1000 performance estimates for the randomly reshuffled clusters. (The 1000-times reshuffling and performance measuring was done for tags separately from dimensions.) The rescaled performance estimates described how much a clustering method

Table 5.2 Comparison of tag-based and dimension-based method performances.

Performance metric	Mean rescaled performance		t
	Tags	Dimensions	
Maximal spread	4.43	8.67	5.88***, df = 177.73
Maximal consistency	-7.28	-12.67	-5.22***, df = 159.72
Adjusted R-squared	9.66	13.37	3.38***, df = 187.74

Table 5.3 Top 5 Tags order by the amount of use to describe activities.

Tag	Amount of Use	Description
fun	91	subjective
relaxing	57	subjective
interesting	44	subjective
social	34	objective
exciting	34	subjective

performed better than chance, and was directly comparable between tags and dimensions. A series of three t-tests – with each data point as a performance estimate for a single participant – showed that dimensions outperformed tags, Table 5.2.

The main idea with tagging is to increase the knowledge of an adaptive system related to a user by creating three-dimensional correlations between users-tags-items so that we could eventually easily implement it in recommendations systems [57, 14]. The tagging method has been popular in the recommendation system field because of several reasons, like low cognitive cost (easy-to-use), simple to implement in a system, etc [57].

We have identified two main disadvantages with the use of Tags, especially for the purpose of creating users-tags-items correlations.

Firstly, tags have some properties which are related to the specific item we are describing, so categories of tags emerge (generic tags, synonym tags, invented, etc.) [14]. In our case the Tags are activities, and as we could see in Table 5.3 the majority (4 out of 5) of the Top used tags are subjective to the user experience.

Secondly, another disadvantage comes from the mathematical analysis of the tags. We could see at Figure 5.6 that the percentage of tags used only one time is high (79%) meaning a low (about 21%) overlapping of activity tags, which means that the relational matrix of users-tags-activities will big and initially very sparse. This will affect the precision and recall of recommendation systems that use this information.

Table 5.3 shows the frequency of use of tags that describe the activities. The table shows that from the top 5 tags more used, 4 of them are subjective.

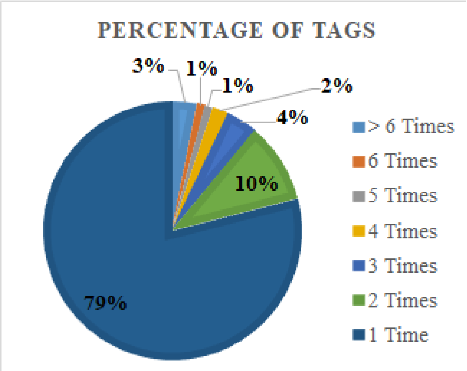


Fig. 5.6 Percentage of used tags.

5.7 Discussion

The study has explored a dimension-based method of leisure-preference description and compared it against the classical, tag-based method.

5.7.1 Describing Leisure

The set of 17 dimensions was well-suited for describing leisure. No dimension was redundant, which implied that each dimension described a different and, at least partially, independent aspect of leisure. All but one dimension correlated with leisure preferences of some participants, demonstrating the utility of dimension in describing preference and the high interpersonal diversity of leisure preference, as could be expected. Finally, dimensions allowed for grouping similar activities in clusters and these clusters allowed for recommending activities much better than random clusters (Figure 5.5). Such result has also explicitly validated the main assumption of recommendation systems – that people would like items similar to the few they explicitly marked as liked – which past recommendation-system research rarely explicitly mentioned and validated in user studies.

The dimension-based approach outperformed the classical, tag-based approach, and allowed for more accurate leisure recommendation (Table 5.2). We aimed for a fair comparison of the two approaches, trying to ensure that approximately equal amounts of crowd work were put in both datasets. The tag dataset ultimately contained data equivalent to the work of 213 people, slightly less than 230 people for the dimension dataset. However, the tag data collection lasted longer than the dimension data collection, and the actual amount of work would be larger for the tag dataset, even though it performed worse than dimension. In addition to the crowd-work, the tag dataset required a substantial amount of our time for correcting spelling mistakes and differences (e.g., American English vs British English), truncating sentences to keywords, and converting verbs to gerunds (they are forms of the same word). Such tasks could not be easily automated and highlight one more issue of tag-based approaches.

5.7.2 Implications for Leisure Recommendation

Relying on leisure-relevant dimensions to represent activities could address the issues of the tag-based recommendation methods, including the cold-start problem, excessive load on the user, and inflexibility of recommendation. First, dimension-based methods do not experience the cold-start problem because, unlike the tag-based methods, they introduce a limited shared vocabulary - dimensions - to describe both recommended items and user preferences. The tag-based methods do not restrict the vocabulary of tags and have to wait until their dataset of user-preference tags and dataset of item-descriptive tags grow large enough to have a substantial overlap between them. Newly created interactive systems often cannot afford such a long waiting period.

Second, the user would spend less effort describing their preferences using a short set of clearly formulated dimensions than using an undefined set of tags. All dimensions can be visualized - e.g., as a set of slider UI elements - and the user would relate their preference to all dimensions; whereas the entirety of possible tags cannot be visualized and the user would need to recall a multitude of tags suitable for the domain of recommendation, assess whether a tag actually describes their preference, assess whether the tag is too generic or too specific, type the tag in - which may be problematic on small-screen devices - and repeat the whole tag-search process if, after typing in, the system does not find items to recommend for the chosen tag or recommended items do not match user preference.

Finally, dimensions allow for a more flexible user preference expression than tags, particularly for non-large datasets. Using dimensions results in numeric data (e.g., the importance of leisure features for the user on a scale from 1 to 7), whereas using tags results in binary data: tag present or tag absent. A recommendation system could always find items to recommend that are most similar to the user preferences expressed as a vector of values on dimensions. However, finding relevant items to recommend is problematic if user preferences are described as a list of tags that do not match any items or match too many items. Countering such issues with tags require sophisticated algorithms and large datasets.

5.7.3 Interactive Systems

Switching from tag-based to dimension-based leisure recommendation allows for several new interaction patterns, with implications and possibilities for the interaction design of future systems. For example, dimensions would allow the user to effortlessly explore her preferences by re-adjusting the acceptable ranges of dimension values and looking through the items that a system recommends based on the acceptable ranges. Such interaction pattern would not be possible for tags, not only because tags require much mental effort to be thought of by the user, but also because they are binary (tag either present or not for an activity) and allow no graduation. A tag can characterize an activity as 'risky' or not, but it does not specify how risky.

The Dimension-based recommendation would also allow for effortless browsing through possible options. The user could specify dimensions that are crucial (e.g., low physical effort and high sociability) and sort recommended items by another dimension that is not crucial but desired (e.g., sorting by cost and exploring cheaper options first). Tags do not allow for sorting items, and they could not be easily specified and unspecified as crucial, since they often take a long time to think of and try out.

Restricting the range of possible tags could lower the effort to specify the tags that return a good match for user interests. For example, IMDb has introduced genres, which can act as tags to label movies with. This approach does restrict the vocabulary the user would use to define their movie preferences, but it loses the flexibility that the unrestricted-tag approach offered and still provides only binary data, without any information of how good a representative of the genre each movie is. For example, if a movie is labeled as action and comedy, the ratio between the two is still unclear. Dimensions would provide such information.

5.8 Contributions

This chapter main contribution is the analysis of the LAR Dimension Model, collaborating with the (**RQ1, RQ2**), and also gives the initial analysis of possible contributions of recommendation systems (**RQ3**).

The following list summarizes the contributions of this chapter:

- Datasets of Activities: We have a comprehensive group of activities that are described by tags and by dimensions and preference. The data was obtained from crowd-workers. There are three datasets: Activities described with Tags, Activities described with Dimensions and Rating preferences of users over activities.

- LAR Dimension Model Evaluation: we evaluate the Dimension Model implemented in FER. The comparison was between Tag-based model and dimension-based model.

We discussed the advantages and disadvantages of such models based on the statistical analysis done in this chapter.

Chapter 6

Evaluation of Leisure Activity Recommendations using the FER

An algorithm must be seen to be believed.
Donald Knuth

Current implementations of memory-based recommendation systems use ratings to represent the relations between users and items. These algorithms have become popular because of the simplicity of implementation and because of the development of algorithms that could estimate with great accuracy the relations between users and items.

So, it is necessary to evaluate this basic approach of recommendation algorithms and compare them with the specific content-based recommendation implemented in LAR. The recommendation algorithms and evaluation metrics are described in the FER: Precision, Recall, F1-measure, nDCG, accuracy, coverage, and transparency.

The evaluations of this chapter collaborate with the understanding of leisure activities and recommendations systems. This is done by comparing the results of the algorithms to understand how different approaches are behaving in our context.

The main focus of this chapter is the evaluation of the results and the discussion of the advantages and disadvantages of rating oriented recommendation systems and our implementation of LAR, collaborating with the **RQ1**. The metrics Precision, Recall, F1-measure, and nDCG are metrics useful for the comparison of the algorithms implemented in this thesis, collaborating with the **RQ2**. The analysis of the result and characteristics of the different algorithms collaborate with answering the **RQ3**.

6.1 The Data Collection

The dataset used for this chapter is the same described in the Chapter of evaluation of the LAR Model. Specifically, we use the following datasets:

- The list of activities: 135 Activities.
- The list of dimensions: 17 Dimensions, together with their rating over the Activities.
- The list of preferences of activities: 100 Participants rate the 135 Activities between 1 and 7 according to their preferences.

6.1.1 Activities Similarity

Additionally, to evaluate the perceived inter-activity similarity, we administered an online crowdsourcing study.

Participants

English-speaking crowdworkers (n = 480, 259 male, M age = 31.72 years, from 18 to 70 years old) received 0.9 US dollar for a 12-minute data collection session. A crowdworker was limited to participating once to decrease individual influences on the overall results.

Procedure

After accepting terms and conditions of this online study, crowdworkers filled out a demographic questionnaire and rated 113 activity pairs on their similarity using a 7-point item "*How much do these two activities have in common?*" with scale anchors "nothing at all" and "a lot". The pairs were generated randomly, but their appearance on the left or right of the screen was balanced. Out of a total of 100 unique activity pairs, 13 were rated twice for data quality control purposes.

Data

A review of twice-rated activity pairs and time-to-rate revealed 110 participants whose data could not be trusted, either because of zero rate-rerate correlation between the scores of twice-rated pairs or because ratings were given unrealistically quickly (less than 1 sec in many cases). After excluding these data, the data of the other 370 participants were aggregated across participants per each activity pair. Before aggregation, the data were scaled (converted to z-scores) separately for each participant to counter individual differences in scale use.

Most activity pairs were rated as having nothing in common at all and only a few were rated as having a lot in common, which was expected.

6.2 Implemented Algorithms

The recommendation algorithms we are evaluating are the following:

- User-based recommendation algorithm: we use the rating history of the community and personal ratings.
- Item-based recommendation algorithm: we use the rating history of the community and personal ratings.
- Model-based SVD recommendation algorithm: we use the rating history of the community with the SVD linear algebra model and personal ratings.
- Content-based recommendation algorithm: using the Activity model (activities-dimensions) with a clustering algorithm of the activities, and personal ratings.
- Hybrid-based recommendation algorithm: we use the rating history of the community, personal ratings and dimensions descriptors of the activities. Basically, we implement an User-based recommendation where the proximity between users is calculated using ratings and activity dimensions similarity.

6.3 The evaluations

We implement, describe and use the following metrics related to Information Retrieval (IR):

- Precision measures the amount of relevant selected elements over the total amount of retrieval elements, as shown in the Formula 2.1.
- Recall measures the amount of relevant selected elements over the total relevant elements, as shown in the Formula 2.2.
- F-measure measures the relations between Precision and Recall, as shown in the Formula 2.3.
- Normalized Discounted Cumulative Gain (nDCG) measures the ranking quality of a list of recommendations, as shown in the Formula 2.6.

For the metrics, we calculate the average of 3 runs to reduce any issues regarding the particular execution of the algorithms. Also, the recommendations are asked to all the users of the datasets, having the average metrics of all the recommendations.

Additionally, three metrics were relied on for comparison: accuracy, coverage and transparency. Accuracy, and Coverage (cf. [84]) required having user preferences for activities as the ground truth, whereas transparency - measured as similarity of user's previously-liked activities to the recommended activities, and dissimilarity of user's previously-disliked activities to the recommended activities - required having estimates of inter-activity similarity

6.4 Results

To understand better the results we suggest to focus the analysis in the following three main testing perspectives:

- Improvement with increasing testing dataset.
- Achieve the best value of the metric when using full dataset as a testing dataset.
- Performance differences between algorithms.

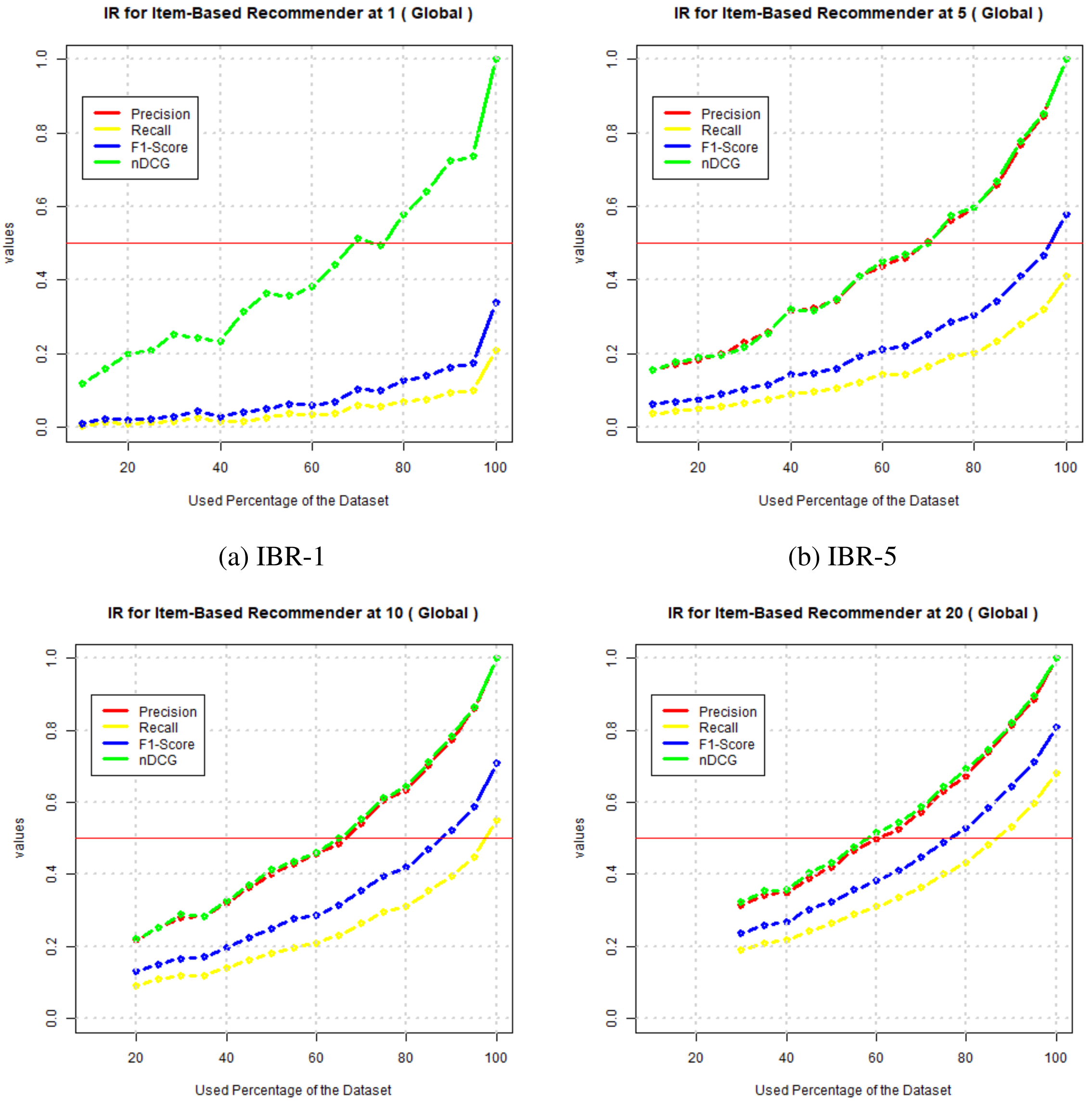
6.4.1 Item-based Recommendations

Item-based recommendations base recommendation on item similarity. Item-based and User-based are about the same approach to the problem of estimating a recommendation, just from different angles. In the basic implementation, item-based algorithms tend to be faster and can be pre-computed to increase performance. The light-weight general process of these algorithms makes them more appropriate for bigger datasets.

The obtained results with the Item-based recommendation algorithm are the following:

The Figures 6.1a, 6.1b, 6.1c, and 6.1d show the increasing tendency of the Precision, Recall, F1-Score and nDCG for the Item-based Recommendation algorithm requesting 1, 5, 10, 20 recommendation, respectively. One highlight is on the overlapping between Precision and nDCG, having these two measures overrun the Recall. Also is important to notice that the increasing request for recommendation gives a smoother tendency line, mainly affecting the improvement of the Recall with the increasing requests.

Considering that the values we are showing in these figures ideally should be 1.0 we highlight the 0.5 value to have a base reference. In this way, Figure 6.1a arrive at the base reference with a train database size of about 70% of the full dataset. Then, Figure 6.1b arrives at the base reference with a train database size of about 70% of the full dataset. Next, Figure



(a) IBR-1

(b) IBR-5

(c) IBR-10

(d) IBR-20

Fig. 6.1 Precision, Recall, F1-measure, and nDCG for the item-based Recommendation requesting (a) 1 recommendation, (b) 5, (c) 10, and (d) 20 recommendations

6.1c arrives at the base reference with a train database size of about 65% of the full dataset. Finally, Figure 6.1d arrives at the base reference with a train database size of about 60% of the full dataset.

6.4.2 User-based Recommendations

User-based recommendations are the conventional style of recommendation systems. These algorithms usually are not the fastest, considering that the size of the dataset influence a lot in the performance.

The obtained results with the User-based recommendation algorithm are the following:

The Figures 6.2a, 6.2b, 6.2c, and 6.2d also show the increasing tendency of the Precision, Recall, F1-Score and nDCG for the Item-based Recommendation algorithm requesting 1, 5, 10, 20 recommendation, respectively. One highlight is on the overlapping between Precision and nDCG, having these two measures overrun the Recall. Also is important to notice that the increasing request for recommendation gives a smoother tendency line, mainly affecting the improvement of the Recall with the increasing requests.

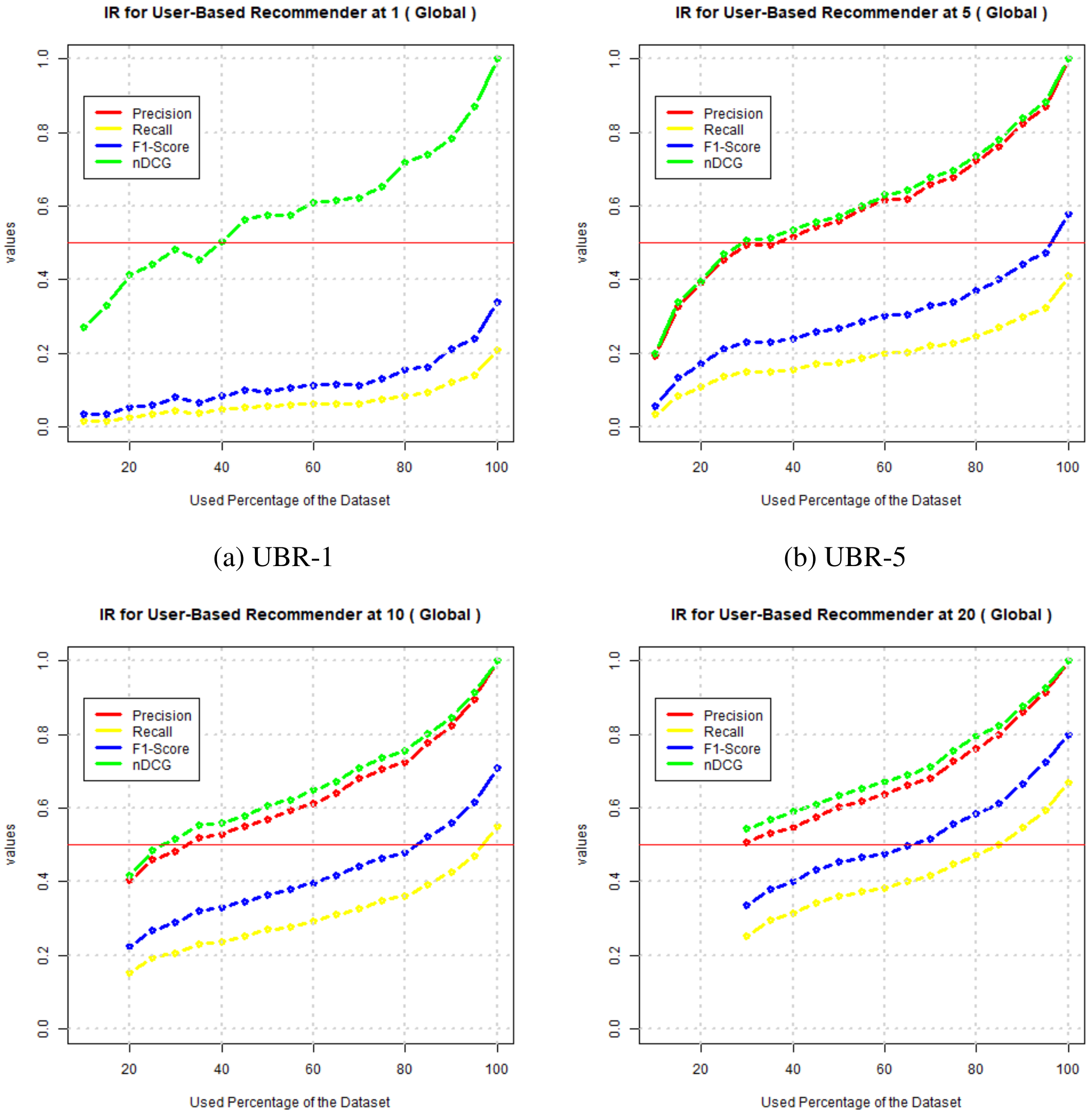
Considering that the values we are showing in these figures ideally should be 1.0 we highlight the 0.5 value to have a base reference. In this way, Figure 6.2a arrives at the base reference with a train database size of about 40% of the full dataset. Then, Figure 6.2b arrives at the base reference with a train database size of about 30% of the full dataset. Next, Figure 6.2c arrives at the base reference with a train database size of about 30% of the full dataset. Finally, Figure 6.2d arrives at the base reference with a train database size of about 30% of the full dataset.

6.4.3 SVD-based Recommendations

The SVD-based recommendation is a model-based recommendation that uses as a model the SVD matrix factorization for reducing the number of activities to consider to finally estimate the best recommendations. This type of algorithms are more complex than the basics item-based and user-based algorithms, and, lately, are very popular for achieving good performance in certain domains [7].

The obtained results with the SVD-based recommendation algorithm are the following:

The Figures 6.3a, 6.3b, 6.3c, and 6.3d also show the increasing tendency of the Precision, Recall, F1-Score and nDCG for the Item-based Recommendation algorithm requesting 1, 5, 10, 20 recommendation, respectively. Precision and nDCG are overlapping and overrunning the Recall. Also, the figures show a smoother tendency line with the increasing request for recommendation mainly affecting the improvement of the Recall.



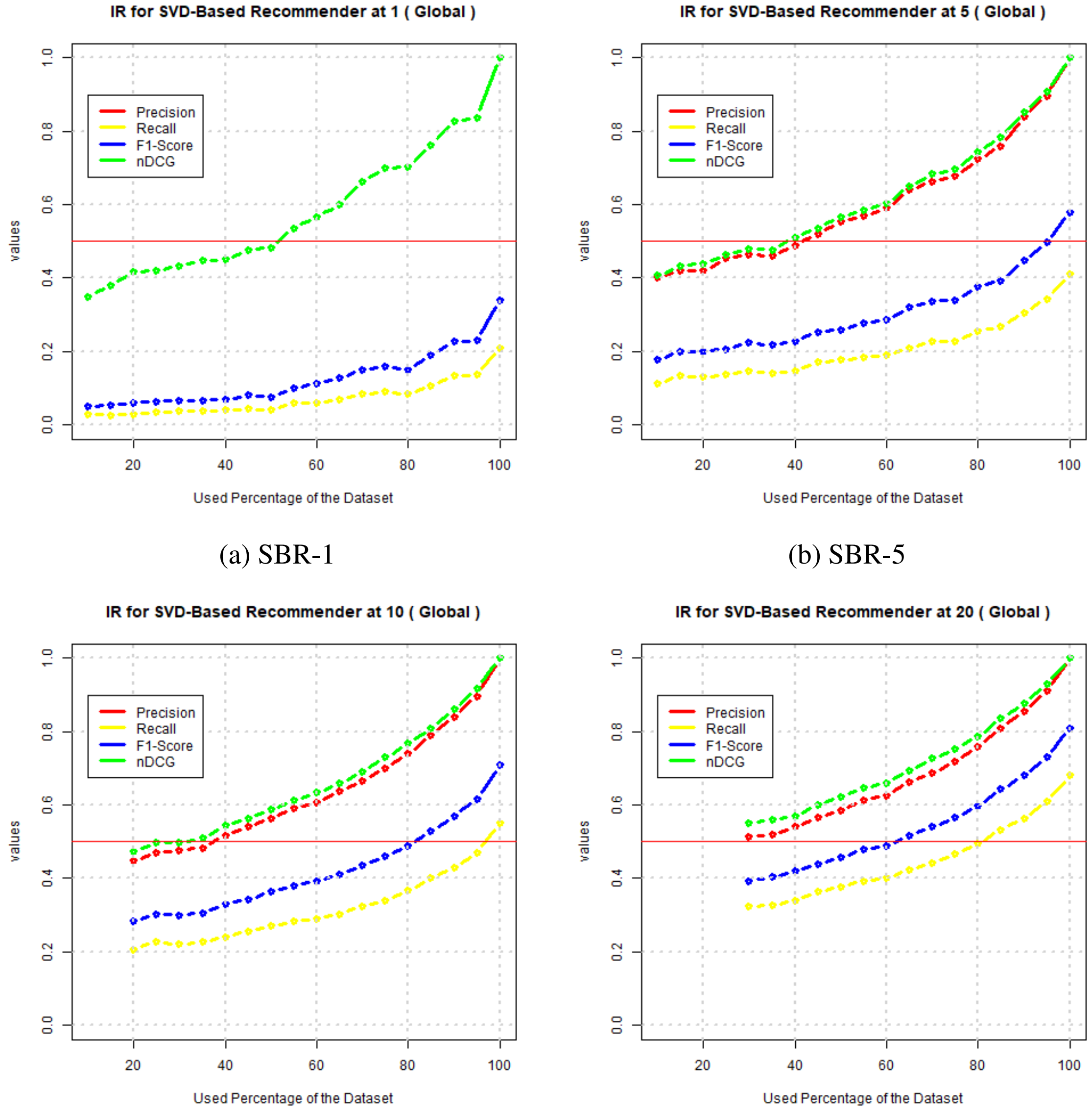
(a) UBR-1

(b) UBR-5

(c) UBR-10

(d) UBR-20

Fig. 6.2 Precision, Recall, F1-measure, and nDCG for the User-based Recommendation requesting (a) 1 recommendation, (b) 5, (c) 10, and (d) 20 recommendations



(a) SBR-1

(b) SBR-5

(c) SBR-10

(d) SBR-20

Fig. 6.3 Precision, Recall, F1-measure, and nDCG for the SVD-based Recommendation requesting (a) 1 recommendation, (b) 5, (c) 10, and (d) 20 recommendations

Considering that the values we are showing in these figures ideally should be 1.0 we highlight the 0.5 value to have a base reference. In this way, Figure 6.3a arrives at the base reference with a train database size of about 50% of the full dataset. Then, Figure 6.3b arrives at the base reference with a train database size of about 40% of the full dataset. Next, Figure 6.3c arrives at the base reference with a train database size of about 30% of the full dataset. Finally, Figure 6.3d arrives at the base reference with a train database size of about 30% of the full dataset.

6.4.4 Content-based Recommendations

Content-based recommendations is an approach searching for taking advantage of the knowledge over the items to recommend, in our case the Activities. As described in the Chapter of Analysis of LAR, we developed an alternative implementation for the recommendation of activities using the activities and dimensions. Differently, than with previous collaborative-based approaches, this approach does not require the information from other users to estimate the recommendations, it only requires the information of Activities and Dimensions with the rating of the user we want to recommend. So, in this case, the Used Percentage of the dataset is understood over the recommended user.

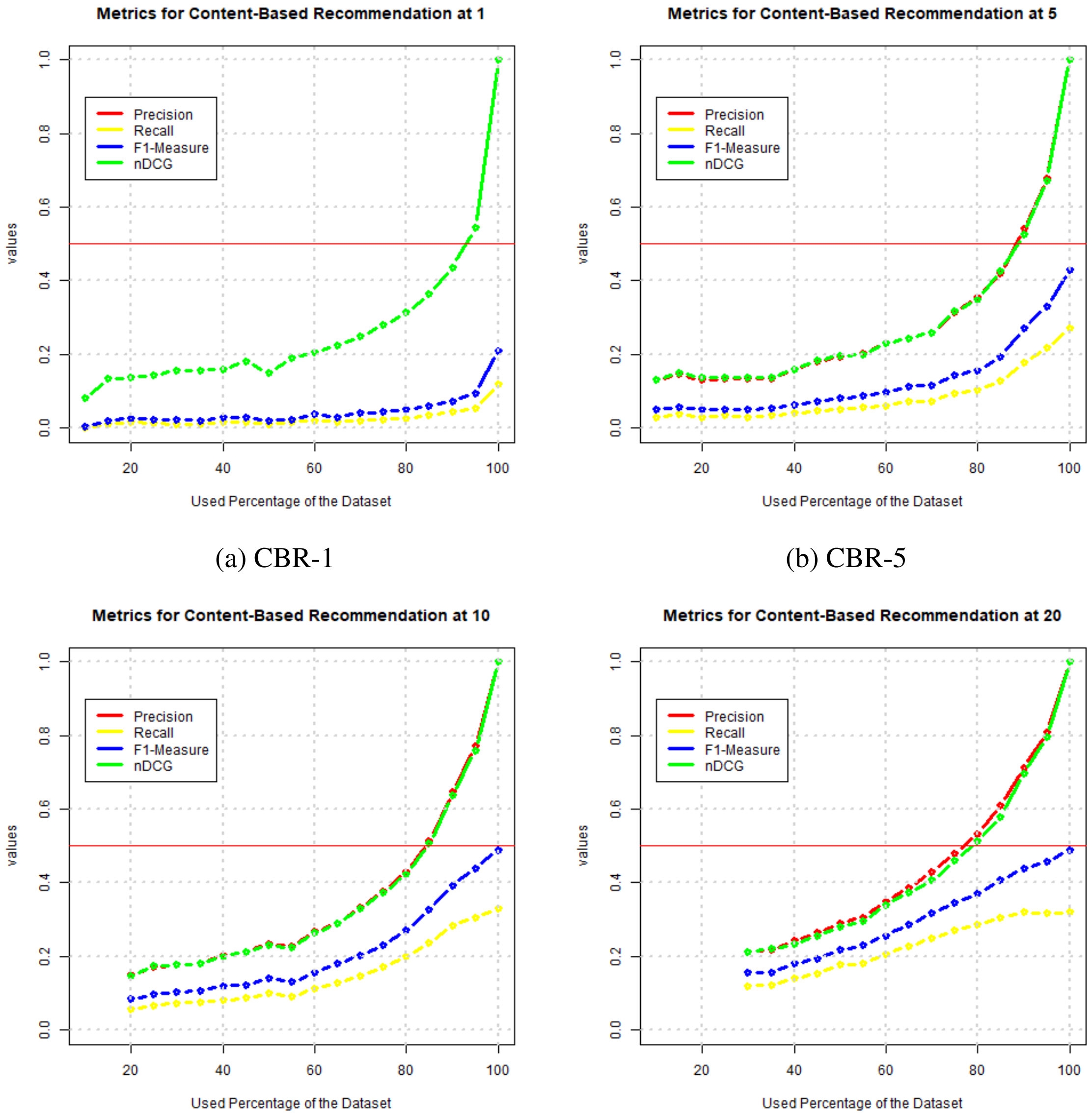
These are the results obtain processing the evaluations considering the content-based recommendation algorithms:

Similarly, the Precision, Recall, F1-Score and nDCG for Item-based recommendation algorithms for 1, 5, 10, 20 recommendations show an increasing tendency in the figures 6.4a, 6.4b, 6.4c, and 6.4d. The results related to the overlapping of Precision and nDCG and the Recall smoother tendency with the increasing amount of recommendations have been similar to the SVD-based recommendation.

Considering that the values we are showing in these figures ideally should be 1.0 we highlight the 0.5 value to have a base reference. In this way, Figure 6.4a arrives at the base reference with a train database size of about 95% of the full dataset. Then, Figure 6.4b arrives at the base reference with a train database size of about 90% of the full dataset. Next, Figure 6.4c arrives at the base reference with a train database size of about 85% of the full dataset. Finally, Figure 6.4d arrives at the base reference with a train database size of about 80% of the full dataset.

6.4.5 Hybrid-based Recommendations

Hybrid-based recommendations is a style of recommendation systems where we can combine any of the previous techniques to take advantages or mitigate the problems of the more



(a) CBR-1

(b) CBR-5

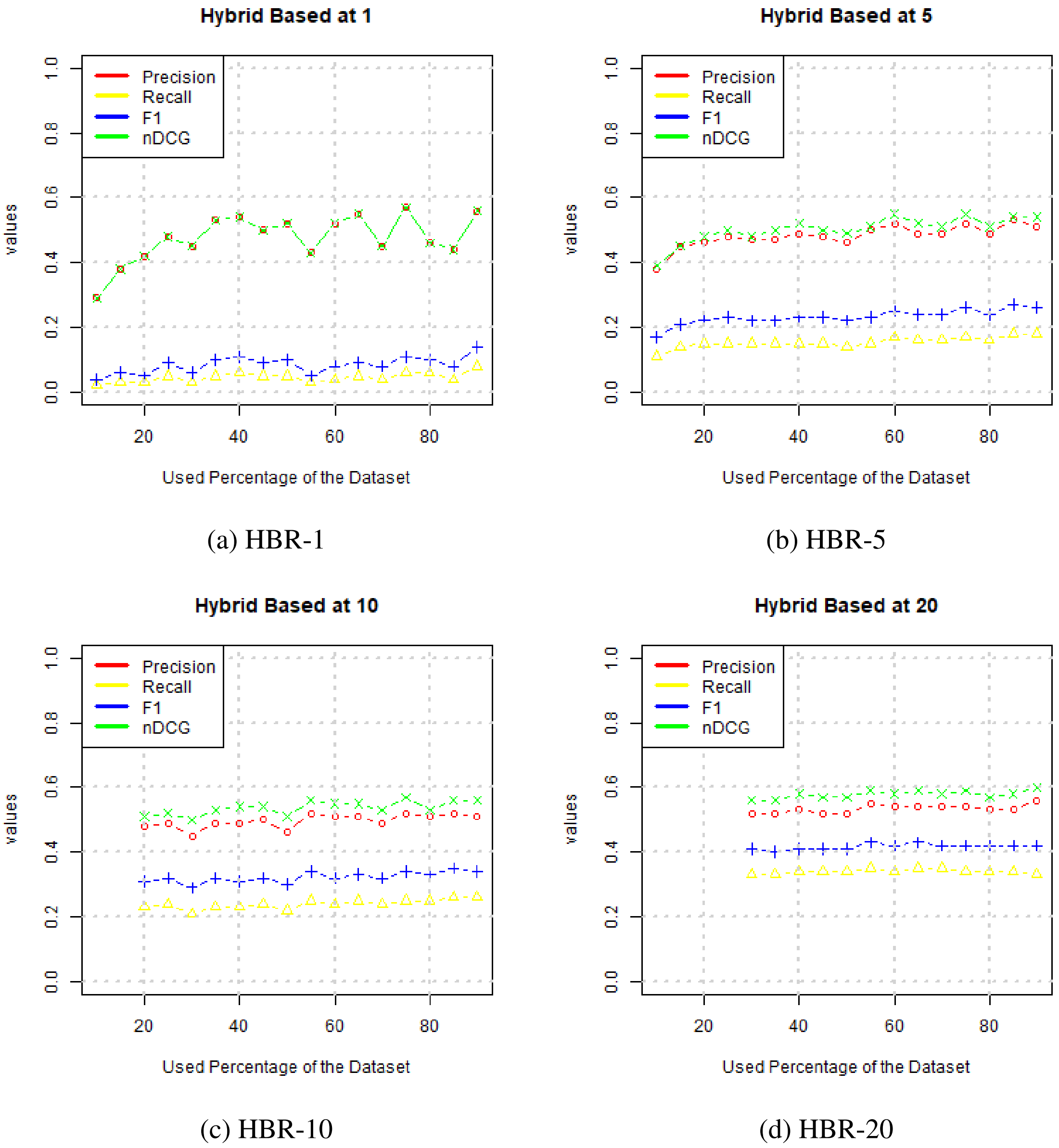
(c) CBR-10

(d) CBR-20

Fig. 6.4 Precision, Recall, F1-measure, and nDCG for the Content-based Recommendation requesting (a) 1 recommendation, (b) 5, (c) 10, and (d) 20 recommendations

traditional approaches. These algorithms usually borrow the characteristics of the underlying techniques.

The obtained results with the Hybrid-based recommendation algorithm are the following:



(a) HBR-1

(b) HBR-5

(c) HBR-10

(d) HBR-20

Fig. 6.5 Precision, Recall, F1-measure, and nDCG for the Hybrid-based Recommendation requesting (a) 1 recommendation, (b) 5, (c) 10, and (d) 20 recommendations

In this case, the Precision, Recall, F1-Score and nDCG for Item-based recommendation algorithms for 1, 5, 10, 20 recommendations show an minor increase tendency shown in the figures 6.5a, 6.5b, 6.5c, and 6.5d. The results related to the overlapping of Precision and nDCG and the Recall smoother tendency with the increasing amount of recommendations have been similar to the other results.

6.5 Additional Results

To estimate recommendation accuracy, coverage, and transparency, the dataset was split into training and testing parts, with the training part increasing from 10% to 90% of dataset sequentially with the step of 5%. The splitting was repeated seven times to average out random deviations from the true performance. For this additional results, we have considered that the Content-based approaches work by having groups of 1 activity so that we could estimate directly the distance between activities and not the average as seen with the previous results of content-based for dimensions and tags.

6.5.1 Accuracy

Accuracy was estimated as the root mean square error (RMSE) for the computed scores of the top 10% of activities recommended by an algorithm. Figure 6.6 shows both content-based methods to perform similarly and the user-based approach to outperform others. The item-based collaborative filtering approach performed poorly with smaller amounts of training data but improved rapidly with more training data.

6.5.2 Coverage

Coverage was estimated as the Gini index for the number of times an activity was in the top 10% of recommendation. The index describes the disparity among items in their chances of being recommended, with 0 correspondings to a situation when all items are recommended with the same probability, and 1 corresponding to the situation when a single item is chosen all the time. Having index values closer to 1 implies that some activities are neglected by a recommendation system. Coverage should be viewed together with accuracy estimates as it tends to increase with worse accuracy performance, as evidenced by Random recommendation having the best coverage (Figure 6.7a). The User-based recommendation had the worst coverage, while the other methods performed similarly.

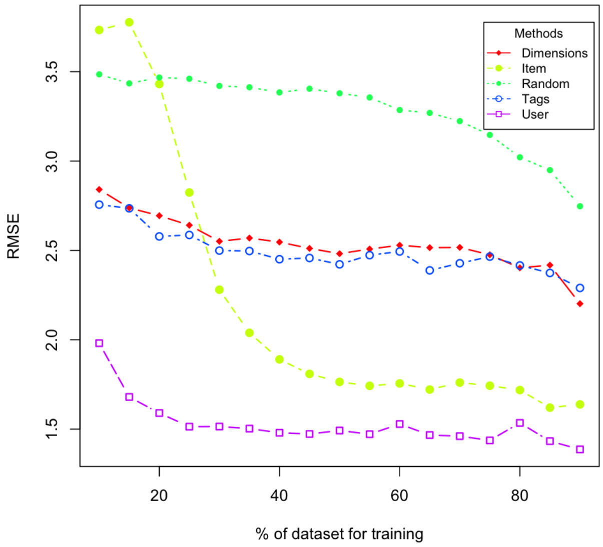


Fig. 6.6 Root mean square error estimates for the five recommendation approaches. "Random" serves as the baseline to interpret the performance of other methods. Smaller values indicate better performance.

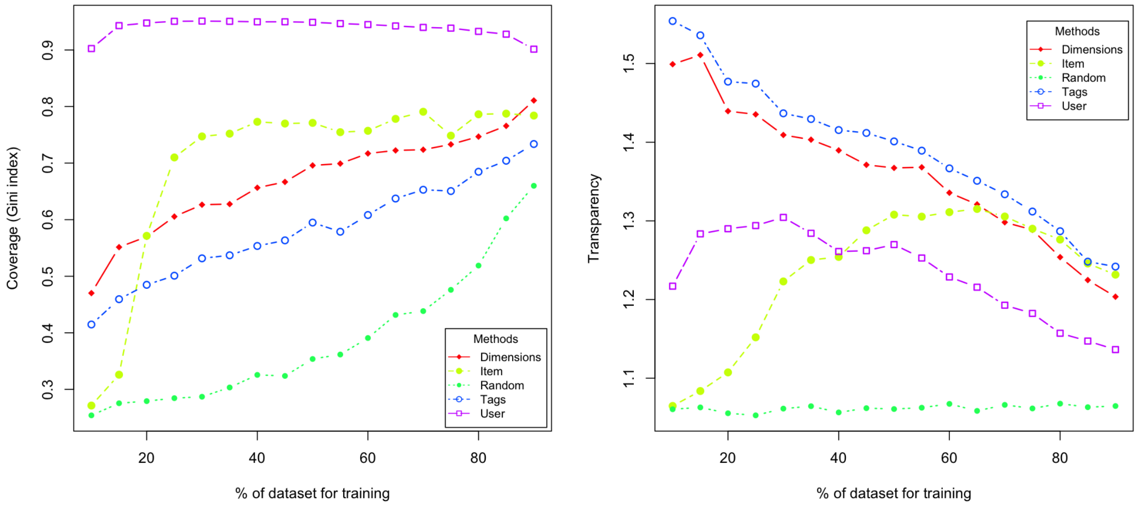
6.5.3 Transparency

Estimating transparency included two components: similarity of top 10% of recommended items to the top quarter (based on user preference scores) of the training set and similarity of top 10% of recommended items to the bottom quarter of training set. To combine the components in a single metric, similarity to the top training quarter (positive quality) was divided by the similarity to the bottom training quarter (negative quality). The Similarity was estimated the average squared pairwise similarity (each activity in top 10% recommended items paired with each activity in a top/bottom quarter of training items) using similarity scores collected in a user study. Both content-based approaches performed well, whereas the user-based collaborative filtering performed worse than other methods, but still better than a random recommendation, Figure 6.7b.

6.6 Discussion

We explored many evaluation metrics for the leisure recommendation, with no single method clearly outperforming all others on all performance criteria.

An important understanding of these measures is that the Recall value was in all cases lower than the Precision, and this is because of the selection of our relevant Items. Basically, for the relevant Items, there were selected the best N ratings of the user and if the last value



(a) Gini index estimates for five of the recommendation methods, describing each method item-space coverage. Lower values imply better coverage.

(b) Transparency estimates for the five recommendation methods, which describes how similar recommended items are what the user liked in the past. Higher values imply better transparency.

Fig. 6.7 Evaluation results for (a) Coverage and (b) Transparency

continues to appear in the next items we were also selecting them. For example, if we ask 5 recommendations, we select the 5 top ratings which could be all of value 7, but since this user could have more than 5 ratings of value 7, let say 10 ratings, all of them are considered as relevant Items. This only affects the Recall, and we call it the Arrangement problem.

To solve the Arrangement problem we just add all the values that have the same value and we cannot identify which is better. Ideally, we could request additional information to the users to solve the Arrangement problem with a better model of the rating of the activities.

Then, to study the arrangement behaviour of the algorithms we used the nDCG which models the importance of the order of the recommendations. This measurement could show better results with a model where the Arrangement problem is solved. Other arrangement metrics could be eventually be implemented.

About the F1-measure is the relation between Precision and Recall, this means that the results will show an intermediate value between them.

Considering the characteristics of the LAR, the benefits of this approach are the following:

- Privacy: This algorithm could be executed with only personal information, without the need for shared information.
- New Items: This algorithm gives a solution to possible new items in the dataset.

On the other hand, the disadvantages of the LAR are the following:

- Performance: The current implementation has poor performance values in terms of Precision, Recall, F-measure, nDCG.
- New Users: Because of the performance issues, recommendations to new users with not many evaluations of the items it will be a problem.

However, the implementation of the LAR could be improved by implementing a better model for the preferences of users over the activities to solve the Arrangement Problem. Also, other metrics for comparing the distances of groups of activities could be implemented, for example, distance to the exemplars of the clusters, or using Kendall Rank Coefficient instead of Pearson Coefficient, etc.

Finally, some hybrid approaches could also improve the LAR, by putting together collaborative filtering algorithms with the content-based approach.

6.6.1 Additional Performance Comparison

Collaborative-filtering approaches were the most accurate: User-based recommendation had consistently the lowest RMSE, whereas Item-based recommendation performed poorly with low amounts of training data, but quickly improved as the proportion of training data increasing and became close to the User-based recommendation in terms of recommendation accuracy (Figure 6.6). Despite their superior accuracy, collaborative-filtering approaches may be a sub-optimal choice, particularly if user privacy is crucial [53], e.g., when the choices of leisure are affected by user's health condition, which may be highly sensitive information. Content-based approaches use only the data of the target user - unlike collaborative-filtering approaches - and do not search through and inadvertently reveal the preferences of other users, who often belong to the social circle of the target user (cf., Facebook event suggestions). Such exclusive reliance on the target-user data makes content-based approaches better suited for the privacy-sensitive recommendation.

Content-based approaches also produced recommendations of higher transparency than collaborative-filtering approaches (Figure 6.7b). Transparency likely affects user trust in a recommendation system, which then translates in system adoption and uses - the ultimate goal of system designers[16]. While the original definitions of recommendation transparency focused on the direct measurement of user understanding of why an item was recommended [85], we focused on one key aspect of this understanding by measuring how seemingly similar recommended activities are to those previously liked and dissimilar from those previously disliked. This approach allows for estimating transparency in an offline experiment, without needing to conduct a new user study for each explored recommendation method.

Finally, content-based approaches had better item-space coverage than collaborative filtering approaches, particularly than User-based approach (Figure 6.7a). Coverage - the propensity of a recommendation method to recommend diverse items from the entirety of its item catalogue - may be particularly relevant for leisure recommendation, since the user engages in leisure at their discretion and may like to try experiencing rare, unusual or exotic activities. Such activities are rarely rated and appreciated by other users, and thus, rarely recommended by collaborative filtering approaches, which may be a drawback. The issue of coverage is closely related to the cold-start issue: a collaborative-filtering recommendation cannot competently recommend items until it has enough data for each item. We did not model the cold-start issue in this paper (a training dataset had the same amount of data for each item), but would expect content-based approaches to outperforming collaborative-filtering approaches in a cold-start scenario, because their performance improves only marginally with an increase in training data amount, unlike the performance of collaborative filtering approaches (cf., Figure 6.6, Item-based method performance increase).

6.6.2 Describing Leisure

In addition to modest accuracy, content-based approaches have other drawbacks, including needing to research ways to model a recommended item, and then, to collect data to build item models, which is costly. This research has addressed the first of these drawbacks and suggested describing leisure with 17 dimensions. This set of 17 dimensions was well-suited for describing leisure, as no dimension was redundant, which implied that each dimension described a different and, at least partially, independent aspect of leisure.

The performance of the dimension-based approach was similar to the performance of another content-based approach, which used free-form tags. Tags are a popular and domain-independent mechanism to describe and search through recommendation items (e.g., Meetup relies on tags), but they require a lot of users effort to generate and are susceptible to a sub-kind of cold-start issue, as each item needs to be labeled with multiple tags before it can be reliably compared with other items. We witnessed the high cognitive load on the user in our tag-collection study. Crowd-workers often labeled activities with overly-generic tags, such as "fun", which was used over 100 times, and took substantially longer to label 10 activities than to rate 62 activities. In addition to the crowd-work, the tag dataset required a substantial amount of our time for correcting spelling mistakes and differences (e.g., American English vs British English), truncating sentences to keywords, and converting verbs to gerunds (in case of forms of the same word). Such tasks could not be easily automated and highlight one more issue of tag-based approaches.

The use of 17 dimensions could allow future research to study different aspects of human leisure preference. For example, a study on physical health and exercise could use dimension Physical Effort to translate participants' reports on their leisure in an estimate of exercise, whereas dimensions Sociability, Mental Effort, and Level of Participation could be used for the same purposes in studies of mental health and well-being.

6.7 Contributions

The main contribution of this chapter is the evaluation of the results of different leisure activity recommendations approaches. Additionally, an important aspect of this chapter is the discussion of the advantages and disadvantages of the implemented algorithms that collaborates with the **RQ3**. For example, the SVD algorithms out-performance the other algorithms but also is the algorithms that more computational resources needed. Another example is that the user-based approach has similar performance than the SVD based alternative with a significantly lowest process time.

The main contribution regarding the evaluation of the LAR is that if we consider the requirement analysis in the recommendations systems, the privacy is a real concern, and the content-based approach is ideal because does not need collaborative information.

The metrics Precision, Recall, F1-measure, and nDCG are metrics useful for the comparison of the algorithms implemented in this thesis, collaborating with the **RQ2**. The analysis of the result and characteristics of the different algorithms collaborate with answering the **RQ3**.

The following list summarizes the contributions of this chapter:

- Comparison between LARs implementations in FER: we evaluate the different recommendation algorithms using FER and the obtained dataset described in the previous chapter. The results show that the SVD approach outperforms the other approaches, but also is the slows algorithm. The User-based approach shows the second best results, suggesting that the user collaborative information is a good approach in this context.

- Evaluation Metrics: After obtaining the results and understanding the algorithm's behaviour, we found that the metrics selected are not fair for the content-based recommendation system of leisure activities because it does not need the full rating matrix as for the other approaches.

Chapter 7

Evaluation of a User Tag Model using FER

You can design and create, and build the most wonderful place in the world. But it takes people to make the dream a reality.
Walt Disney

Clustering algorithms allow us to understand the relationships between entities, having a deeper understanding of how entities are related. The entities that we are studying are users and activities using their descriptors (tags, dimensions).

This chapter analyses the tags' representation of the preferences of the users, searching to group them by the tag descriptor from a dataset obtained from the Meetup social network. We perform older adult group clustering based on affinity to create social groups with the following clustering algorithms: Affinity Propagation, K-means, and Fuzzy K-means. The evaluation of the groups generated by clustering algorithms is based on the comparison of the created groups with existing groups in a Meetup dataset. A priori, evaluation of new groups created by different clustering algorithm can then lead social researchers to analyze the relations and distribution of data generated by the social interactions.

We classify the evaluations of these algorithms in two types: evaluation with an internal criterion, and evaluation with an external criterion. The implemented evaluations using internal criterion are execution time and quality of the groups. Further, the implemented evaluations with external criterion are the correctness of groups and the pairwise comparison.

Distances between cluster members, the density of the data space, statistical distributions are just some of the possible parameters utilized in order to evaluate the quality of either the existing or created clusters. All these performance measures, are valid clusters evaluation measures but are dependent on the used clustering algorithm.

This chapter analyzes mainly the clustering implication of users into a social network, which focused on the socialization of users to eventually meet-up in real life.

The main focus of this chapters is the analysis of the user model using clustering algorithms and a particular technological use case for social engagement like is the Meetup social network, collaborating to answer the **RQ2** and **RQ3**. Eventually, the discussions of this chapter could leverage to build a more elaborate hybrid recommendation system using the clustering model as part of a model-based recommendation algorithm that will be the initial step for reducing the size of the dataset and then evaluate the leisure activity recommendation using more traditional collaborative base approaches, giving potential contributions to **RQ1** and **RQ3**.

7.1 The Meetup Dataset

For our study, we obtained a testing dataset from the Meetup [1] social network API [54]. The Meetup is an event-based social network that facilitates hosting events in various localities around the world. Users are subscribed to Meetup mainly to organize or participate in meet-ups. Furthermore, Meetup users can create groups manually or subscribe to existing ones.

We created two different datasets (including users, groups, and related tags) based on the users of the Meetup's base group called "60+ Happy Hour":

- Meetup 1 is the dataset of the users and groups from the base group and related users and groups (all related members of the related members of the base group).
- Meetup 2 is the dataset of the users and groups from the related users and groups of the base group (only the first layer of related members of the base group).

Table 7.1 shows some general statistics of the two created datasets. Please note that in both datasets there are more groups than users: in fact in the Meetup social network service any single user can choose to participate in more than one group.

Even tho, the qualitative analysis of the groups is not the focus of this chapter, we found that most groups are oriented to be specific to a particular type of activity, that eventually is done with a certain periodicity, and in specific places. Examples are political groups that organize discussion meetings, or happy hour groups (e.g. 60+ Happy Hour) that meet after work for a drink or for dinner.

[1] https://www.meetup.com/

Table 7.1 Meetup dataset data distribution.

General Statistics	Meetup 1	Meetup 2
Number of users	2111	489
Number of tags	4340	1248
Average tags per user	26.01	22.99
Number of groups	3767	942
Average groups size	5	5

7.2 Evaluation

7.2.1 Evaluation Metrics

The *quality* of a group is defined by taking into account the parameters over which the data-point have been grouped. Since our parameters for grouping are the tags, we measure the quality by calculating the percentage of tags of the group that belongs also to the users, also called cohesion. We choose to evaluate the clusters by using internal and external criteria, namely:

- Internal criterion: this evaluation means that the parameters used in the evaluation of the cluster quality are derived from the clusters themselves, hence obtained without introducing external factors.

- External criterion: this evaluation means that the parameters used in the evaluation of the cluster quality come from the ground truth. In this case, the evaluation of the cluster quality is based on a comparison between the cluster generated and the ground truth group.

Another measure we use in our evaluation framework is "clusters user correctness": it describes the percentage of users of the algorithm's generated cluster that is present in a ground truth group. The "clusters tags correctness" percentage instead describe the percentage of tags of the generated clusters that are present in a ground truth group.

Then, the "pairwise comparison" measure is obtained by calculating the ability of the algorithm to classify pairs of instances or tags correctly. A pair of instances are classified as correct when both the elements of the pair present in a ground truth group are also present in a cluster generated. We say then that the algorithm produces 100% correct results when each instances pair contained in all the ground truth groups are also contained in the clusters generated. Regarding tags instead, the comparison logic is the same but based on cluster tags pairs. In pairwise comparison, we evaluate the algorithm using precision, recall, and f-measure ($\beta = 0.5$) evaluation measures.

The precision, in our study, shows that we are comparing the results of the algorithms and the ground truth as a classification problem of the users. So, precision means the percentage of assigned groups that correspond to the ground truth within all assigned groups. On the other hand, recall means the percentage of assigned groups that correspond to the ground truth within all groups of the ground truth.

7.2.2 Internal Evaluations

The first internal evaluation we have considered is the collection of the execution time of the three algorithms (Affinity Propagation, K-means, and Fuzzy K-means). Our results tell us that the execution time is influenced more by the number of users' tags in the dataset than the number of users for all analyzed clustering algorithms.

Then we have focused our analysis on the internal quality evaluation and we report our results in Figure 7.1 obtained using the Meetup 2 dataset. In this figure, the Y-axis group with ranges of the qualities of groups, and the X-axis shows the percentage of the cohesion of the groups within the ranges over the total amount of groups. We think that the overall internal evaluation quality does not go beyond 30% because the average overlap over the user's tags and the group's tags are around 30% in our dataset. Considering the 30% average overlap between user's tags and group's tags, the ideal result should have bigger percentages at the 20-30% range, and Fuzzy k-means is the only algorithms that achieved this range (7%), with a bigger value in the 10-20% (53%). These values give us the idea that the Fuzzy k-means algorithm provides slightly better quality results over the whole spectrum of qualities of groups.

7.2.3 External Evaluations

The correctness evaluation for clusters based on users and based on tags shown in Table 7.2 shows that Affinity propagation has better average values and compared to the rest almost always have a smaller gap between the min and max values. Similarly, the pairwise evaluation for pairs based on users and based on tags shown in Table 7.3 shows that Affinity propagation has better results when comparing pairwise tags, and when comparing with pairwise users it has a slide (less than 1%) lost against K-means.

In Table 7.2 we notice that the Correctness is better in terms of tags than in term of users. In our dataset, this is related to the fact that the tags of the clusters are calculated out of the tags of the users and not based on an explicit description of the ground truth. These results are correlated with better precision as shown in the table 7.3.

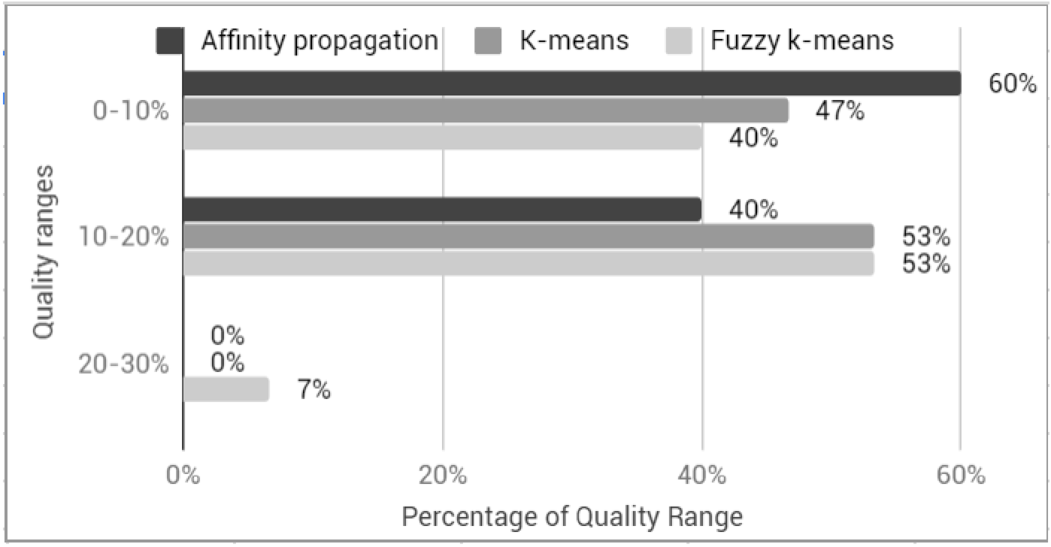


Fig. 7.1 Cluster cohesion distribution resulted by running Affinity Propagation, K-means and Fuzzy K-means algorithms on the Meetup 2 dataset.

In Table 7.3 we notice that in terms of pairwise comparison the Recall is significantly low in all cases. This relates to the fact that all algorithms are not covering well most of the relevant results. Also, we notice that the Recall using tags pairs is somehow better (but still low) than the Recall using users pairs because the tags of the groups are calculated base on the users of the groups, thus improving the precision.

Table 7.2 Results on cluster user and tags correctness made by using the external criterion for the clusters generated by the three algorithms implemented on top of the framework using the Meetup 2 dataset.

Algorithm	**Cluster users' correctness**			**Cluster tags' correctness**		
	AVG	**MAX**	**MIN**	**AVG**	**MAX**	**MIN**
Affinity propagation	56.45%	79.17%	13.64%	95%	99.36%	89.44%
K-means	52.28%	85.71%	24%	94.20%	99.76%	88.78%
Fuzzy K-means	38.50%	100%	12.50%	60%	100%	49.92%

7.2.4 Datasets comparison

The proposed evaluation framework also allows a comparison between different datasets, so we could see the similarities and differences of the clustering algorithms on a different type of datasets.

Table 7.3 Results on pairwise users and pairwise tags comparisons made by using the external criterion of the clusters generated by the three algorithms implemented on top of the framework using the Meetup 2 dataset.

Algorithm	**Pairwise users comparison**			**Pairwise tags comparison**		
	Precision	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
Affinity propagation	62.04%	3.82%	7.21%	99.38%	17.22%	29.35%
K-means	62.24%	4.13%	7.76%	99.22%	15.28%	26.49%
Fuzzy K-means	19.50%	0.01%	0.19%	80.02%	0.70%	1.39%

The results in Table 7.4 show that the precision and recall of the classification are higher in Meetup 2. Also, the recall is significantly lower in Meetup 1. In general, the low recall in all investigated algorithms means that they are missing to discover a large number of possible groups.

Analyzing Tables 7.1 and 7.4, the datasets Meetup 1 and Meetup 2, have proportionally the same data distribution. However, since the Meetup 2 dataset is composed of users coming from a common group, we expect that, by running clustering algorithms on this dataset, we would obtain higher evaluations of precision and recall. The evaluation study confirms this assumption.

Table 7.4 Average pairwise evaluation using the external criterion of the clusters generated by the three algorithms implemented for the two analyzed datasets.

Dataset	**Pairwise users comparison**			**Pairwise tags comparison**		
	Precision	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
Meetup 1	14.22%	0.38%	0.75%	95.23%	1.99%	3.90%
Meetup 2	54.76%	5.84%	10.30%	99.01%	16.19%	27.82%

7.3 Discussion

We have developed and implemented an extensible Java framework with the aim of giving developers of clustering algorithms the opportunity to implement, evaluate and compare their algorithms. The framework is designed to execute data mining algorithms on users' data. Moreover, the framework architecture allows extending to different algorithms, evaluation metrics, and different domains.

An example for the extension of the framework could be the implementation of a new clustering algorithm, which need to implement the *ClusteringService* Interface that indicates that for doing so, the functions run, *preProcessing*, *postProcessing*, *process*, *saveCircles*, *printStatistics* and *deletePrecedentResults()* as shown in Chapter 3. Additionally, the existing *Statistic*, *DistanceMeasure*, *UserProfile* and *Circle* DAO classes are available with their corresponding Factories, having the possibility to simply extend any of these elements with a new version.

To test the framework we have used two datasets obtained from a specific use case related to older adults in the Meetup social network web site. We thus used the proposed evaluation framework to compute and compare a number of quality metrics using three state of the art clustering algorithms. The preliminary results highlighted the ability of the framework to produce comparable quality measures and algorithms evaluations. Moreover, the framework structure gives us the possibility to execute, evaluate and compare the selected (and implemented) algorithms, also by allowing to quickly change their execution parameters. In fact, we were able to draw some conclusions about the different algorithms behavior and their results for the specific datasets.

Considering that the Meetup dataset is a real representation of users that describe their preferences with tags, we can understand how good is the representation of user preferences with Tags. Especially, because Meetup social network is oriented to support socialization for eventual face to face meetup of users.

For instance, our evaluation framework helped us to understand, that the Affinity Propagation algorithm provides better results if we analyze its quality performances using an external criterion instead of an internal one.

7.4 Contributions

The main contribution of this chapter is the user model analysis of activities within a real dataset, contributing with the **RQ2** and **RQ3**. This is done by comparing the performance metrics and discussion of the results of this tag based model. The results show that Affinity Propagation algorithms provide better results if we analyze its quality performances using an external criterion instead of an internal one.

On the other hand, the discussion provide in this chapter could leverage to build a more elaborate hybrid recommendation system using the clustering model as part of a model-based recommendation algorithm that will be the initial step for reducing the size of the dataset and then evaluate the leisure activity recommendation using more traditional collaborative base approaches, giving potential contributions to **RQ1** and **RQ3**.

The following list summarizes the contributions of this chapter:

- User Tag Model Evaluation: we evaluate the user tag model with the datasets obtained from the Meetup social network. The evaluation metrics were the internal evaluation with cluster cohesion and the external evaluation using the external dataset of Meetup.

- Clustering Analysis: using the proposed FER system, we have made a detailed study of the comparison of clustering algorithms with the groups of users in a real social network. We found that for our specific dataset (obtained from the Meetup web site) Affinity Propagation algorithm outperforms the other approaches implemented in FER (namely K-Means and Fuzzy K-Means).

Chapter 8

Conclusion and Future Work

To succeed, jump as quickly at opportunities as you do at conclusions.
Benjamin Franklin

This chapter presents the collection of contributions and conclusions obtained with the work done in this thesis. We will also present the limitations and possible future works to understand better the boundaries of the thesis and the possibilities for increasing and improving the understanding of leisure activity recommendations systems.

The main contribution of the FER systems is the definition of an evaluation environment based on the current best practices in terms of development and capability to support the evaluation and comparison of clustering and recommendation algorithms in a systematic way.

It is important to keep in mind that the context of our problem is related to two aspects: recommending leisure activities to people, and recommendation systems in social network sites. So, our problem is in the middle of the social problem of leisure activity selection and the technological problem of using a recommendation system for recommending items to Users.

The following three Research Questions (RQ) of this Thesis influence the Chapters as described in Table 8.1.

RQ1. Can a feasible data model be developed for a Leisure Activities Recommendation System that represents users, leisure activities, and users' preferences?

RQ2. What are the most appropriate metrics to compare recommendations of leisure activities?

RQ3. What are the most performing recommendations system approaches to meet end users preferences?

Table 8.1 Chapters influence levels related to the Research Questions.

Chapters	**Research Questions**		
	RQ1	RQ2	RQ3
Chapter 2. Recommendation Systems. Background	medium	low	low
Chapter 3. Design and Development of FER	medium	low	low
Chapter 4. Design of a LAR	high	high	high
Chapter 5. Evaluation of a LAR Dimension Model	medium	high	high
Chapter 6. Evaluation of LARs using FER	medium	high	high
Chapter 7. Evaluation of a User Tag Model using FER	medium	high	medium

8.1 Contributions

Considering the **RQ1**, the Chapter 2 with the analysis of existing recommendations systems, the Chapter 3 with the developed evaluation framework, and especially with the discussion of the implications of Leisure Activity Recommendations, and Chapter 4 have been focused on the possibilities to implement a recommendation system of users, activities and their preference relations. Additionally the Chapters 5, 6 and 7 show evaluation examples of such systems.

Then, for the **RQ2**, Chapters 2 and 3 present possible evaluations for recommendations systems and Chapter 4 present the analysis of such implementations. Also, some of the evaluations and metrics were measured and compared in Chapters 5, 6, 7.

Finally, for the **RQ3**, the Chapter 4, developed the main discussion for understanding the leisure activity recommendation systems that have been tested in Chapter 5 with clustering evaluation, in Chapter 6 with a proposed model of Activities, and in Chapter 7 with performance evaluation of recommendations algorithms. This Chapter summarize the contributions and limitations of our study of selecting leisure activities and recommendations systems.

As mentioned in Chapter 1, the Introduction of the Thesis, we follow a software engineering approach, so we divide the contributions into the four stages of this thesis: Requirements, Design, Implementation, and Testing and group them in the following two groups:

- Requirements and Design.
- Implementation and Testing.

For the Requirements and Design of this Thesis, the main contributions are the following:

- **Recommendation systems**: This chapter gives the theoretical background for understanding the implementation and evaluation of recommendation systems. Basically presents the four typical techniques for recommendations systems: content-based filtering, collaborative filtering, hybrid filtering, and clustering systems. The clustering systems approach is proposed to eventually be used as part of a model-based or content-based approach. Also, some evaluation metrics are described. Finally, some examples of implementation are presented.

- **User Requirements**: the social understanding of the activities is an important factor of this thesis, that is why chapter 2 presents the analysis of perceptions of a specific domain of users: older adults. This will allow a better understanding of the user, and eventually, critique better the benefits and problems of possible recommendation systems implementations.

- **Evaluation Framework**: the main contribution of this chapter is to provide a playground for comparing recommendations algorithms. Basically, we present the three implemented clustering algorithms: K-Means, Fuzzy K-Means, and Affinity Propagation. Also, presents the four implemented recommendation algorithms: item-based recommendation, user-based-recommendation, SVD recommendation (model-based), and content-based recommendation. For the evaluation of the clustering we present: internal criteria implementation, we present some evaluation metrics: precision, recall, f1-measure, nDCG, RMSE.

- **Framework Extensibility**: the different characteristics of the framework (interfaces, abstract classes, design patterns) allow the possibility to extend the framework to other algorithms, metrics, or even domains.

- **Framework Scalability**: The implemented algorithms have the possibility for scalability to perform big-data executions because of the use of mahout libraries.

Finally, for the Implementation and Testing of this Thesis, the main contributions are the following:

- Activity dimension Model: the dimension based activity model that describe the activities from characteristics closely related to leisure activities like physical effort, mental effort, environment, duration, time independence, planning horizon, time specificity, temporal structure, possible multitasking, sociability, level of participation.

- Standardization: The proposed activity dimension model could be an initial idea for the standardization of leisure activities.

- Content-based Leisure Activity Recommendation: a novel content-based recommendation system has been proposed in Chapter 4. The theoretical advantages are the following: a better understanding of leisure activities, low influence on the cold-start problem, and could have a good performance time-wise (if clustering is performed in batch).

- Recommendation system analysis: based on the current state of the art of recommendation systems we discuss the alternatives and analysis of techniques and approaches to implement leisure activity recommendations systems.

- Datasets of Activities: We have a comprehensive group of activities that are described by tags and by dimensions and preference. The data was obtained from crowd-workers. There are three datasets: Activities described with Tags, Activities described with Dimensions and Rating preferences of users over activities.

- LAR Dimension Model Evaluation: we evaluate the Dimension Model implemented in FER. The comparison was between Tag-based model and dimension-based model. We discussed the advantages and disadvantages of such models based on the statistical analysis done in chapter 5.

- Comparison between LARs implementations in FER: we evaluate the different recommendation algorithms using FER and the obtained dataset. The results show that the SVD approach outperforms the other approaches, but also is the slows algorithm. The User-based approach shows the second best results, suggesting that the user collaborative information is a good approach in this context.

- Evaluation Metrics: After obtaining the results and understanding the algorithm's behaviour, we found that the metrics selected are not fair for the content-based recommendation system of leisure activities because it does not need the full rating matrix as for the other approaches.

- User Tag Model Evaluation: we evaluate the user tag model with the datasets obtained from the Meetup social network. The evaluation metrics were the internal evaluation with cluster cohesion and the external evaluation using the external dataset of Meetup.

- Clustering Analysis: using the proposed FER system, we have made a detailed study of the comparison of clustering algorithms with the groups of users in a real social network. We found that for our specific dataset (obtained from the Meetup web site) Affinity Propagation algorithm outperforms the other approaches implemented in FER (namely K-Means and Fuzzy K-Means).

8.2 Limitations

We divide the limitation of this thesis with respect to the four stages presented in section 1.4: Requirements, Design, Implementation, and Testing.

The Requirement and Design stages, where we studied the problem from the leisure activity perspective and the recommendation systems perspective, has the following challenges and limitations:

- Activity model: The models to explain the human behaviour and activity selection are complex, making it difficult to implement a comprehensive model that also is simple enough to be implemented in a real recommendation system. For example, for the selection of activities, internal human factors like motivation and personality are part of the behavioural model, and we did not represent this in our model.

- Availability of Information: The information we have analyzed is not easily obtained, especially if we want to test algorithms meant to be executed on a big scale of data.

- Privacy: The increasing requirement of privacy in the technological environment is a big challenge to machine learning techniques. So the collaborative filtering approaches should be considered with particular care.

Then, for the Implementation and Testing stages, we have the following challenges and limitations:

- Standardization of Activities: Our work proposes a way to describe the activities, but the state of the art on leisure activities has no standard definition of activities.

- Human-Computer Interaction: The implementation has been done minimizing the human-computer interaction considerations.

- Clustering of Users: the clustering approaches are not integrated into the activity recommendation into the FER system.

- Framework comparison: we did not execute comparison testing of our framework with other existing frameworks to understand the efficiency and time consumption of the algorithms.

8.3 Future work

The main future works are related to the study of the activity model. Future research may rely on 17 dimensions to explore various aspects of leisure preference. For example, we might expect to see a mismatch between the activities the user would like to engage in and activities the user actually engages in, e.g., due to a higher level of physical exercise being socially acceptable, but actually disliked by the majority of users (Table 5.1).

Future research might also explore the differences in leisure preferences among demographics e.g., studying how leisure preferences change with age by testing the link between liking activities and amount of physical effort that activities require.

Additionally, more collection of data for enlarging the datasets could be done to test the execution in a parallel context. This collection together with preferences relations of the proposed dimensions could eventually serve as a user's model for proposing model-based collaborative filtering.

Also, we can explore whether our activity clusters (appendix B) appear meaningful and intuitive to the user. We might speculate that they do not fully describe the clusters, as we ourselves struggled to name some clusters and some activities seemed out of place in a cluster. However, this would not imply that activities are dissimilar in a cluster, but that a human cannot think of all 17 dimensions at once and looks for similarity along one or few dimensions.

For the finalization of the development of the LAR, it is needed to implement a user interface for the direct evaluation of the recommendation system from users, together with a user study to understand the implications of the implementation and use of such technologies.

Regarding the evaluation metrics, possible extensions could be related to the time-related metrics and the complexity analysis of the algorithms, analyzing also the sizes of the input data.

Finally, another version of hybrid recommendation systems using a clustering model could be implemented in our domain of leisure activity recommendation systems.

References

[1] ACANTO Consortium (2015). ACANTO: A CyberphysicAl social NeTwOrk using robot friends. Founded by the European Union under grant agreement Nro. 643644. http://www.ict-acanto.eu/acanto.

[2] Adams, K. B., Leibbrandt, S., and Moon, H. (2011). A critical review of the literature on social and leisure activity and wellbeing in later life. *Ageing & Society*, 31(4):683–712.

[3] Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge & Data Engineering*, 17(6):734–749.

[4] Apache Mahout (2018a). Apache mahout. https://mahout.apache.org/. Accessed: 2018-03-13.

[5] Apache Mahout (2018b). Fuzzy k-means. https://mahout.apache.org/users/clustering/fuzzy-k-means.html. Accessed: 2018-03-13.

[6] Bellotti, V., Begole, B., Chi, E. H., Ducheneaut, N., Fang, J., Isaacs, E., King, T., Newman, M. W., Partridge, K., Price, B., Rasmussen, P., Roberts, M., Schiano, D. J., and Walendowski, A. (2008). Activity-based serendipitous recommendations with the magitti mobile leisure guide. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 1157–1166, New York, NY, USA. ACM.

[7] Bennett, J., Lanning, S., et al. (2007). The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. New York, NY, USA.

[8] Boratto, L. and Carta, S. (2011). *State-of-the-Art in Group Recommendation and New Approaches for Automatic Identification of Groups*, pages 1–20. Springer Berlin Heidelberg, Berlin, Heidelberg.

[9] Boyd, D. M. and Ellison, N. B. (2007). Social network sites: Definition, history, and scholarship. *Journal of computer-mediated Communication*, 13(1):210–230.

[10] Brooks, G. A., Butte, N. F., Rand, W. M., Flatt, J.-P., and Caballero, B. (2004). Chronicle of the institute of medicine physical activity recommendation: how a physical activity recommendation came to be among dietary recommendations. *The American journal of clinical nutrition*, 79(5):921S–930S.

[11] Brucker, P. (1978). On the complexity of clustering problems. In Henn, R., Korte, B., and Oettli, W., editors, *Optimization and Operations Research*, pages 45–54, Berlin, Heidelberg. Springer Berlin Heidelberg.

[12] Cantador, I., Bellogín, A., and Vallet, D. (2010). Content-based recommendation in social tagging systems. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 237–240, New York, NY, USA. ACM.

[13] Cantador, I. and Castells, P. (2012). *Group Recommender Systems: New Perspectives in the Social Web*, pages 139–157. Springer Berlin Heidelberg, Berlin, Heidelberg.

[14] Carmagnola, F., Cena, F., Cortassa, O., Gena, C., and Torre, I. (2007). Towards a tag-based user model: How can user model benefit from tags? In *International Conference on User Modeling*, pages 445–449. Springer.

[15] Caspersen, C. J., Powell, K. E., and Christenson, G. M. (1985). Physical activity, exercise, and physical fitness: definitions and distinctions for health-related research. *Public health reports*, 100(2):126.

[16] Cramer, H., Evers, V., Ramlal, S., van Someren, M., Rutledge, L., Stash, N., Aroyo, L., and Wielinga, B. (2008). The effects of transparency on trust in and acceptance of a content-based art recommender. *User Modeling and User-Adapted Interaction*, 18(5):455.

[17] Csikszentmihalyi, M. and Csikszentmihalyi, I. (1975). *Beyond boredom and anxiety*, volume 721. Jossey-Bass San Francisco.

[18] De Angeli, A., Paladino, M. P., Coventry, L., Targher, S., and McNeill, A. (2016). Deliverable 1.3 (1.2.2) motivation and persuasion report.

[19] Del Vasto-Terrientes, L., Valls, A., Zielniewicz, P., and Borras, J. (2016). A hierarchical multi-criteria sorting approach for recommender systems. *Journal of Intelligent Information Systems*, 46(2):313–346.

[20] documentation, A. M. (2018). Apache mahout. http://mahout.apache.org/. Accessed: 2018-03-13.

[21] Drewnowski, A. and Evans, W. J. (2001). Nutrition, physical activity, and quality of life in older adults: summary. *The Journals of Gerontology Series A: Biological Sciences and Medical Sciences*, 56(suppl_2):89–94.

[22] Eibe, F., Hall, M., and Witten, I. (2016). The weka workbench. online appendix for" data mining: Practical machine learning tools and techniques. *Morgan Kaufmann*.

[23] Everard, K. M., Lach, H. W., Fisher, E. B., and Baum, M. C. (2000). Relationship of activity and social support to the functional health of older adults. *The Journals of Gerontology Series B: Psychological Sciences and Social Sciences*, 55(4):S208–S212.

[24] Fleder, D. M. and Hosanagar, K. (2007). Recommender systems and their impact on sales diversity. In *Proceedings of the 8th ACM conference on Electronic commerce*, pages 192–199. ACM.

[25] Frey, B. J. and Dueck, D. (2007). Clustering by passing messages between data points. *science*, 315(5814):972–976.

[26] Fried, L. P., Bandeen-Roche, K., Williamson, J. D., Prasada-Rao, P., Chee, E., Tepper, S., and Rubin, G. S. (1996). Functional decline in older adults: expanding methods of ascertainment. *The Journals of Gerontology Series A: Biological Sciences and Medical Sciences*, 51(5):M206–M214.

[27] Friedman, A., Knijnenburg, B. P., Vanhecke, K., Martens, L., and Berkovsky, S. (2015). *Privacy Aspects of Recommender Systems*, pages 649–688. Springer US, Boston, MA.

[28] Gibson, L., Moncur, W., Forbes, P., Arnott, J., Martin, C., and Bhachu, A. S. (2010). Designing social networking sites for older adults. In *Proceedings of the 24th BCS Interaction Specialist Group Conference*, pages 186–194. British Computer Society.

[29] Gong, S., Ye, H., and Dai, Y. (2009). Combining singular value decomposition and item-based recommender in collaborative filtering. In *Knowledge Discovery and Data Mining, 2009. WKDD 2009. Second International Workshop on*, pages 769–772. IEEE.

[30] Guan, R., Shi, X., Marchese, M., Yang, C., and Liang, Y. (2011). Text clustering with seeds affinity propagation. *IEEE Transactions on Knowledge and Data Engineering*, 23(4):627–637.

[31] Gunawardana, A. and Shani, G. (2015). Evaluating recommender systems. In *Recommender systems handbook*, pages 265–308. Springer.

[32] Guo, G., Zhang, J., Sun, Z., and Yorke-Smith, N. (2015). Librec: A java library for recommender systems. In *UMAP Workshops*, volume 4.

[33] Hartigan, J. A. and Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108.

[34] Health, N. and (NHATS), A. T. S. (2018). Dataset of favourite activities of older adults.

[35] Herlocker, J. L., Konstan, J. A., and Riedl, J. (2000). Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 241–250. ACM.

[36] Hope, A., Schwaba, T., and Piper, A. M. (2014). Understanding digital and material social communications for older adults. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3903–3912. ACM.

[37] Ilijašić, L. (2018). Java affinity propagation library. https://github.com/lovro-i/apro. Accessed: 2018-03-13.

[38] Isinkaye, F., Folajimi, Y., and Ojokoh, B. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, 16(3):261–273.

[39] Jameson, A. and Smyth, B. (2007). Recommendation to groups. In *The adaptive web*, pages 596–627. Springer.

[40] Jorro-Aragoneses, J. L., Diaz Agudo, M. B., and Recio Garcia, J. A. (2018). Madrid live: A context-aware recomendar system of leisure plans. In *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, volume 2017-November, pages 796–801.

[41] Kim, K.-j. and Ahn, H. (2008). A recommender system using ga k-means clustering in an online shopping market. *Expert systems with applications*, 34(2):1200–1209.

[42] King, A. C., Rejeski, W. J., and Buchner, D. M. (1998). Physical activity interventions targeting older adults: A critical review and recommendations. *American journal of preventive medicine*, 15(4):316–333.

[43] Kompan, M. and Bielikova, M. (2014). Group recommendations: survey and perspectives. *Computing and Informatics*, 33(2):446–476.

[44] Konstas, I., Stathopoulos, V., and Jose, J. M. (2009). On social networks and collaborative recommendation. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 195–202. ACM.

[45] Lazar, A. and Nguyen, D. H. (2017). Successful leisure in independent living communities: Understanding older adults' motivations to engage in leisure activities. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 7042–7056. ACM.

[46] Lissoni, D. (2017). Clustering Algorithms and Recommender Systems Analysis. A Comparative Java Oriented Framework. Master's thesis, Department of Information Engineering and Computer Science. University of Trento, Trento, Italy.

[47] London, M., Crandall, R., and Fitzgibbons, D. (1977). The psychological structure of leisure: Activities, needs, people. *Journal of Leisure Research*, 9(4):252–263.

[48] Longino Jr, C. F. and Kart, C. S. (1982). Explicating activity theory: A formal replication. *Journal of Gerontology*, 37(6):713–722.

[49] Lu, J., Wu, D., Mao, M., Wang, W., and Zhang, G. (2015). Recommender system application developments: a survey. *Decision Support Systems*, 74:12–32.

[50] Marchese, M., Rodas Britez, M. D., Ramos, I., and Brauchoff, I. (2017a). D4.6. User communities' creations based on user's profile matching (dynamic and adaptive profile). ACANTO Project Deliverable, University of Trento and ATOS.

[51] Marchese, M., Rodas Britez, M. D., Ramos, I., and Brauchoff, I. (2017b). Deliverable 4.2. User Profile Repository (Final). ACANTO Project Deliverable, University of Trento and ATOS.

[52] Maus, A. N. and Atwood, A. K. (2015). Surveying older adults about a recommender system for a digital library. In *Proceedings of the 20th International Conference on Intelligent User Interfaces Companion*, pages 41–44. ACM.

[53] McSherry, F. and Mironov, I. (2009). Differentially private recommender systems: Building privacy into the netflix prize contenders. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 627–636, New York, NY, USA. ACM.

[54] Meetup Inc. (2018). Meetup api. https://www.meetup.com/meetup_api/. Accessed: 2018-03-13.

[55] Meilă, M. (2007). Comparing clusterings—an information based distance. *Journal of multivariate analysis*, 98(5):873–895.

[56] Menec, V. H. and Chipperfield, J. G. (1997). Remaining active in later life: The role of locus of control in seniors' leisure activity participation, health, and life satisfaction. *Journal of aging and health*, 9(1):105–125.

[57] Milicevic, A. K., Nanopoulos, A., and Ivanovic, M. (2010). Social tagging in recommender systems: a survey of the state-of-the-art and possible extensions. *Artificial Intelligence Review*, 33(3):187–209.

[58] Miller, G. (1998). *WordNet: An electronic lexical database*. MIT press.

[59] Mokhtarian, P. L., Salomon, I., and Handy, S. L. (2004). A taxonomy of leisure activities: the role of ICT. *Research report (University of California, Davis. Institute of Transportation Studies)*.

[60] Mokhtarian, P. L., Salomon, I., and Handy, S. L. (2006). The impacts of ICT on leisure activities and travel: A Conceptual Exploration. *Transportation*, 33(3):263–289.

[61] Mongo, D. (2016). Top 5 considerations when evaluating nosql databases. *White Paper*.

[62] Moreno, A., Valls, A., Isern, D., Marin, L., and Borràs, J. (2013). Sigtur/e-destination: ontology-based personalized recommendation of tourism and leisure activities. *Engineering Applications of Artificial Intelligence*, 26(1):633–651.

[63] Musen, M. A., Middleton, B., and Greenes, R. A. (2014). Clinical decision-support systems. In *Biomedical informatics*, pages 643–674. Springer.

[64] Nef, T., Ganea, R. L., Müri, R. M., and Mosimann, U. P. (2013). Social networking sites and older users–a systematic review. *International psychogeriatrics*, 25(7):1041–1053.

[65] Nelson, M. E., Rejeski, W. J., Blair, S. N., Duncan, P. W., Judge, J. O., King, A. C., Macera, C. A., and Castaneda-Sceppa, C. (2007). Physical activity and public health in older adults: recommendation from the american college of sports medicine and the american heart association. *Circulation*, 116(9):1094.

[66] Nurbakova, D., Laporte, L., CALABRETTO, S., and Gensel, J. (2017). Users Psychological Profiles for Leisure Activity Recommendation: User Study. In *Proceedings of CitRec 2017: Workshop on Recommender Systems for Citizens*, Como, Italy.

[67] Organization, W. H. (2016). World health organization regional office for europe. http://www.euro.who.int/.

[68] OrientDB (2018). Why a multi-model database? | multi-model database. https://orientdb.com/multi-model-database/. Accessed: 2018-03-13.

[69] Park, D. H., Kim, H. K., Choi, I. Y., and Kim, J. K. (2012). A literature review and classification of recommender systems research. *Expert Systems with Applications*, 39(11):10059–10072.

[70] Pelleg, D., Moore, A. W., et al. (2000). X-means: Extending k-means with efficient estimation of the number of clusters. In *Icml*, volume 1, pages 727–734.

[71] Perrin, A. (2015). *Social Media Usage: 2005-2015: 65% of Adults Now Use Social Networking Sites–a Nearly Tenfold Jump in the Past Decade*. Pew Research Trust.

[72] Pree, W. (1995). Framework development and reuse support. *Burnett et al.[5]*.

[73] Pu, P. and Chen, L. (2006). Trust building with explanation interfaces. In *Proceedings of the 11th international conference on Intelligent user interfaces*, pages 93–100. ACM.

[74] Ramos, I., Brauchoff, I., and Marchese, M. (2017). Deliverable 4.4. Social Activity Repository (Final). ACANTO Project Deliverable, ATOS and University of Trento.

[75] Ramos, I., Mediavilla, C., Marchese, M., and Rodas, M. (2016). Deliverable 4.5. User communities creations based on user's profile matching (static profile): social network creation and evolution in older adults communities. ACANTO Project Deliverable, ATOS and University of Trento.

[76] Rodas Britez, M., Lissoni, D., and Marchese, M. (2018). An evaluation framework for groups' clustering algorithms in social networks - the use case of a meetup dataset of older adults. In Tallón-Ballesteros, A. J. and Li, K., editors, *Fuzzy Systems and Data Mining IV - Proceedings of FSDM 2018, Bangkok, Thailand, 16-19 November 2018.*, volume 309 of *Frontiers in Artificial Intelligence and Applications*, pages 417–427. IOS Press.

[77] Rodas Britez, M., Marchese, M., and Cernuzzi, L. (2017). Towards a social and physical activities recommendation system for active ageing. *IX Congreso Iberoamericano de Tecnologías de Apoyo a la Discapacidad - Iberdiscap 2017, ISSN 2619-6433*, pages 452–459.

[78] Rubenstein, L. Z. (2006). Falls in older people: epidemiology, risk factors and strategies for prevention. *Age and ageing*, 35(suppl_2):ii37–ii41.

[79] SAP (2018). Database as a graph store. https://blogs.sap.com/2016/06/20/sap-hana-database-as-a-graph-store-introduction/. Accessed: 2018-03-13.

[80] Sassi, I. B., Mellouli, S., and Yahia, S. B. (2017). Context-aware recommender systems in mobile environment: On the road of future research. *Information Systems*, 72:27–61.

[81] Schafer, J. B., Konstan, J. A., and Riedl, J. (2001). E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, 5(1):115–153.

[82] Schütze, H., Manning, C. D., and Raghavan, P. (2008). *Introduction to information retrieval*, volume 39. Cambridge University Press.

[83] Schutzer, K. A. and Graves, B. S. (2004). Barriers and motivations to exercise in older adults. *Preventive medicine*, 39(5):1056–1061.

[84] Shani, G. and Gunawardana, A. (2011). *Evaluating Recommendation Systems*, pages 257–297. Springer US, Boston, MA.

[85] Sinha, R. and Swearingen, K. (2002). The role of transparency in recommender systems. In *CHI '02 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '02, pages 830–831, New York, NY, USA. ACM.

[86] Sinha, R. R., Swearingen, K., et al. (2001). Comparing recommendations made by online systems and friends. In *DELOS*.

[87] Soler, J., Tencé, F., Gaubert, L., and Buche, C. (2013). Data clustering and similarity. In *FLAIRS Conference*.

[88] Steck, H. (2013). Evaluation of recommendations: rating-prediction and ranking. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 213–220. ACM.

[89] Szanton, S. L., Walker, R. K., Roberts, L., Thorpe Jr, R. J., Wolff, J., Agree, E., Roth, D. L., Gitlin, L. N., and Seplaki, C. (2015). Older adults' favorite activities are resoundingly active: findings from the nhats study. *Geriatric Nursing*, 36(2):131–135.

[90] Tang, J., Hu, X., and Liu, H. (2013). Social recommendation: a review. *Social Network Analysis and Mining*, 3(4):1113–1133.

[91] Team, R. C. (2014). R: A language and environment for statistical computing. r foundation for statistical computing, vienna, austria (2012) isbn 3-900051-07-0. *www.r-project.org*.

[92] Textbook, E. S. (2010). Finding the right number of clusters in k-means and em clustering: v-fold cross-validation. Technical report, Technical report, 2010. 6.

[93] Tinsley, H. E. and Eldredge, B. D. (1995). Psychological benefits of leisure participation: A taxonomy of leisure activities based on their need-gratifying properties. *Journal of Counseling Psychology*, 42(2):123.

[94] Torres, G. J., Basnet, R. B., Sung, A. H., Mukkamala, S., and Ribeiro, B. M. (2009). A similarity measure for clustering and its applications. *Int J Electr Comput Syst Eng*, 3(3):164–170.

[95] Ungar, L. H. and Foster, D. P. (1998). Clustering methods for collaborative filtering. In *AAAI workshop on recommendation systems*, volume 1, pages 114–129.

[96] Unger, L. S. and Kernan, J. B. (1983). On the meaning of leisure: An investigation of some determinants of the subjective experience. *Journal of Consumer research*, 9(4):381–392.

[97] Valeri, B. (2015). *Effective Recommendations for Leisure Activities*. PhD thesis, University of Trento.

[98] Vroman, K. G., Arthanat, S., and Lysack, C. (2015). "who over 65 is online?" older adults' dispositions toward information communication technology. *Computers in Human Behavior*, 43:156–166.

[99] Zanda, A., Menasalvas, E., and Eibe, S. (2011). A social network activity recommender system for ubiquitous devices. In *Intelligent Systems Design and Applications (ISDA), 2011 11th International Conference on*, pages 493–497. IEEE.

Appendix A

List of Selected Activities

Animal care	Engaging in lifelong learning	Practicing martial arts
Attending a meet-up	Environmental volunteering	Quilting
Attending a political rally	Exercising in gym	Radio listening
Attending a sports event	Feeding birds	Reading fiction
Attending a theatre play	Fishing	Reading newspapers and magazines
Attending musical performances	Gardening	Reading nonfiction
Attending religious services	Getting a massage	Refereeing a sports game
Attending social group meetings	Going to a flea market	Restoration of art works
Attending the University of the Third Age	Having a night out	Restoration of buildings
Backpacking	Hiking	Restoration of furniture
Bicycling	Horseriding	Riding a motorcycle
Birdwatching	Household cleaning	Rollerskating
Bowling	Hunting	Sailing
Camping	Jogging	Seeing a costumed carnival
Canoeing	Learning a craft	Shopping
Car racing	Learning a language	Sitting on a bench
Caring for house plants	Making origami	Skiing cross-country
Chatting on the phone	Meditating	Skiing downhill
Chatting online	Mentoring others	Skiing on water
Chatting with friends	Organizing photo albums	Solving jigsaw puzzles

Climbing
Collecting antiques
Collecting autographs
Collecting books
Collecting bottles
Collecting coins
Collecting fine-art photographs
Collecting stamps
Coloring a book
Cooking
Cultivating vegetables
Dancing
Decorating the house
Doing ceramics
Doing embroidery
Doing macrame
Doing photography
Doing pilates
Doing yardwork
Doing yoga
Drawing
Drinking and socializing
Driving cars
Eating out
Engaging in arts and crafts
Painting
Participating in research
Picnicking
Playing a musical instrument
Playing arcade games
Playing baseball
Playing billiard
Playing bingo
Playing board games
Playing bocce
Playing bridge
Playing cards
Playing checkers
Playing chess
Playing croquet
Playing frisbee
Playing golf
Playing mahjong
Playing poker
Playing racquetball
Playing soccer
Playing tennis
Playing video games
Playing volleyball
Playing with a pet
Speed dating
Storytelling
Sun-bathing
Swimming
Teaching a class
Theater acting
Traveling
Visiting a museum
Visiting art shows and galleries
Visiting friends and relatives
Volunteering for a charity
Volunteering for scout movements
Volunteering in an emergency
Volunteering: care giving
Volunteering: political activism
Volunteering: psychological or emotional support
Walking
Walking pets
Watching movies
Watching movies in a theater
Watching sports
Watching television
Weight lifting
Woodworking
Working part-time

Appendix B

Example of Clustering of Activities

This is a clustering example of Affinity Propagation algorithm using Dimensions and Pearson Correlation as distance measure.

Cluster Name	Activities
Event attendance	attending_a_sports_event, attending_a_theatre_play, attending_musical_performances, attending_the_university_of_the_third_age, getting_a_massage, seeing_a_costumed_carnival, theater_acting, visiting_art_shows_and_galleries
Home activities	caring_for_house_plants, collecting_photographs, coloring_a_book, organizing_photo_album, playing_arcade_games, playing_video_games, playing_with_a_pet, radio_listening, reading_newspapers_and_magazines, reading_nonfiction, sitting_on_a_bench, watching_movies, watching_sports, watching_television
Collecting	animal_care, camping, collecting_antiques, collecting_books, collecting_stamps, decorating_the_house, doing_photography, engaging_in_arts_and_crafts, traveling_for_leisure
Collecting	birdwatching, collecting_autographs, collecting_bottles, collecting_coins, doing_macrame
Yard activities	bicycling, cultivating_vegetables, doing_yardwork, environmental_volunteering, feeding_birds, gardening, hiking, walking_pets
Socializing	chatting_on_the_phone, chatting_with_friends, drinking_and_socializing, eating_out, going_to_a_flea_market, picnicking, shopping, visiting_friends_and_relatives, walking

Crafts	doing_ceramics ,drawing, engaging_in_lifelong_learning, learning_a_craft, learning_a_language, restoration_of_art_works, restoration_of_furniture, volunteering__psychological_or_emotional_support
Crafts	doing_embroidery, making_origami, painting, playing_billiard, solving_jigsaw_puzzles, woodworking
Competence activities	mentoring_others, participating_in_research, playing_board_games, playing_bridge, working_part_time
Social event activities	attending_religious_services, attending_social_group_meetings, chatting_online, having_a_night_out, playing_bingo, sun_bathing, visiting_a_museum, watching_movies_in_a_theater
Card games	bowling, meditating, playing_a_musical_instrument, playing_cards, playing_checkers, playing_chess, playing_mahjong, playing_poker, reading_fiction, storytelling
Field ball games	playing_baseball, playing_bocce, playing_racquetball, playing_tennis, playing_volleyball, practicing_martial_arts, quilting, volunteering_in_an_emergency
Lawn games	playing_croquet, playing_frisbee, playing_golf, playing_soccer, rollerskating, speed_dating, volunteering_for_scout_movements
Adventurous activities	backpacking, canoeing, car_racing, climbing, fishing, horseriding, hunting, restoration_of_buildings, sailing, skiing_cross_country, skiing_downhill, skiing_on_water
Activism	attending_a_meet_up, attending_a_political_rally, refereeing_a_sports_game, teaching_a_class, volunteering_for_a_charity, volunteering__care_giving, volunteering__political_activism
Fitness activities	cooking, dancing, doing_pilates, doing_yoga, driving_cars, exercising_in_gym, household_cleaning, jogging, riding_a_motorcycle, swimming, weight_lifting