TU WIEN Informatics

Exhibit rating prediction and visitor path prediction in a museum setting

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Data Science

eingereicht von

Bc. Marek Furka
Matrikelnummer 11934086

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Ass. Mag.rer.nat. Dr.techn. Julia Neidhardt
Ko-Betreuung: DI Niklas Reisz
Dr. Vito D.P. Servedio

Wien, 21. April 2022

Marek Furka

Julia Neidhardt

TU WIEN Informatics

Exhibit rating prediction and visitor path prediction in a museum setting

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Data Science

by

Bc. Marek Furka
Registration Number 11934086

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Ass. Mag.rer.nat. Dr.techn. Julia Neidhardt
Co-advisors: DI Niklas Reisz
Dr. Vito D.P. Servedio

Vienna, 21st April, 2022

Marek Furka | Julia Neidhardt

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Bc. Marek Furka

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 21. April 2022

Marek Furka

Acknowledgements

The author would like to thank the thesis supervisor Julia Neidhardt for great support throughout the work. I would also like to thank Niklas Reisz and Vito Servedio from CSH Vienna for making this thesis possible and for great and friendly support throughout the process as well.

Kurzfassung

Die Vorhersage von Bewertungen und Pfaden von Besuchern eines Museums oder einer Ausstellung, ist ein wichtiges Problem, welches jedoch in der Forschung bisher wenig Aufmerksamkeit gefunden hat. Die Literatur zu diesem Problem ist begrenzt, und Autoren vergleichen selten mehrere Ansätze, weshalb der aktuelle Stand der Technik unklar ist. Diese Studie basiert auf einem Datensatz, der aus expliziten Bewertungen von Museumsbesuchern besteht, die mithilfe einer Webanwendung in einer Ausstellung in Rom gesammelt wurden. Wir testen das Potenzial zahlreicher relevanter Ansätze und Baselines, welche wir bei Bedarf an unser Setting anpassen, und für welche wir mehrere geeignete Modifikationen der Ansätze vorschlagen. Diese Ansätze vergleichen wir anschliedend anhand eines umfassenden Bewertungsrahmens. Unser vorgeschlagener Bewertungsrahmen berücksichtigt die Reihenfolge der Ausstellungsobjekte, in der die Besucher die Ausstellungsobjekte besucht haben, was unserer Meinung nach in der physischen Umgebung notwendig ist. Wir analysieren auch die Performance der Ansätze in Bezug auf die Besucherpfadtiefe (die Anzahl der Bewertungen, die ein Benutzer bereits abgegeben hat). Im Fall der Bewertungsvorhersage konzentrieren wir uns auf die Präzision der Rangfolgemetrik bei k, und im Fall der Pfadvorhersage konzentrieren wir uns auf die Genauigkeitsmetrik der Folge. Bei dem Bewertungsvorhersageproblem beobachten wir, dass unsere Ansätze Verbesserungen gegnüber der Baseline bringen, je mehr Bewertungen der Benutzer abgibt, und schlechter abschneiden, wenn wir eine höhere Anzahl von Elementen bewerten. Wir haben außerdem herausgefunden, dass ein State-of-the-art-Ansatz, der für einen großen Datensatz aus einer anderen Domäne entwickelt wurde, im Allgemeinen überraschenderweise die beste Leistung erbringt. Für die Aufgabe zur Pfadvorhersage kommen wir zu dem Schluss, dass die Algorithmusleistung an verschiedenen Stellen in der Ausstellung weniger stabil ist, aber insgesamt schneidet ein Deep-Learning-Ansatz mit unserer vorgeschlagenen Modifikation im Allgemeinen selbst bei einem so relativ kleinen Datensatz am besten ab.

Abstract

Rating prediction and path prediction tasks in a museum/exhibition setting are important problems, yet not well explored. Existing literature is limited and authors rarely compare multiple approaches, leaving an apparent research gap. In this study, we aim to close this research gap by giving a comprehensive comparison of state-of-the-art techniques for predicting user ratings and paths, which we test in an exhibition setting. The whole study is based on a dataset consisting of explicit museum visitor ratings collected using a web application in an exhibition in Rome. We select numerous relevant approaches and baselines, adapt them to our setting if needed, propose several modifications of the approaches and compare them using a comprehensive evaluation framework. Our proposed evaluation framework considers the order of the items in which the visitors visited the items that we argue is necessary for the physical setting. We also analyze the performance of the approaches in terms of visitor path depth (the number of ratings a user has already given). In the case of the rating prediction, we focus on the ranking metric precision at k, and in the case of path prediction, we focus on the subsequence accuracy metric. In the evaluation for the rating prediction problem we observe that the approaches over-perform the baselines more, the more ratings the user gives and perform worse if we rank a higher number of items. In addition, we discovered that a state-of-the-art technique created for a large dataset from another domain unexpectedly performed best in general. For the path prediction task, we conclude a deep learning approach with our proposed modification performs the best in general unexpectedly on such a relatively small dataset.

Contents

CHAPTER 1

Introduction

Recommender systems are an essential part of many modern services in various domains. The systems are an important part of these services because they improve users' satisfaction with the service [KCL21], help to better understand the user [RRS10], increase the number of items sold, and therefore also increase the revenue [LH14].

One of the domains where the existing research regarding recommender systems is very limited is the museum/exhibition domain, despite several relevant possible use cases for such systems. Even though there are generally fewer items for a user to select from in this setting than in the common digital use cases of recommender systems, it is still often needed to recommend a subset of items to a user. The reason is that some visitors do not have the time to visit all the exhibits or might not be interested in seeing all the exhibits because of their preferences. Another common problem in museum exhibitions is overcrowding of certain exhibits or parts of the museum. This problem can be avoided up to a certain extent if we can predict the future exhibits and redirect users accordingly in real-time. Based on this motivation, the two of the most important use cases we analyze are rating prediction and path prediction of visitors.

1.1 Problem statement

The first problem is the rating prediction problem, where for each user we aim to predict her/his test set item ratings based on her/his and other users' training set ratings. In particular, we are interested in a specific variant of this problem where we aim to rank the unseen items according to likeability also referred to as a ranking problem. Since some users are not interested in seeing all the exhibits or do not have enough time to see all of them, we can use the predicted ratings to select a subset of exhibits that the visitor will like the most, for those visitors who have limited time or do not wish to visit all the exhibits.

Prediction of occupancy of exhibits is important because with this information we can redirect the visitors in way that minimizes the overcrowding. This is the second problem studied in this work, that we call the path prediction problem. The task is to predict the future visited exhibits for a visitor based on her/his previous actions. The obtained sequence predictions can then be used for predicting the occupancy of exhibits. This can be done in real-time, which means the visitors could be redirected appropriately in case too many visitors are predicted at a certain exhibit. The redirection of visitors can be possibly done in combination with the predicted ratings from the previously outlined problem so that users can be redirected to exhibits they would prefer to visit. This possible application is especially relevant in the context of physical distancing due to the ongoing COVID-19 pandemic.

For the rating prediction, the current research gap consists of a non-existent systematical evaluation of multiple approaches in the same setting. Additionally, some of the most common approaches used for rating prediction, or common rating prediction baselines in other domains, which are also applicable in our setting were not yet tested in this domain. In addition to that, our proposed evaluation procedure reflects the specific real-life usage pattern of such a system which is different than in the case of recommender systems used in the digital setting as explained in chapter 3 in detail.

Similarly, for the path prediction problem, evaluation of multiple approaches in the same setting does not exist in the literature according to our knowledge. Also, the datasets used in the other similar experiments were collected in different ways - IoT (internet of things) sensors and physical data collection. Thus, another contribution of the thesis is an answer to the question of how accurate are these approaches in the case of explicit user feedback data collection procedure. As in the case of the rating prediction task, the evaluation procedure for path prediction reflects the specifics of the real-life use case in the museum/exhibition setting.

The research was done in cooperation with Complexity Science Hub Vienna and Sony CSL. The research institute developed a web application used for enhancing the museum/exhibition visitor experience and collecting data about user's visit and her/his preferences [App]. The application lets the user select an exhibit, rate the exhibit and offers some additional content accompanying the exhibit as seen in the figure 1.1. This application was tested in an exhibition in Rome called "Uncertainty. Interpreting the present, foreseeing the future", [Exh] taking place from the beginning of October 2021 until the end of February 2022.

1.2 Research questions

Based on the motivation and problems outlined in the previous section we have defined two research questions:

1. "What are appropriate methods to accurately predict user ratings in an exhibition or museum setting?"

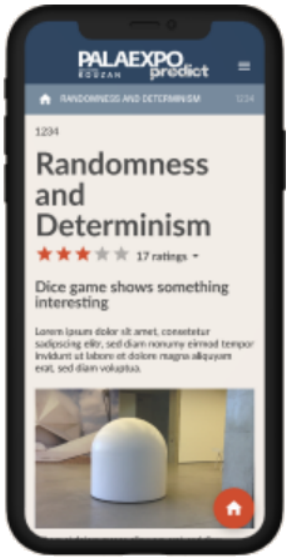

Figure 1.1: Screenshot of the Kouzan application used to collect the data

2. "What are appropriate methods to accurately predict the next visited installation at every point of a user's path?"

1.3 Methodological approach

Our methodological approach is based on one of the most commonly used data science frameworks - CRISP-DM Model (Cross Industry Standard Process for Data Mining) [WH00]. It is a model with six phases that naturally describe the data science life-cycle. The steps are:

1. Business understanding - identify the problems, understand the specifics of the museum setting.

2. Data understanding - understand the data collection process and the actual data structure. Check for possible problems and anomalies in the data.

3. Data preparation - clean the data, remove problematic records, missing value handling, outlier removal etc. In this step, remove the users that had unrealistically short visits according to the data.

4. Modeling - based on literature review, select several most relevant models and approaches for both problems, including proposed modifications of these models.

5. Evaluation - create domain specific evaluation framework with evaluation metrics that express the relevance for the use cases.

6. Deployment - implement the best performing approaches to the Kouzan application. This step is left for future work as it is out of the scope of this thesis.

1.4 Main thesis contributions

The main contributions of thesis are:

- A novel domain specific evaluation framework that considers the characteristics of the domain was created.
- Relevant rating prediction approaches were compared using the evaluation framework at various visitor path depths.
- Relevant path prediction approaches were compared using the evaluation framework at various visitor path depths.
- Analysis of sensitivity to the total dataset size of all the approaches for both problems was performed.
- One of the proposed modifications for the path prediction problems turns out to slightly outperform the original algorithm.

1.5 Thesis structure overview

The thesis is structured as follows:

- Chapter 2, explains the data collection procedure, analyzes the collected dataset, presents an overview of the existing literature on the topics, and explains the context of the selected approaches which are analyzed.
- Chapter 3 serves as a detailed explanation of the overall approach including the experiment design, statistical significance testing, prediction metrics, selected approaches from an algorithmic perspective, and a brief overview of the implementation.
- Chapter 4 presents the detailed results from the experiments in terms of plots visualizing the metrics and textual interpretations for both rating and path prediction tasks.
- The final chapter 5 summarizes the results, gives an outlook on the possible future applications of the recommenders, and also discusses the limitations of this work and possible future work.

CHAPTER 2

Background & Related Work

This chapter offers more contextual information, relevant for understanding the problems introduced in the first chapter. The application used for data collection is presented in subsection 2.1.1 to facilitate an understanding of the data structure and its properties. The actual collected data with its characteristics is presented in subsection 2.1.2 which is relevant for understanding the proposed solution design. Sections 2.2 and 2.3 offer a literature overview for rating prediction and path prediction problems respectively and explain the proposed solutions within the context of the existing literature.

2.1 Data collection and analysis

The following subsection presents the application used for data collection in a more detailed way, as well as explains the properties and various characteristics of the resulting collected data.

2.1.1 Application used for data collection

As briefly explained in the introduction chapter, the data used for the research comes from the Kouzan web application developed by Complexity Science Hub Vienna in cooperation with Sony CSL [App]. The application provides simple functionality for visitors of exhibitions. First, the users register, provide basic demographic information which consists of their age group, gender, education level, and a binary attribute that indicates whether the visitor has a scientific background. The possible values of these attributes with their frequencies can be seen in figures 2.4,2.5,2.6 and 2.7. Then the registered users see a map of the exhibits, where they can select one of the exhibits, by either scanning the exhibit QR code or manually entering numeric code corresponding to the exhibit. The QR and numeric codes are attached to the exhibits. This implies that a user needs to be standing next to the exhibit in order to rate it. After selecting

an exhibit, the users can rate it on a scale of zero to five stars, with the possibility to also give half stars (.5 values). The application also guarantees each exhibit is rated only once. The structure of the collected data is therefore very simple and it is possible to apply the results of our research elsewhere without being restricted to this specific application that was used in this case. When users select an exhibit, they are also provided with an additional content regarding the exhibit which motivates the visitors to use the application as seen in the figure 1.1. Additionally, when a user rates an exhibit, then he/she is provided with feedback on how predictable his/her action was, therefore the application is also an exhibit itself in a way. For this, the authors have implemented simple algorithms serving as placeholders that are not based on the literature which is the reason why these are not included in our evaluation framework. The application was tested at an exhibition in Rome called "Uncertainty. Interpreting the present, foreseeing the future" [Exh]. The exhibition took place from the beginning of October 2021 until the end of February 2022. The collected data was processed in accordance with the GDPR and no personal data was exposed in this thesis.

2.1.2 Analysis of collected data

During the duration of the exhibition, we have collected 2646 individual ratings given by 495 distinct visitors. The exhibition originally contained 24 exhibits. Exhibit number 22 was removed in the early stages of the exhibition, therefore due to very few ratings of this item, we remove it from the data used for our research. This leaves us with 23 unique exhibits.

Since the visitors are being tracked only by their explicit actions (assigned ratings), some users might not use the application during the entire visit. As seen in the figure 2.1, many users have extremely short visitor paths. It is very unlikely someone would visit the exhibition just to see two or three exhibits, instead it is most likely the case that some visitors get bored after using the application briefly and decide not to use it for the rest of their visit. There is no definite way how to determine, which users have used the application properly and which have not. It is therefore needed to set a cutoff threshold, to eliminate the unrealistically short paths, while still preserving a reasonable amount of data, which is an obvious trade-off. As we can see in the distribution in the figure 2.1, the number of visitors stops decreasing and starts increasing again at a visitor path length of 5. We have therefore decided to remove all the visitors with path lengths lower than 5, based on the assumption that this number should be increasing up to a certain most common path length (which is 8 in this case with the exception of the total path length of 23). The effect of the filtering can be seen in box-plots visualizing the distributions of the visitor path length before and after the filtering in figures 2.2 and 2.3 respectively.

After filtering out the users with too short visitor paths we are left with 170 distinct users and 2131 ratings. The following analysis was done on the filtered dataset as well all the other results in this thesis.

The demographic characteristics of the visitors based on the data they have provided are

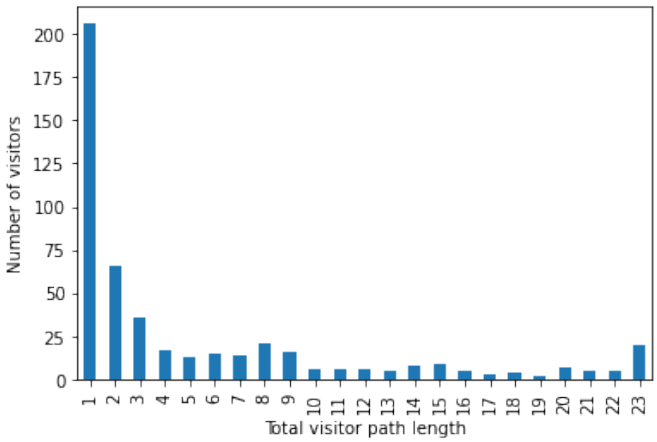

Figure 2.1: Number of visitors per different total lengths of visitor paths

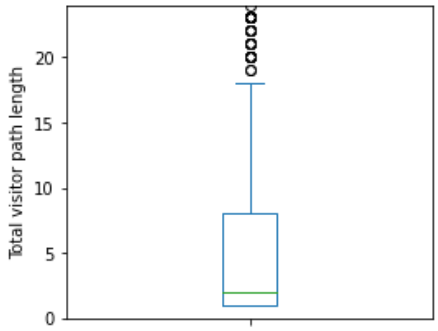

Figure 2.2: Distribution of number of ratings per user visualized by a box plot

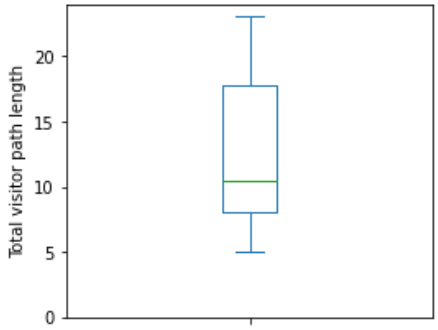

Figure 2.3: Distribution of number of ratings per user after filtering visualized by a box plot

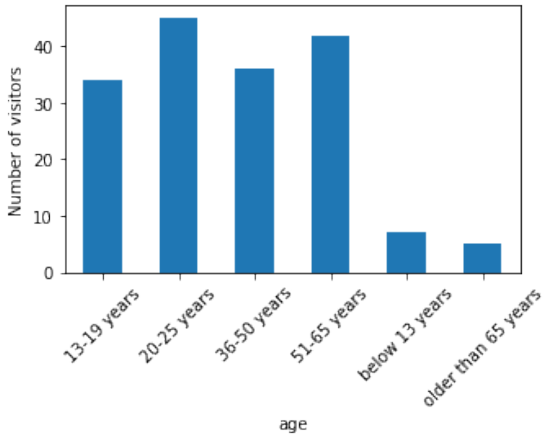

Figure 2.4: Demographics of the visitors - number of visitors per distinct values per age group

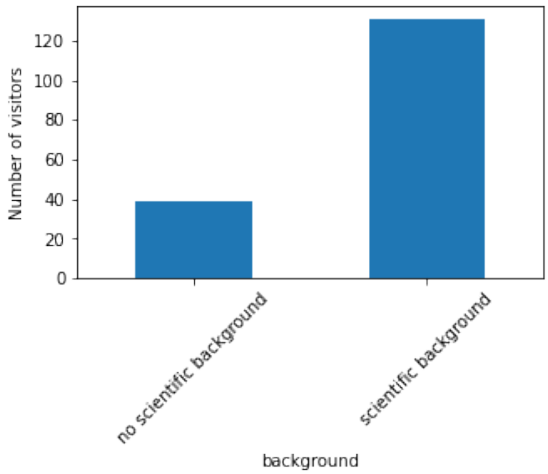

Figure 2.5: Demographics of the visitors - number of visitors per distinct values of background attribute

shown in figures 2.4,2.5,2.6 and 2.7. The age of the visitors is diverse, as we can see the age groups are roughly balanced with exception of the youngest and oldest age groups which are less common. Since the contents of the exhibition are rather oriented towards a scientific audience, we can see the majority of the visitors claim to have a scientific background. This is also clearly visible in the education attribute where the majority of visitors have higher education status. Regarding the gender structure, we can see the men and women ratio is almost balanced.

In the figure 2.8, we can see the distribution of the ratings on a per exhibit basis, illustrated by box plots. We can observe that users are generally more likely to give higher rating values and we can also notice some exhibits are in general less popular than

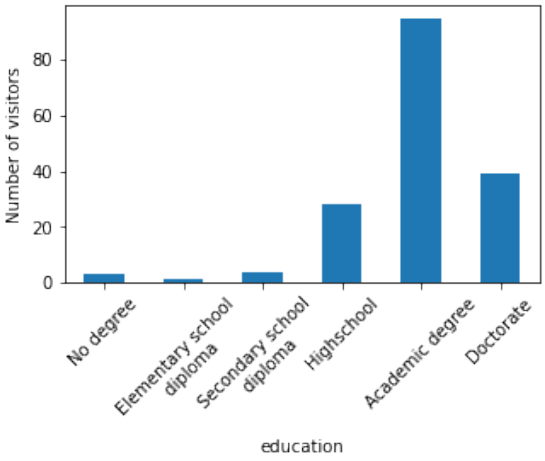

Figure 2.6: Demographics of the visitors - number of visitors per distinct values of education attribute

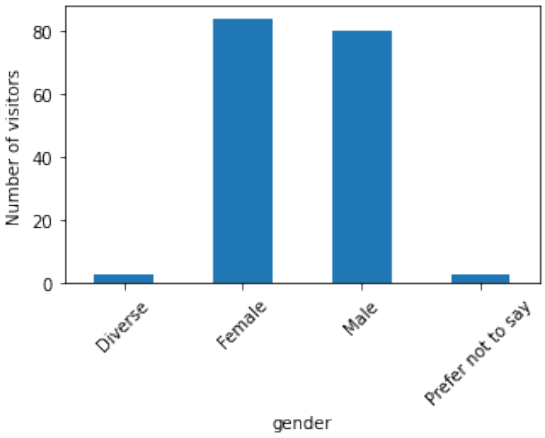

Figure 2.7: Demographics of the visitors - number of visitors per distinct values of gender attribute

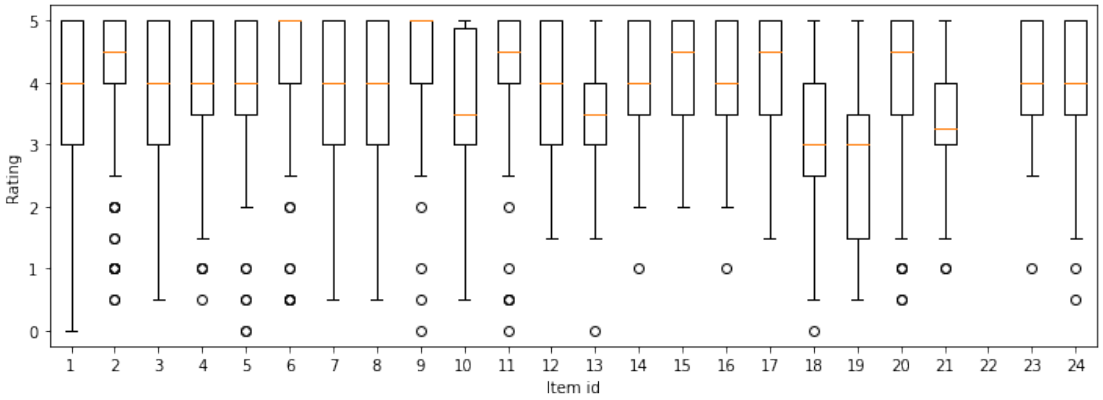

Figure 2.8: Box-plot of rating distribution per exhibit

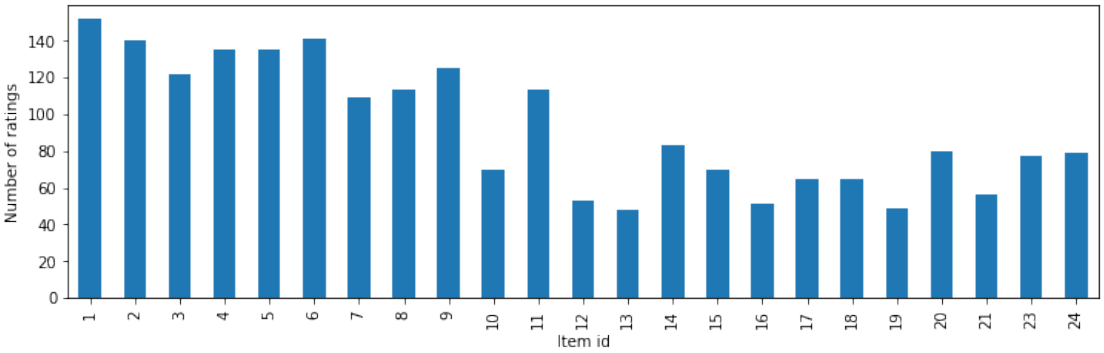

Figure 2.9: Number of ratings per exhibit

others.

In the figure 2.9, we can see the total number of ratings per exhibit after the filtering. As expected, the items in the initial part of the exhibition are rated more often, since some of the visitors do not visit all the exhibits, which shows up at the number of ratings of exhibits placed further from the entrance to the exhibition (mostly exhibits with id number 12 and up).

In the figure 2.10, we can see the number of ratings per distinct possible values. We can again observe that users tend to use the higher rating values more and we can also notice the half values (.5) are less commonly used than integer values.

For the purpose of the path prediction task, for each visitor, we consider the rated items in ascending order of the rating timestamp as the sequence of visited exhibits. The physical layout and transitional relationships between the exhibits are illustrated in the figure 2.11. The numbered nodes shown in the figure correspond to the exhibit ids and the directed edges show the Markov transition probabilities between the exhibits. We can see the visitor flow is more straightforward in the initial part and gets complicated in

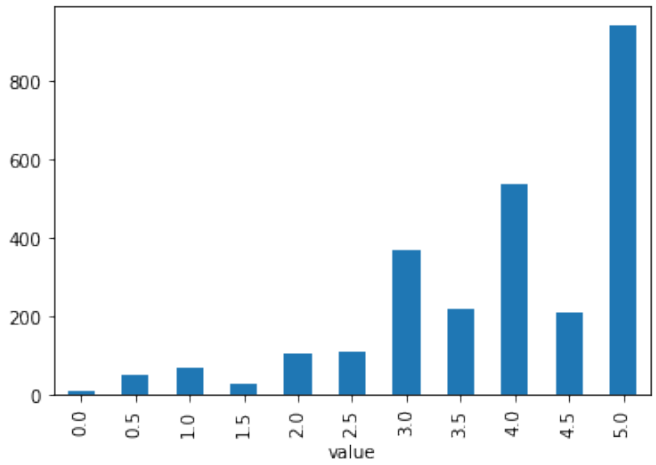

Figure 2.10: Number of ratings per distinct possible ratings values

the case of the exhibits which are typically visited later within the visitor path. We have also added an additional exit state representing the final point in each visitor's sequence denoted as "-1" (the state is treated as any other exhibit in the algorithms). Without this data modification, it would not be possible to predict that the visitor will leave the exhibition for some approaches.

The exhibits in the three smaller rooms on the right side of the figure 2.11 are in physical space located right next to each other therefore it is rather random which item the visitor scans first. For the main motivation of the path prediction task, which is the occupancy prediction it makes more sense to aggregate such exhibits. Therefore, we consider the predictions at this level of granularity as unnecessary and also rather unfeasible. As we can see in the figure 2.11, the transitional patterns within the rooms are complicated. This results in the exhibits 12,13,14,15,16 being represented as room 101, the exhibits 17,18,19 being represented as room 102 and the exhibits 20,21,23 being represented as room number 103. While we also report the evaluation results for data without aggregated rooms, our major focus is the aggregated alternative. For one of the later explained path prediction approaches where we also need the ratings, we consider average room ratings in this case. The resulting Markov probabilities for the exhibits with aggregated rooms are shown in the figure 2.12.

2.2 Rating prediction in museum/exhibition setting

The following subsection gives an overview of the relevant literature for the rating prediction problem in the museum setting, as well as an overview of the selected approaches, within the context of the related literature.

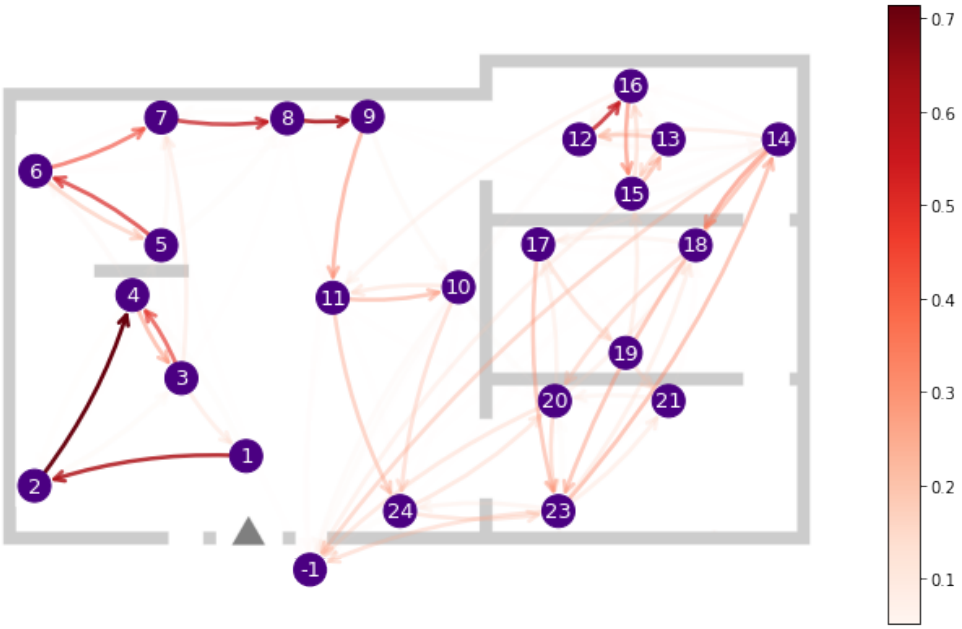

Figure 2.11: Exhibition map with nodes representing the exhibits with an additional "exit state" denoted as -1, and the edge colors representing Markov chain transition probabilities

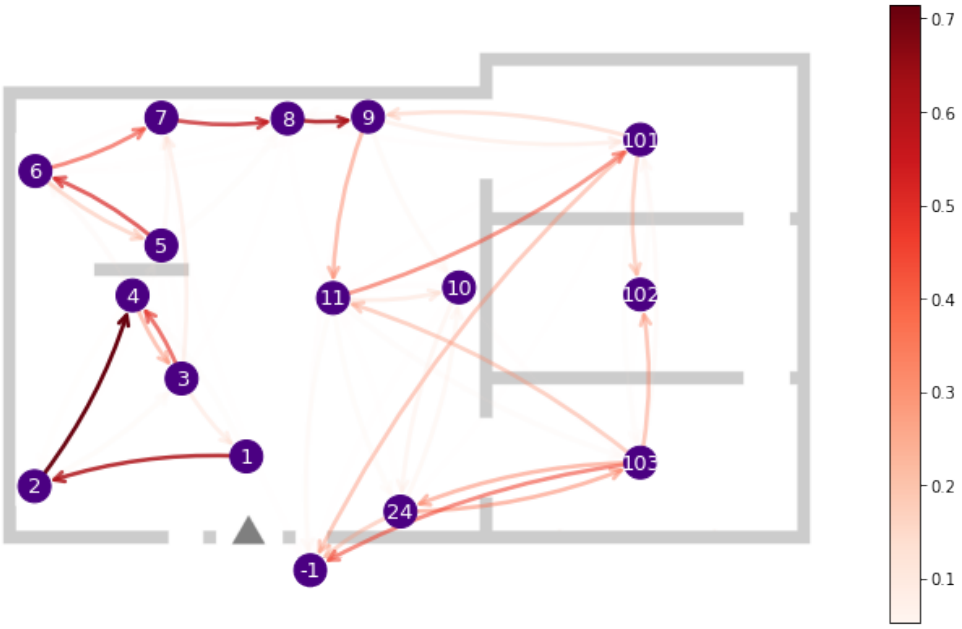

Figure 2.12: Exhibition map with nodes representing the exhibits with an additional "exit state" denoted as -1 and aggregated rooms, and the edge colors representing Markov chain transition probabilities

	i_1	i_2	i_3	...	i_n
u_1		4	5		
u_2	3.5		4		
u_3		5			
...					
u_m					

Table 2.1: An example of a sparse rating matrix with n items, m users and the ratings as the values

2.2.1 Background and related literature

The rating prediction task consists of predicting empty elements in a so-called rating matrix with m users and n items, where the values represent the ratings assigned by the users to the items in the system as illustrated in table 2.1. The predicted ratings can then be used to rank the items, the visitor has not seen yet, to select a subset of exhibits the visitor will like the most. This is for example useful in case she/he does not wish to visit all the exhibits. This matrix is typically very sparse [RRS10]. For example, one of the most popular datasets used for evaluating rating matrix completion approaches called MovieLens 1M dataset [Mov], contains 6000 users, 4000 items, and 1000000 ratings. Therefore, only about 4.2% of the ratings are known. The rating sparsity is typical for almost all of the real-life use cases. In video streaming platforms, music streaming platforms, e-commerce stores, social networks, etc. there are at least several thousands of items for the user to rate/interact with and a typical user only rates/interacts with a very small fraction of the available items. This is different from our use case, where a recommender system is being applied in a physical setting. The number of available items for users in our setting is much lower - in our collected dataset, there are 23 items/exhibits (and therefore the rating matrix density is a lot higher). With 2131 ratings and 170 users in our dataset, this gives us a rating density of 54.5%. Of course, the outputs of this work could be applied to several times bigger exhibits/museums, however, the rating density is still expected to be significantly higher when compared to the digital setting. Therefore, we consider this property as one of the characteristics of our research problem.

The second characteristic which distinguishes our setting, from the common application settings of rating prediction approaches, is the selection of the first items the user interacts with. In a digital setting, the first items the user rates can be any items. On the other hand, in the physical museum setting, the selection of the first items is dependent on the physical layout of the exhibition. The first items are typically near the entrance to the exhibition, as observed in our data. This is shown in the figure 2.13, where we see the exhibition map, with node sizes and labels representing the number of times the particular exhibit was within the first four visited exhibits within the visitor path.

The previously explained characteristic is also taken into account in the design of the evaluation procedure. The training set is the portion of data, we use to train an algorithm

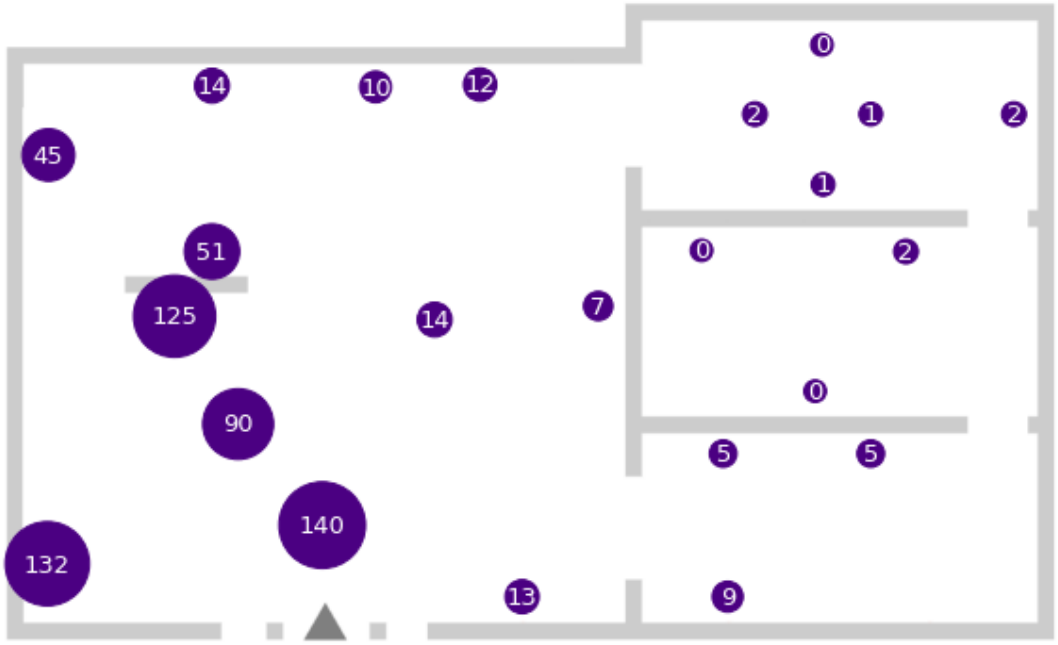

Figure 2.13: Exhibits visualized on an exhibition map with node labels and node sizes representing the number of times the exhibit was within the first four visited exhibits within the visitor path

and the test set is the portion of the data, used to evaluate the performance of the algorithm. In general, for the rating prediction algorithms, the training and test set data split is done irrespective of the order, in which the users rated (visited) the items. On the other hand, we consider the order of ratings of each user when splitting the data to simulate the real-life use case - i.e. the first k items for a test set user are assigned to the training set. Therefore, for example at $k = 4$ the items in the training set will be more biased towards the entry area of the exhibition as demonstrated in the figure 2.13. We also evaluate the approaches at different k which we call "visitor path depth" - i.e. the number of items the user has already rated as explained in more detail and visualized in chapter 3. To our knowledge, this point of view was not analyzed in the literature previously.

In certain domains, the training and test split is also taking into account the time of rating, in a sense that ratings given later in general, are assigned to the test set [CDC14] (instead of considering per user rating order). Therefore the train-test cutoff point would then be a certain point in time instead of the number of already visited exhibits. The intuition behind this point of view, is that in some domains, it is important to consider certain trends that might evolve over time - for example, a different movie on a movie

streaming platform, might be more popular at a certain point in time. In a museum setting, we do not expect the items to be more or less popular over time and therefore we do not consider time-aware splits.

This research problem is relatively novel and previous work in museum exhibition/domain rating prediction is very limited. There are several works dealing with similar problems in a related setting, however, they are often not applicable to our problem definition.

Smartmuseum is an example of a similar application [RHS+13]. The authors have developed mobile as well as desktop applications, that can recommend points of interest in an outdoor scenario, as well as museum artifacts in an indoor scenario. The users can provide feedback on the items, but instead of a rating scale, she/he can only assign positive or negative feedback. The main difference is, the recommendation process in this application, is ontology-based and uses semantic web approaches, such as ontology-based reasoning, query expansion, search results clustering, etc. The recommendation process in this application also requires certain inputs from the curators, such as which artifacts are suitable for families, school groups, solo visitors, and other contextual information. In contrast, our problem definition is a lot more general and the information in our dataset does not contain such complex exhibit content information, which makes such an approach not applicable.

Another similar application to Smartmuseum is Cultural Heritage Information Personalization (CHIP) [WWN+09]. This complex application has several functions, such as planning functionality which lets users plan their museum visit in advance, the application helps with navigation throughout the museum and also provides an exhibit recommendation functionality. Again, unfortunately, the recommendation procedure is based on semantic web technologies, as in the previous case which is not applicable to our data.

GECKOmmender is another related application, which was used in previous research [BZL12]. This application is capable of collecting explicit star ratings, similarly to ours, and is also used for providing recommendations based on the predicted ratings. However, the approach used for rating prediction is content-based filtering. The prediction approach consists of calculating a weighted average of the items the user has already rated, where the weights are content-based similarities with the target item, which is being predicted. The similarities are based on textual annotations of the exhibits, while in our setting we do not have any content attributes .

A completely different point of view for predicting museum visitors' interest in exhibits, was taken in a game-based museum exhibit, called "FUTURE WORLDS" [EHR+20]. The authors have used visitors' facial expressions, eye gaze, posture from dedicated sensors together with interactions logs from the interactive exhibits as machine learning features, which they used to predict visitors' dwell time - the amount of the time the visitor spends with an exhibit.

2.2.2 Selected approaches

All of the selected approaches are explained in detail in chapter 3, this chapter offers an overview of how the approaches are related and why they were selected.

We select several baseline predictors for our evaluation framework to see whether the more complex algorithms are performing better than these naive approaches. Average user rating baseline, average item rating baseline, and a baseline approach taking into account user and item biases were selected.

Two major types of rating prediction approaches are content-based approaches and collaboration-based approaches. The main idea of content-based approaches [LdGS11], is to calculate the similarity between items based on the contents of the items (which can be defined in various ways). Then the idea is to recommend similar items to the items the user previously liked. As already mentioned in the previous literature overview, we do not rely on any data that would define the item contents therefore content based approaches are not relevant for us. The advantage is that for many exhibitions, defining the contents of the exhibits might be very time-consuming or very difficult to describe.

The other category - collaboration-based approaches rely only on the ratings. Collaborative filtering approaches belong to this category, they calculate the similarity between users or items based on the ratings and then recommend an item, that similar users have previously liked, or items that are similar to the items the target user has previously liked [RRS10]. Matrix factorization [KBV09] and Regularized kernel matrix factorization [RST08] are well-performing approaches which aim to find a model-based representation of user and item characteristics which are then combined to produce a rating. These approaches suffer from the so-called new user cold start problem, which means we can't provide good recommendations until the user rates sufficient number of items. There is also a new item cold start problem, which means a new item that does not have enough ratings yet, can't be predicted accurately. However, this will rarely be a problem in our setting, since new items are rarely added to an existing exhibition, compared to a movie streaming platform for example. Therefore, only new user cold start is relevant. A possible solution to this problem, besides a non-personalized baseline, is a demographic-based recommendation approach since we have also collected basic demographic information about the visitors. Inspired by previous work, in the tourism domain [WCN12], we tested a machine learning-based approach, which relies only on the demographic user attributes. Additionally, we have proposed a combination of demographic-based and collaboration-based approaches, where we combine the demographics and the collaboration-based predictions. We have also tested the performance of the current state-of-the-art approach for general rating prediction dataset in a digital setting - specifically for the popular Movielens 1M dataset [Mov]. The approach is called G-Local K [HLL+21].

The most relevant previous work for our setting is a graph-based recommender, which was shown superior to other recommender approaches [MKK17]. The authors have used an application that allows the users to rate the so-called presentations, which are multimedia artifacts corresponding to the exhibits. Each exhibit can have multiple presentations.

The experiments were done in a similar setting, the data was collected over several months at Hecht Museum in Haifa where the users used an application that allowed them to rate the presentations on a three or five point rating scale. The concept of rating multimedia presentations instead of the actual exhibits is of course a difference between this work and our setting, nevertheless, it is still the most similar previous research to our problem. Another difference is that in this work, the original dataset contained more information such as exhibit themes, positions, spatial relationships, and item similarities which were all used in the prediction process, which is explained in detail in chapter 3 together with the necessary adaptations. The authors have compared their graph-based approach with item-based collaborative filtering (CF), matrix factorization, hybrid recommendation consisting of a scaled combination of the two previous approaches, and finally a non-personalized random baseline. Their proposed graph-based approach was shown to be superior in terms of ranking-based metric - mean reciprocal rank (MRR). MRR is a ranking-based metric that only focuses on the most relevant item according to the ground truth ratings. It is computed as the mean of the reciprocal ranks for all queries. For example, if the item with the highest ground truth rating is ranked third according to the predictions then the reciprocal rank is $\frac{1}{3}$. Since this previous work is the only one that compares multiple approaches in a setting similar to ours, we consider their proposed graph-based approach as the state-of-the-art approach.

2.3 Path prediction in museum/exhibition setting

The following subsection gives an overview of the relevant literature for the path prediction problem in the museum setting, as well as an overview of the selected path prediction approaches, explained within the context of the related literature.

2.3.1 Background and related literature

In the path prediction task, we aim to predict the sequence of the next visited exhibits based on the visitor's previous actions. This information can then be used to predict exhibit occupancy in real-time and opens up the possibility to redirect the visitors accordingly. Because different visitors might spend different amounts of time at different exhibits, predicting exhibit occupancy too far in the future based on this information would be too unreliable. Due to this limitation, we focus only on the three next future visited exhibits in the evaluation.

Several related recommender system types exist, which are seemingly relevant for this task, but they differ in certain conceptual details. One example are sequential recommender systems [WHW+19]. This type of recommender systems is most commonly used in an e-commerce setting, where the task is to predict the next best item for the user, based on the items she/he has previously interacted with. For example, if a user buys a smartphone, a smartwatch then the system could recommend an item such as compatible headphones or similar. On the other hand in our physical setting, the next predicted item is not based on contextual content relatedness rather on spatial relationships between

items and visitor's patterns of museum exploration. However, some general approaches also used in sequential recommenders such as Markov chains, are relevant for our setting as explained later.

Another related recommender system type is a point of interest (POI) sequence recommender systems. These are mostly relevant in the context of location-based social networks (LBSN), which are a specific category of social networks, where the user "checks-in" at various types of POIs that they visit - for example restaurants, cafes, shops, cultural venues, etc. [IMSA22]. The data also often contains a friendship relationship structure among the users [LGHZ16], POI reviews, etc. The task is to either predict the next POI or even a sequence of next POIs, in which the user might be interested. Some of the factors which influence the users in the next POI selection, that are considered in POI recommendations approaches differ from our setting. The selection of the next POIs is influenced by temporal aspects (e.g. a person is interested in visiting a different place in the morning vs. in the evening) [YCS14] and other contextual factors such as weather [ZZLW21]. The influence of spatial proximity is also different among POIs compared to museum exhibits. The scale of spatial relationships is different, thus are the properties modeled in POI recommenders different as well (e.g. users tend to check-in around the same location where they live, while at museum scale they are more likely to explore items spread across the area). Another conceptual difference is the social influence considered in POI recommenders, while in our data we do not have any information regarding the user social structure.

There are a few other problems in the literature that might at first resemble our problem definition at least by the name, but the approaches are not applicable in our setting. Trajectory prediction is a task where the aim is to predict future trajectory in a scene, in terms of future position coordinates of humans or objects such as cars. The inputs for the prediction process differ across the datasets and approaches - the input signals can consist of past motion history, RGB scene images, location and gaze of other pedestrians in the scene, location of other objects such as cars in the scene, etc. The form of the outputs also differs case by case [MAGM20].

The trajectory prediction task as defined above was even analyzed in the museum domain [BLBDB17]. The authors have collected a dataset based on footage from several cameras in a large crowded room of a museum. The authors have proposed a novel deep learning-based approach, which is able to model and estimate how much the various elements in the scene influence the trajectory of the visitor.

Several studies focus on an analysis of visitor movement throughout the museum, without analyzing the path prediction problem. One previous work demonstrated that it is possible to categorize museum visitor behavior into several categories based on data collected using a museum guide application [KBZ12]. The categories were defined as follows: "the ANT visitor tends to follow a specific path and spends a lot of time observing almost all the exhibits; the FISH visitor most of the time moves around in the center of the room and usually avoids looking at exhibits' details; the BUTTERFLY visitor does not follow a specific path but rather is guided by the physical orientation of the

exhibits and stops frequently to look for more information". The authors claim that this classification can be used to better adapt the services to the museum visitors.

In another similar previous study, the authors have used IoT devices to collect data about visitor trajectories at a room scale and then use the data to analyze the visitor behavior [CCCO21]. Thanks to the sensor-based data collection, the authors have analyzed the time of permanence in different rooms which was shown to resemble Weibull distribution. The authors have also calculated room transition probabilities using the Markov chain. The transition probabilities together with the average time spend in the rooms, were used to model the museum and propose a new ticketing and entrance system for the museum.

2.3.2 Selected approaches

All the selected approaches are explained in more detail in chapter 3, this subsection offers an overview and puts them in context with the literature. A few works that deal with predicting the museum visitor path directly at the exhibit granularity exist. Unfortunately, none of them work with explicit feedback data as we have in our problem definition. Instead, the data sets in the previous works were collected either manually or using IoT sensors. Despite this difference, the following studies are the most similar to our setting. None of these previous works compares multiple approaches, therefore it is unclear what approach to consider as state-of-the-art. Additionally to the more sophisticated approaches based on the literature, we have also proposed a simple baseline in which we predict the closest unvisited exhibit. This is based on the assumption that the visitor is influenced the most by spatial proximity.

Markov chain was already used in previous work at room level [CCCO21], and in another case directly at exhibit level [BZB+08]. A hybrid approach was also used in the domain, where the authors have combined the Markov chain probabilities with implicit ratings based on the duration the users spend with different exhibits [BZB+08]. In our case we do not need to rely on implicit ratings, we can directly use the predicted explicit ratings. We have selected regularized kernel matrix factorization to predict the ratings in the hybrid recommender because as later shown in the results section for the rating prediction task, it was the best rating prediction approach in terms of root mean squared error - RMSE (which we argue is relevant when combined numeric scores). RMSE measures the difference between the real ratings and the predicted ratings. The measure emphasizes the more erroneous predictions and it is defined in section 3.1.4.

The most complex approach we have tested is based on previous work, where the authors attempted to solve the problem using deep learning [PGC+20]. This approach is based on sequence to sequence neural network architecture, which is most commonly used to language translation tasks. The input to the network is a sequence of tokens and the target is again a sequence of tokens (tokens are words in language translation task). The authors of the original work have proposed to use transitions between exhibits as tokens (for example sequence 2-3-5 is encoded as (2,2)-(2,3)-(3,5)). We have proposed a more simplified approach and instead, we use directly the exhibit id numbers as tokens. The

intuition is, that more simple input encoding with a lower number of distinct tokens, might facilitate the learning process in case of a relatively small dataset size (in the original work the dataset consisted of about 10000 visitors). Another modification of the deep learning approach which we have proposed, is based on the expectation that different demographic groups move differently around the exhibition and the deep learning could leverage from the additional information. We have encoded all the distinct values of the demographic attributes as unique tokens and then we have added these tokens to the beginning of the deep learning input sequence (one additional token per attribute). This modification was tested for both discussed exhibit encodings (transitions/exhibit only).

CHAPTER 3

Approach

This chapter explains the design of our evaluation framework which is used to test and compare the algorithms. The actual descriptions of the algorithms from a more detailed algorithmic point of view complementing their overview in the previous chapter are also presented in this chapter.

3.1 Evaluation design

The following subsection explains the evaluation process in detail, including the custom training test set split, cross-validation procedure, statistical testing procedure, and all the used evaluation metrics for rating and path prediction problems.

3.1.1 Domain-specific train-test split

Recommender systems that are being used in a physical museum/exhibition setting need a different evaluation procedure than recommenders systems used in a purely digital setting. In a digital setting - for example in a movie streaming platform, music streaming platform, or an e-shop the first items user rates can be any items. On the other hand in a physical setting such as the museum, the first items the user rates are determined by the physical layout of the environment. The first ones are normally located in an entry area of the exhibition, as already previously shown in the figure 2.13.

Because of this difference we do not perform a simple random train-test set split. Instead, we perform a user-based train-test split, where a subset of users is selected for the test set and the rest is assigned to the training set. For the test set users, her/his first n ratings (according to the timestamp of the rating - respecting the order) are part of the training set. Where n is what we call the visitor path depth. We test the algorithms at different path depths. The reason is, that with different amounts of ratings already given by the user, we expect different approaches to be the best-performing ones. We can see

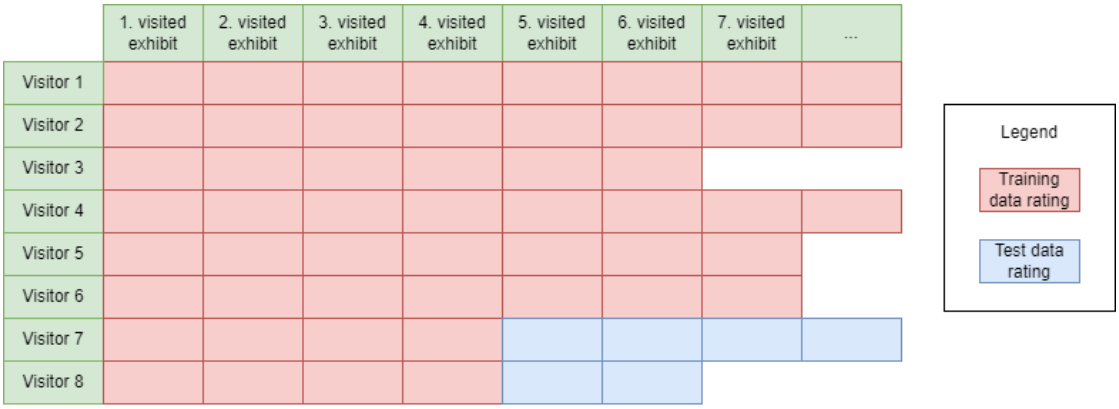

Figure 3.1: Example of a train-test split with 25% test ratio at visitor path depth 4 if we had only 8 visitors

the split procedure illustrated in the figure 3.1. The example illustrates the case if we only had eight visitors, wanted to assign 25% of users to the test set, and choose 4 as the visitor path depth. Analogously for visitor path depth of 8, the first eight ratings of visitors number 7 and 8 in the figure would belong to the training set and the rest to the test set. We split the data analogously in the case of the visitor path depth of 12.

3.1.2 Experiment design

In order to be able to make the experiments more robust and to be able to test statistical significance when comparing the algorithms, we perform k-fold cross-validation. This means $1/k$ number of users are put into the test set and the rest is put to the training set (respecting the previously described split procedure). This is repeated k times where each split is used as a test set split precisely once as illustrated in the figure 3.2. We set k to a commonly used value - 10. Additionally, we repeat the cross-validation procedure twice (after reshuffling the data), which adds further robustness to the obtained results and also to the following conclusions based on the results. The random generator seeds in the implementation are fixed in such a way, that for each of the algorithms the same data splits are used.

We also perform tuning of the model hyper-parameters, where applicable by using the grid search approach. This means that for all the parameter combinations from the specified lists of values a model is trained and evaluated on a so-called validation set. The validation set consists of 20% of users in the training fold (hyper-parameter tuning is therefore done for each of the cross-validation folds). We of course still respect the order of ratings as previously explained. The metric considered for hyper-parameter tuning is top k precision with k=3 and for path prediction, the target metric is the subsequence accuracy.

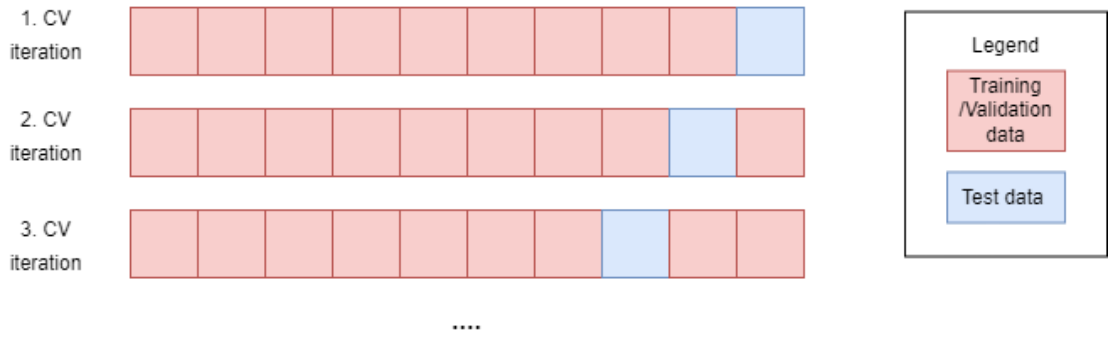

Figure 3.2: First three iterations of 10-fold cross-validation procedure

3.1.3 Statistical significance testing

To support our conclusions about the superiority of certain approaches over the others, we test the obtained metric results for statistical significance. The testing allows us to test the hypotheses about the means of the metric, in other words, we test whether the mean of an algorithm is significantly higher than the mean of another algorithm's metric. There are several types of statistical tests existing, each with different assumptions about the data. One of the common approaches for comparing the performance of two machine learning algorithms is Student's paired t-test. Paired tests make use of the fact that in cross-validation, the results are paired because they were calculated on the same splits of the data. However, the issue with using Student's paired t-test in combination with cross-validation, is that the training sets are not completely independent from each other, which violates one of the test's assumptions which increases type one error probability (which means a null hypothesis is rejected, even though it is true) [Die98].

Another alternative that arguably overcomes this issue is to use a 2-fold cross-validation repeated five times [Die98]. In our setting with a small-sized dataset, would a 2-fold CV leave us with too small amount of training data (also taking into account, that another fraction of the training fold is always used as a validation set).

We, therefore, opt for a non-parametric test - specifically the Wilcoxon signed-rank test which is a non-parametric version of the Student's paired t-test. The test does not assume independence of the samples such as Student's paired t-test and it is a recommended alternative if the t-test assumptions are not met [Dem06]. The calculation of the test statistic consists of calculating differences between the measurements, assigning ranks to absolute values of these differences (one for the smallest, two for the second smallest, etc.) and then the ranks of the differences that had positive and the ones that had a negative sign are summed separately. The smaller of these two sums is the test statistic which is then used for finding the p-value. We are interested in one-sided test results which tell us, whether the mean of the first sample is significantly higher than the mean of the second sample. As significance level alpha we choose a commonly used value of 0.05.

3.1.4 Rating prediction metrics

For the rating prediction metrics, we selected root mean squared error (RMSE) and ranking metric precision at k (with k being set to 3/6/9). These are both some of the most popular and useful metrics in this context [RRS10]. The RMSE metric is defined as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(predicted_i - actual_i)^2} \tag{3.1}$$

Where n represents the number of test set ratings, $predicted_i$ represents the i-th predicted ratings and $actual_i$ represents the actual i-th rating given.

The precision at k metric is defined as following:

$$\text{Top k precision} = \frac{\#\text{relevant}}{k} \tag{3.2}$$

Where $\#relevant$ represents the number of relevant items among the top k items according to the predicted ratings of the user. An item is considered relevant when it is also present in the top k items of a list being sorted according to the actual ratings given by the user.

We focus on the precision at k metric in this work. The reason is, this metric represents the real-life future use case where the ratings would be used to select a subset of exhibits for users who do not wish to visit all the remaining exhibits. The order of the items within the top k items is not relevant for us. This is not an issue in this setting, because the order in which the items will be visited is determined by the physical layout of the exhibition. The idea is, that the shortest path that visits the recommended subset of the items would then be recommended.

3.1.5 Path prediction metrics

As path prediction metrics, we have selected subsequence accuracy and presence accuracy. These metrics were already used in another work proposing a path prediction approach in a similar setting [PGC+20] and they represent well the goal of path prediction in our setting. Subsequence accuracy is defined as follows:

$$\text{Subsequence accuracy} = \frac{\sum_{i=1}^{n} indicator_i}{n} \tag{3.3}$$

Where $indicator_i$ is an indicator vector, in which the i-th element has a value of 1 if the i-th element of predicted items and i-th element of actual visited items equal, otherwise the i-th value is 0. The $indicator$ vector has length of n.

Presence accuracy is a more relaxed version of subsequence accuracy. It is defined as:

$$\text{Presence accuracy} = \frac{|P \cap A|}{n} \tag{3.4}$$

Where P denotes the set of n predicted items, A denotes the set of actual n items according to the real users' trajectory, $|\cdot|$ notation denotes the set size and n denotes the number of predicted items.

The main focus for the path prediction task is the subsequence accuracy since more exact predictions allow us to redirect users more appropriately in order to avoid overcrowding of exhibits.

3.2 Rating prediction approaches

The following subsection explains in detail all of the rating prediction approaches which are being tested using our evaluation framework.

3.2.1 Baselines

In addition to rather complex algorithms and machine learning models, we compare the algorithms with several simple baseline approaches, to see whether they outperform the simple baseline approaches.

Average item rating

The rating of an item is not personalized for a user, it is always calculated as the average from all the training data ratings for that item.

Average user rating

For a user, we always predict the same value calculated as average from all her/his training set ratings. This approach is not evaluated in terms of the ranking metrics since all the items have the same prediction and therefore also the same rank.

User item deviation baseline

This baseline is inspired by the matrix factorization formula [RRS10]. We only consider the overall average rating and user and item biases, which are not learned in this case, but instead simply estimated. The User item deviation baseline is defined as:

$$\tilde{r}_{ui} = \mu + b_i + b_u \tag{3.5}$$

Where $\tilde{r}_{ui}$ represents the predicted rating for user u and item i. μ represents the overall average rating, calculated as average from all the ratings in the training data. b_i represents the item bias, which is calculated as the difference between the overall average rating and an average rating of the item based on the training data. b_u represents the user bias and is calculated as the difference between the overall average rating and average user rating based on training data.

3.2.2 User-user collaborative filtering

The idea of user-user collaborative filtering is based on the idea, that users with similar previous preferences, will also have similar preferences regarding the future items. There are multiple options how to define user similarity, with Pearson correlation coefficient being the most commonly used approach [RRS10]. The similarity of two users is defined as:

$$w_{uv} = \frac{\sum_{i \in I_u \cap I_v} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_u} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_v} (r_{vi} - \bar{r}_v)^2}} \tag{3.6}$$

Where w_{uv} is the resulting similarity between users u and v, I_u and I_v is the set of items rated by users u and i respectively, r_{ui} and r_{vi} are ratings of item i by users u and v respectively and finally $\bar{r}_u$ and $\bar{r}_v$ represent the average rating of users u and v respectively.

The final predicted rating is then calculated as:

$$r_{ui} = \bar{r}_u + \frac{\sum_{v \in N(u)} w_{uv}(r_{vi} - \bar{r}_v)}{\sum_{v \in N(u)} |w_{uv}|} \tag{3.7}$$

Where the term definitions are the same as in 3.6. Additionally, $N(u)$ represents the neighborhood of a user u. The neighborhood consists of most similar users, according to the similarity defined in 3.6. The size of the neighborhood (how many most similar users are considered) is treated as a hyper-parameter. Additionally, other hyper-parameter controls whether we also consider the most dissimilar users (highest absolute similarity values). The intuition is, that a user will like an item a dissimilar user dislikes and vice versa.

3.2.3 Item-item collaborative filtering

The idea of item-item collaborative filtering is similar to user-user collaborative filtering. We assume that a user will like items similar to the ones he liked previously, dislike items dissimilar to the ones he liked previously and vice versa [RRS10]. The similarity between item i and j is defined as:

$$w_{ij} = \frac{\sum_{i \in U_i \cap U_j} (r_{ui} - \bar{r_i})(r_{uj} - \bar{r_j})}{\sqrt{\sum_{u \in U_i} (r_{ui} - \bar{r_i})^2} \sqrt{\sum_{u \in U_j} (r_{uj} - \bar{r_j})^2}} \tag{3.8}$$

Where w_{ij} is the resulting similarity between items i and j, U_i and U_j is the set of users that have rated item i and j respectively, r_{ui} and r_{uj} are ratings of user u of items i and j respectively and finally $\bar{r_i}$ and $\bar{r_j}$ represent the average ratings of items i and j respectively.

The final predicted rating is then calculated as:

$$r_{ui} = \bar{r_i} + \frac{\sum_{v \in N(i)} w_{ij}(r_{uj} - \bar{r_j})}{\sum_{j \in N(i)} |w_{ij}|} \tag{3.9}$$

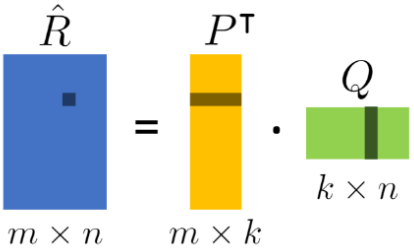

Figure 3.3: Predicted rating matrix $\hat{R}$ in matrix factorization approach represented as inner product of abstract matrices P^T (where each row represents features of a certain user) and Q (where each column represents abstract features of an item) [Sac20]

Where the term definitions are the same as in 3.8. Additionally, $N(i)$ represents the neighborhood of an item i. The neighborhood consists of most similar items according to the similarity defined in 3.8. The size of the neighborhood (how many most similar items are considered) is treated as a hyper-parameter. Additionally, other hyper-parameter controls, whether we also consider the most dissimilar users (highest absolute similarity values). The intuition is that a user will like an item dissimilar to another item he previously disliked and vice versa.

3.2.4 Matrix factorization

Matrix factorization is a model-based collaborative approach. This means we aim to learn abstract user features represented by a learned matrix P^T (where each row represents features of a certain user) and learned abstract item features represented by matrix Q (where each column represents abstract features of an item) [KBV09][Sac20][RRS10]. This is illustrated in the figure 3.3, where the matrix of predicted ratings $\hat{R}$ with m users, n items is represented as an inner product of user feature matrix P^T and item feature matrix Q. Dimension k represents the number of abstract features to be learned.

Additionally to the learned features, user and item biases are also learned. The predicted rating is defined as:

$$\tilde{r}_{ui} = \mu + b_i + b_u + p_u^T + q_i \tag{3.10}$$

Where μ is the overall average rating from all ratings, b_i is the item bias, b_u is the user bias, q_i is a vector of latent dimension k (hyper-parameter) which represents abstract description of item i and p_u^T is a vector representing abstract features of a user u. The error function, which we aim to minimize is defined as:

$$e_{ui} = r_{ui} - \mu + b_i + b_u + p_u^T + q_i \tag{3.11}$$

We minimize this error using a stochastic gradient descent algorithm. This means, we loop through the training ratings and in each iteration, we slightly modify the parameters in the opposite direction of the gradient of the error function for a set number of iterations.

3.2.5 Regularized kernel matrix factorization

This approach extends the idea of standard matrix factorization, but the interactions between the user vector w_u and item vector h_i are kernelized (meaning the predicted rating is calculated using so-called kernel function instead of the dot-product) [RST08]. The selection of the kernel function is treated as a hyper-parameter. The tested kernel functions are:

Linear:

$$K_l(w_u, h_i) =< w_u, h_i > \tag{3.12}$$

Polynomial:

$$K_p(w_u, h_i) = (1+ < w_u, h_i >)^d \tag{3.13}$$

Radial basis function (RBF):

$$K_r(w_u, h_i) = exp(-\frac{||w_u - h_u||^2}{2\sigma^2}) \tag{3.14}$$

In order to avoid over-fitting, the optimal fit of abstract user and item features matrices is not learned. Instead, we add a regularization term, which is added to the error function, which results in higher values in the two matrices being penalized. The strength of the regularization is controlled by the hyper-parameter λ.

3.2.6 Demographic-based SVR/histogram gradient boosting

The approach is based on the previous work in the related tourism domain (Tripadvisor.com dataset), where they have used demographic user characteristics as features for a supervised machine learning algorithm, which aims to predict the ratings of tourist attractions [WCN12]. In the previous work, they have treated the problem as a classification problem and used a support vector machine classification algorithm. In our dataset, we also have half ratings (.5 values), which are a lot less common than integer value ratings. This would cause that for some exhibits we would have only very few samples of certain classes (rating values) which would most likely not be sufficient for a machine-learning algorithm to learn to generalize well. The combination of a few very under-sampled classes, a higher number of total classes to predict and a generally smaller dataset make classification an unfeasible approach in our case.

Instead, we treat the prediction problem as a regression task. We simply train a separate regression algorithm for each of the exhibits. Inspired by the original work, we use support vector machine regression. Additionally, we also try a powerful tree-based ensemble method, called histogram gradient boosting. The approach is purely based on the demographics and no previous ratings are taken into account.

3.2.7 Demographic and collaboration histogram gradient boosting

The approach is basically the same as the previously described purely demographic-based approach. The only difference is, that we also add predicted ratings from one

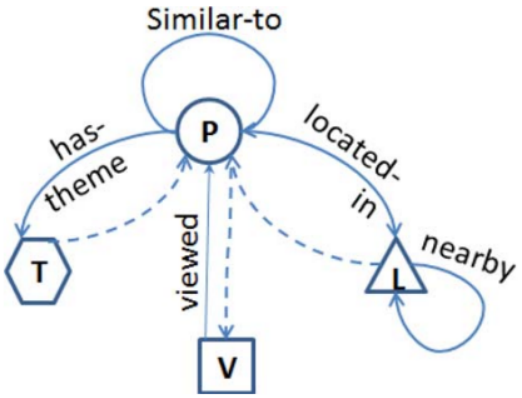

Figure 3.4: Types of nodes and possible relationships in the museum graph used in the original "Graph based recommender" - presentations (P), positions (L), themes (T), and visitors (V) [MKK17]

of the collaborative approaches as an additional feature for the supervised regression machine learning algorithm. The idea is, that a machine learning algorithm could possibly combine the predicted ratings with some useful additional information captured by the demographic data. As the collaboration method, we have selected regularized kernel matrix factorization due to its performance.

First, the regularized kernel matrix factorization is trained, as usual, then the ratings are predicted for the training set. These predicted ratings serve as an additional feature for the regression algorithm, which aims to predict the final ratings in the training set. Similarly, at prediction time we first use the trained matrix factorization to predict the test set ratings and then these ratings are used together with the demographic features in the regression algorithm. The algorithm then predicts the final test set ratings.

3.2.8 Graph-based recommender

Graph-based recommender rating prediction approach is based on previous work in the domain, where they have managed to represent the museum as a graph data structure [MKK17]. Then the idea is to use the personalized page rank algorithm to express the relevance of the exhibit nodes to a user. In the original work, more complex information regarding the exhibits is available. The possible types of nodes were "presentation","visitor","theme","location". The types of nodes and relationships between these nodes are shown in the figure 3.4. The term presentation is slightly different from our case. In this work, an exhibit can have multiple presentations in the application. A presentation is multimedia content complementing the exhibit.

In our more general setting, we do not have such information about the exhibits, therefore we limit our graph structure to only the "visitor" type of graph node. Similarly, we also do not use the "similar-to" relationship connecting different presentation/exhibit nodes.

Therefore, we construct a graph where the visitors are represented as one type of node and the exhibits as another node type. These types of nodes are connected bidirectionally by the "viewed" edge type whose weight corresponds to the ratings given by the visitors.

The prediction procedure consists of using the Personalized page rank (PPR) algorithm, where the node weights are set to the training set ratings given by the user. The resulting PPR values for the exhibits not rated yet (nodes in the graph) are the predictions. The algorithm is very similar to the original Page Rank algorithm, where a "random surfer" browses the pages (graph nodes) with the node transition probability uniformly distributed across the outgoing edges. Additionally, at each step, there is a certain probability when the surfer teleports to another node with uniform probability across all nodes, from where the process then continues. During the process, we keep track of the ratio, expressing the proportion of visits of each node to all the steps taken (total node visits). The process is repeated until convergence. The difference in PPR is the restart operation probabilities are not uniform but custom. In our case, they correspond to the ratings of the user we are making predictions for.

The authors have also weighted the different types of graph edges, which means the probability of choosing an outgoing edge at each step is not uniform, but instead proportional to the weights. The intuition is that different types of relationships have different levels of importance. They have tuned the edge weights for different edge types using an exhaustive search. In our case, we only have a single edge type therefore this approach would not make sense in our setting. Instead, we have experimented with weighting the edges with weights corresponding to the exhibit ratings led by the intuition, that connections with higher ratings are more important. Both variants - with and without weighted edges are presented in the results section.

The range of the predicted values is arbitrary and does not follow the original 0-5 scale. This is of course perfectly fine for the ranking metric purpose (which is our main focus). To be also able to compare this algorithm with others in terms of RMSE as well, we experimented with two possible ways of normalization of the values to the real rating range [PS15]. The first normalization approach is simple min-max normalization which is defined as follows:

$$normalized_i = \frac{r_i - r_{min}}{r_{max} - r_{min}} * (s_{max} - s_{min}) + s_{min} \tag{3.15}$$

Where $normalized_i$ is the resulting normalized i-th rating, r_{min} is the minimum rating value, r_{max} is the maximum rating value, r_i is the predicted rating value before the normalization, s_{min} is the minimum value of the desired scale and s_{max} is the maximum value of the desired scale. In our case if s_{min} set to 0 and s_{max} set to 5.

The other normalization approach we have experimented with, consists of normalizing the predictions to the same mean and standard deviation as have the ratings in the

training set. This is done as follows:

$$normalized_i = r_{mean} + (prediction_i - prediction_{mean}) * (\frac{r_{std}}{prediction_{std}}) \tag{3.16}$$

Where $normalized_i$ is the resulting normalized i-th rating, r_{mean} is the training set mean, r_{std} is the standard deviation of the training set, $prediction_i$ is the i-th predicted rating, $prediction_{mean}$ is the mean value from the predictions and $prediciton_{std}$ is the standard deviation of the predicted values.

The first approach performed very poorly in our experiments, therefore it is not shown in most of the plots for better readability.

3.2.9 GLocal-K

GLocal-K [HLL+21] is the current state-of-the-art approach in the field of recommender systems for a popular MovieLens 1M Dataset [Mov]. This approach uses only the rating data, without any additional information. The approach is based on neural network architecture called auto-encoder, which learns a simplified representation of the inputs a tries to reconstruct the inputs in the network outputs. GLocal-K consists of two stages - the authors call the first one "Auto-Encoder Pre-training" and the second one "Global kernel-based Rating Matrix". In the first stage, the inputs to the auto-encoder are the rating vectors for each item r_i and the neural network tries to reconstruct the vector denoted as $r_i^{'}$ which is defined as follows:

$$r_i^{'} = f(W^{(d)}g(W^{(e)}r_i + b) + b^{'}) \tag{3.17}$$

Where f an g are the activation functions, $W^{(d)}$ and $W^{(e)}$ are the weight matrices (parameters of the neural network), b and $b^{'}$ are the bias vectors. The weight matrices are then reparametrized by a 2d-RBF kernel called local kernelized weight matrix defined as follows:

$$K_{ij}(U, V) = max(0, 1 - ||u_i - v_j||_2^2) \tag{3.18}$$

The changed weight matrix called local kernelized weight matrix $W_{ij}^{'}$ is then defined as Hadamard-product of weight and kernel matrices defined as follows:

$$W_{ij}^{'} = W_{ij} \cdot K_{ij}(U, V) \tag{3.19}$$

The second step of GLocal-K consists of fine-tuning the pre-trained auto-encoder, with a rating matrix produced by a global convolutional kernel. The predicted vectors for items from the first phase are average pooled which is denoted as:

$$\mu_i = avgpool(r_i^{'}) \tag{3.20}$$

These values then play a role as weights of multiple kernels $K = k_1, k_2, ..., k_m$. The number of users is denoted as m. These kernels are aggregated using the inner product operation producing the so-called global kernel denoted as GK:

$$GK = \sum_{i=1}^{m} \mu_i \cdot k_i \tag{3.21}$$

Using this global kernel, we construct a global kernel-based rating matrix defined as:

$$\hat{R} = R \otimes GK \tag{3.22}$$

Where $\otimes$ represents a convolution operation, R is the initial rating matrix with m items and n users. This matrix is then used as an input to the pre-trained auto-encoder in the fine-tuning stage. Finally, the outputs of the auto-encoder in the fine-tuning stage are the final predictions of GLocal-K.

This whole process is illustrated in the figure 3.5.

3.3 Path prediction approaches

The following subsection explains in detail all the path prediction approaches, which are being tested in our evaluation framework.

3.3.1 Nearest unvisited exhibit baseline

The nearest unvisited exhibit baseline is a simple naive approach, that assumes the visitor at each point simply moves to the next physically closest exhibit, which she/he has not visited yet. There are two reasons for considering only unvisited exhibits. The main one is that the application used for data collection guarantees that each exhibit is visited only once and the second reason is, that without this restriction the algorithm would loop infinitely in many cases.

3.3.2 Markov chain

Markov chain is a sequence of states which satisfy two conditions. The first condition is usually called Markov condition and means that if "the system is in state i at time k, then the probability that it will change to state j at time $k + 1$ does not depend on what happened in earlier times $k - 1$, $k - 2$, ..., 0" [CLM12]. In other words, the current state depends only on the directly previous state not on the other states further in the past. The second condition says that "if the system is in state i at time k, then the probability that it will change to state j at time $k + 1$ does not depend on k" [CLM12]. In our setting, this means that it does not matter at what depth in the exhibition the visitor is, the transition probabilities stay the same.

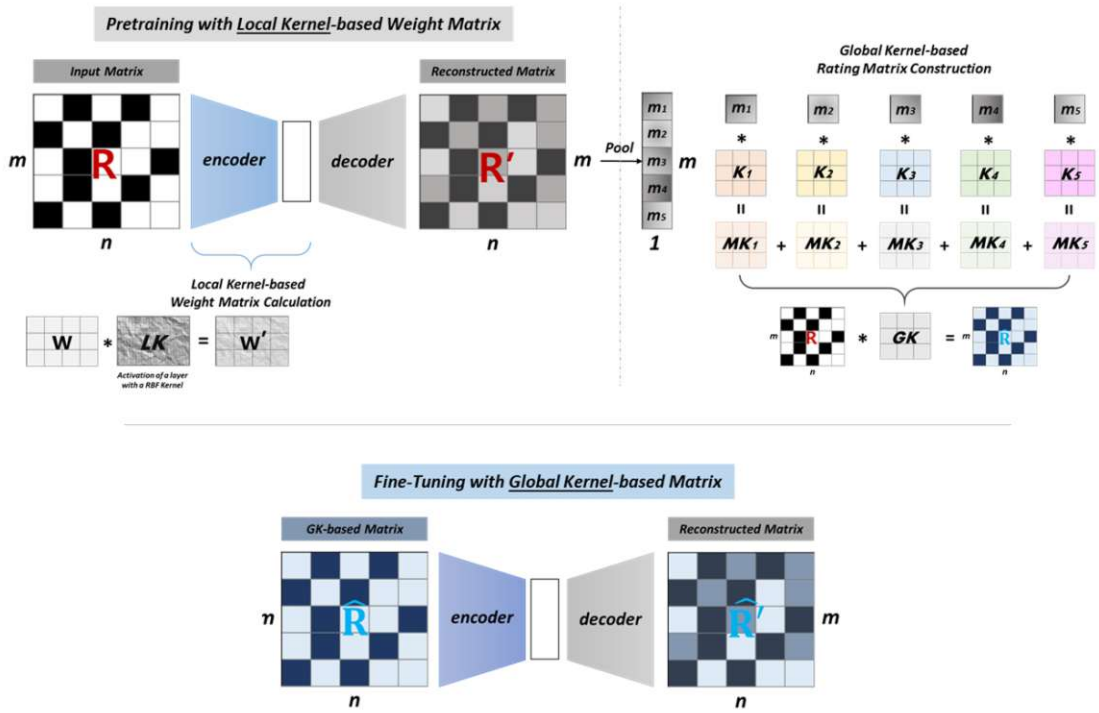

Figure 3.5: Overall architecture of GLocal-K - in the first stage the auto-encoder is pre-trained with a local kernelized weight matrix, in the second stage the auto-encoder is fine tuned with global kernel-based rating matrix which produces the final predictions [HLL+21]

The transition probability between the states 1 and 2 for example (exhibits in our case), is calculated as the number of transitions from state 1 to state 2 divided by the total number of transitions from state 1 to any state.

Additionally, in this setting, we consider for prediction only the states, that the user has not visited yet, similarly as in "Nearest unvisited exhibit baseline" in section 3.3.1. The reasons are again, that the application used for data collection allows to visit each exhibit only once, and secondly, without this restriction, the predicted sequence would often loop infinitely between two states. Therefore, when selecting the next visited exhibit, we first filter only the unvisited exhibits and select the one with the highest transition probability from the current state. In case all the items for which the probability is available (meaning there was at least one transition for the particular state combination), we select a random state from the rest of the unvisited states. If the end state (-1) is predicted, the prediction sequence ends. Technically, with this modification, this approach does not satisfy the full definition of the Markov chain since in a way the prediction is based on the states further in history. However, predicting already visited exhibits does not make sense in this setting as already explained.

3.3.3 Interest transition hybrid model

This approach is based on previous work in a similar setting, where the data was collected manually [BZB+08]. The idea is to create a hybrid sequence prediction approach by combining the Markov chain and predicted interest in the items. In the original work, the interest was estimated based on the time the visitors spent with the exhibits since the data did not contain explicit ratings. Then they have used collaborative filtering to predict the future implicit ratings.

In our setting, we have explicit feedback available, therefore instead of estimating the user interest by time spent with the exhibits we directly use the ratings. As the rating prediction model, we have selected the regularized kernel matrix factorization, which is in general the best performing approach in terms of RMSE. Both the Markov chain transition probability and the predicted ratings are normalized to the same range using min-max normalization. The normalized values are then weighted and combined, which results in the final next exhibit prediction probability. The weights are treated as a hyper-parameter.

As the authors have done in the original paper, we also implement two approaches for sequence prediction, which are treated as hyper-parameter. The first one is called "TopK Prediction". In this approach we simply take top k exhibits we aim to predict (three in our case), according to the combined probabilities, from the point of view of the last visited exhibit. The second approach is called "SequenceK". In this approach, we predict the next exhibits one by one - in other words, we take the one with the highest probability from the point of view of the latest visited exhibit. In the next step, we take the one with the highest combined probability, from the point of view of the last predicted exhibit and we repeat the process until we predict the desired number of items or we reach the end state.

3.3.4 Deep learning sequence to sequence - states/transitions

This approach is based on a deep learning sequence to sequence neural network architecture (also often called seq2seq) and was used in previous work in a similar setting on data collected from IoT sensors [PGC+20]. This architecture is a common solution to the language translation problem, where we use a sequence of words in one language (usually called tokens) as an input to the network and we try to predict a sequence of words in the target language. The process is very similar in our setting, but instead of words, our tokens represent the exhibits in a certain way.

In the original work, the authors have used the transitions between exhibits as input and output tokens. For example, the sequence of exhibits 1,3,7 would be encoded as three tokens - (1,1),(1,3),(3,7). The output tokens are encoded in the same way. This approach is in the result section called "Deep learning seq2seq transitions". As an alternative, we have proposed to use a simpler token encoding - instead of transitions we simply use each exhibit id number as a distinct token within the input and output sequences. The

intuition is, the simpler encoding with less possible distinct inputs and outputs might be more suitable for smaller datasets such as ours.

The sequence to sequence neural network architecture consists of two parts - encoder and decoder. This architecture is illustrated by a language translation task example, in the figure 3.6 by [ben18]. The input tokens (German words in this example) are fed through the yellow embedding layer and the embeddings are then used as inputs to the decoder recurrent neural network (RNN). Special token "<sos>" represents the start of the sequence and "<eos>" is a special token representing the end of the sequence in this example. In this case, the RNN consists of Long Short-Term Memory units (LSTM). The inputs to these units are the token embeddings, as well as the hidden states from the previous token. The final hidden state is then passed to the decoder as a dense representation of the input sequence (the outputs from the encoder are discarded). In the decoder, the initial "<sos>" token is first fed to the RNN, together with the encoder's final hidden state which gives us the first predicted word of the target sequence as the output. Then this word is used as an input to the next unit together with the hidden state from the previous step. This process is repeated until the "<eos>" token is generated.

As in the original work our network consists of 128 hidden neurons in the encoder and decoder layers. Additionally, we also try 64 and 256 neurons as a hyper-parameter evaluated on the validation set (due to relatively long training time the other parameters are fixed). The rest of the parameters were the same as in the original work - l_2 regularization of $1 * 10^{-4}$ applied to the layers, Adam optimizer with a learning rate of $1 * 10^{-3}$, categorical cross-entropy loss function, batch size of 16 and usage of Keras EarlyStopping function as stopping criteria with patience level set to five and the number of epochs is set to 500.

3.3.5 Deep learning sequence to sequence with demographics

In another proposed modification of the original work [PGC+20] in addition to the alternative token encoding (states instead transitions), we have tried to add additional tokens representing the user demographics to the input. This idea was led by intuition, that different visitors of different demographic groups behave differently in the museum. Each unique value across all the demographic attributes is assigned a distinct letter. For example, the six distinct values of attribute age are assigned letters from "a" to "f". The second gender attribute which has two distinct values, gets assigned letters "g" and "h". The process is the same for the other attributes. The demographic attributes encoded in this way are then attached to the beginning of the input sequence, always in the same order. In other words, the first token always represents the attribute age, the second token always represents attribute gender, etc.

The proposed modification of added demographic tokens is tested in combinations with both examined encodings of the sequence of the exhibits (states/transitions).

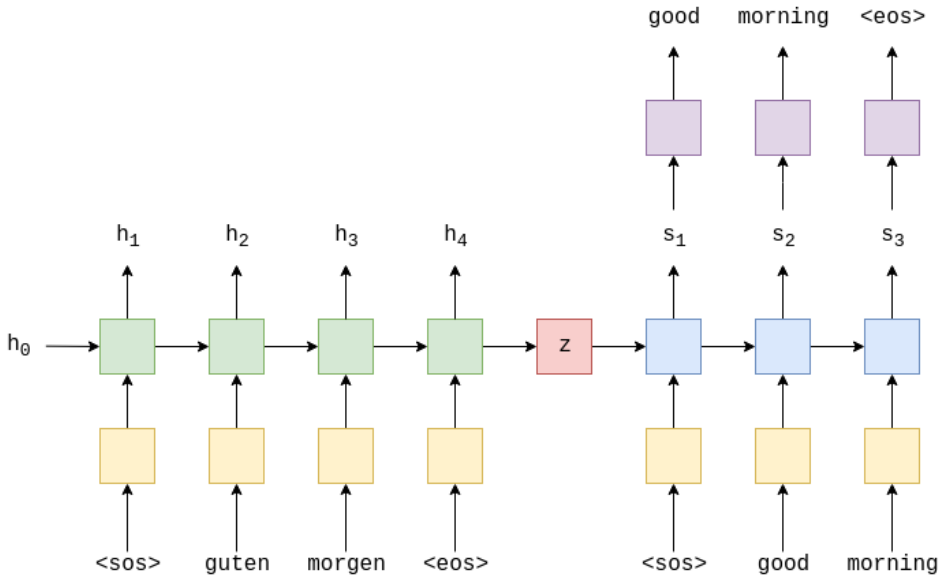

Figure 3.6: Architecture of a sequence to sequence network illustrated by language translation example [ben18]

3.4 Implementation overview

All the implementation was done in the Python programming language. The main libraries used were the standard libraries commonly used for data science tasks such as scikit-learn [PVG+11], Numpy [HMvdW+20], Pandas [pdt20], Scipy [VGO+20], and also more specialized libraries used in some of the algorithms such as Networkx [HSSC], and Tensorflow [AAB+15]. There was no code available for most of the algorithms, so the majority of them were implemented, based on the descriptions in the corresponding papers. For some of the approaches, such as the regularized kernel matrix factorization and GLocal-K the authors provide Python implementations, which were incorporated in our framework.

The algorithms are implemented in separate Python files for the two problems. Each algorithm is represented as a class that has public functions “get_algorithm_name” which return the algorithm’s name, “set_parameters” which sets the hyper-parameters, “train” which trains the algorithm based on the training set, and “predict” which computes the results for the provided test set. All the algorithm classes are child classes of an abstract parent class called “RatingPredictor”/“PathPredictor” respectively. These classes implement the cross-validation procedure (including the visitor path depth cutoffs), hyper-parameter optimization, and call the previously mentioned public functions of the individual algorithm classes.

The “cross_validation” function is then iteratively being called on instances of the various algorithm classes, for different visitor path depths in individual Jupyter notebooks, for

the two tasks. The Jupyter notebooks read the data, either directly from the database of the Kouzan application based on the connection properties specified in a configuration file, or alternatively, the data can be read from a CSV prepared from the database data, which is refreshed each time we connect to the database. The notebooks contain several configuration options - the user sub-sample ratio to be used, for the path prediction, we can set a flag that indicates whether the small rooms should be aggregated or not, and in each notebook we can set a flag for each algorithm, which indicates whether we wish to recompute the results or look for the results in a pre-computed results file. The notebooks contain various visualizations of the data and the results, which are all saved in corresponding folders. Because two approaches in the path prediction task require different mutually incompatible Tensorflow library versions, this notebook was split into two notebooks where each of them needs to be executed using a different virtual environment.

CHAPTER 4

Results

The following chapter presents the results of the evaluation of all the approaches for rating and path prediction problems, which were presented in the previous chapters. As explained previously, we test all the algorithms at different visitor path depths (the number of exhibits the test set user already rated/visited). To be able to easily analyze this dimension across the numerous evaluation metrics, we have plotted the resulting average metric values from the cross-validation iterations as a line chart, where the x-axis represents the visitor path depth and the y-axis represents the metric value. The different algorithms are represented by different line colors, additionally annotated by various geometric symbols for better readability due to the relatively high number of approaches being visualized. The conclusion from this results chapter is further summarized in chapter 5, in particular within the context of the initial research questions defined in chapter 1.

4.1 Rating prediction

In the rating prediction task, it is not possible to use the average user rating baseline for ranking metrics, therefore this approach is omitted in the case of ranking metrics (all the items would have had the same rank). For the graph-based recommender approach, the two normalization variations result in the same ranking metric values, therefore only one of them is shown (because normalization preserves the item rankings). In the case of the figure showing the RMSE metric, the min-max normalization for graph-based recommender is omitted for clarity of the figure, since the RMSE values are a lot higher than the other algorithms.

In the line chart showing precision at k=3 metric in the figure 4.1, we can see the GLocal-K algorithm performs very well compared to the other approaches at visitor path depths of 8 and 12. At visitor path depth of 4, where we do not have as much data about the test user yet, we can see this algorithm fails to outperform other approaches. This is

not surprising, since the approach was originally designed for a much larger dataset with more items. It is arguably a bit unexpected that it outperforms the other approaches at depths 8 and 12 due to the same argument.

Other approaches seem to fail to noticeably outperform the best performing baseline - user-item deviation baseline. Additionally, in the case of the lowest visitor path depth of 4, no approach seems to outperform all the baselines, including the demographic-based approaches, which do not rely on user rating history. The demographic-based approaches, unfortunately, do not perform very well in general - none of the two variants seems to be a consistently better option than the other. It is most likely the case, that the simple demographic characteristics which we have collected are not sufficient to identify users' preferences. The proposed modification of the approach, consisting of feeding in the predicted ratings as an extra feature for the demographic-based approach, was not successful and did actually make the machine learning model perform worse than its counterpart without this additional feature.

The graph-based recommender approach which we considered as a state-of-the-art solution also performs rather poorly at all visitor path depths. The reason is most likely the necessary approach simplification, compared to the original work. In the original work, the approach was outperforming all other tested approaches due to the integration of various sources of information about the exhibition (based on data that we do not have). Also, none of the two variants of the approach (with or without graph weights) seems to consistently outperform another.

According to the statistical significance testing for precision at k=3 at depth of 4, we can conclude the GLocal-K algorithm is not significantly better than all the baseline approaches, at a significance level of alpha=0.05. Neither is any other tested approach.

At depth of 8, we can see the GLocal-K performs better than all the other approaches, according to the statistical significance testing at significance level alpha=0.05, as we can see in the figure 4.3. All the other approaches fail to show a significant difference in performance when compared to all the other algorithms.

Statistical significance testing at depth of 12, shows the better average performance for precision at k=3 shown in the figure 4.1 of GLocal-K is not significantly higher than all the other approaches. More specifically, it fails to significantly outperform average item rating baseline, demographic-based histogram gradient boosting, matrix factorization, item-item CF, and user-item deviation baseline.

When we take a look at the rating prediction results for precision at k=6 metric in the figure 4.1, we can see no approach is noticeably dominating, as in the case of precision at k=3. In the case of visitor path depth of 4, the highest average value is reached by regularized kernel matrix factorization. However, it is not significantly better than all the other approaches - namely, the two which are not outperformed are average item rating baseline and user-item deviation baseline. At a visitor path depth of 8, regularized kernel matrix factorization is again the best approach based on the mean metric value. Once again, it is not significantly better than all the approaches at significance level

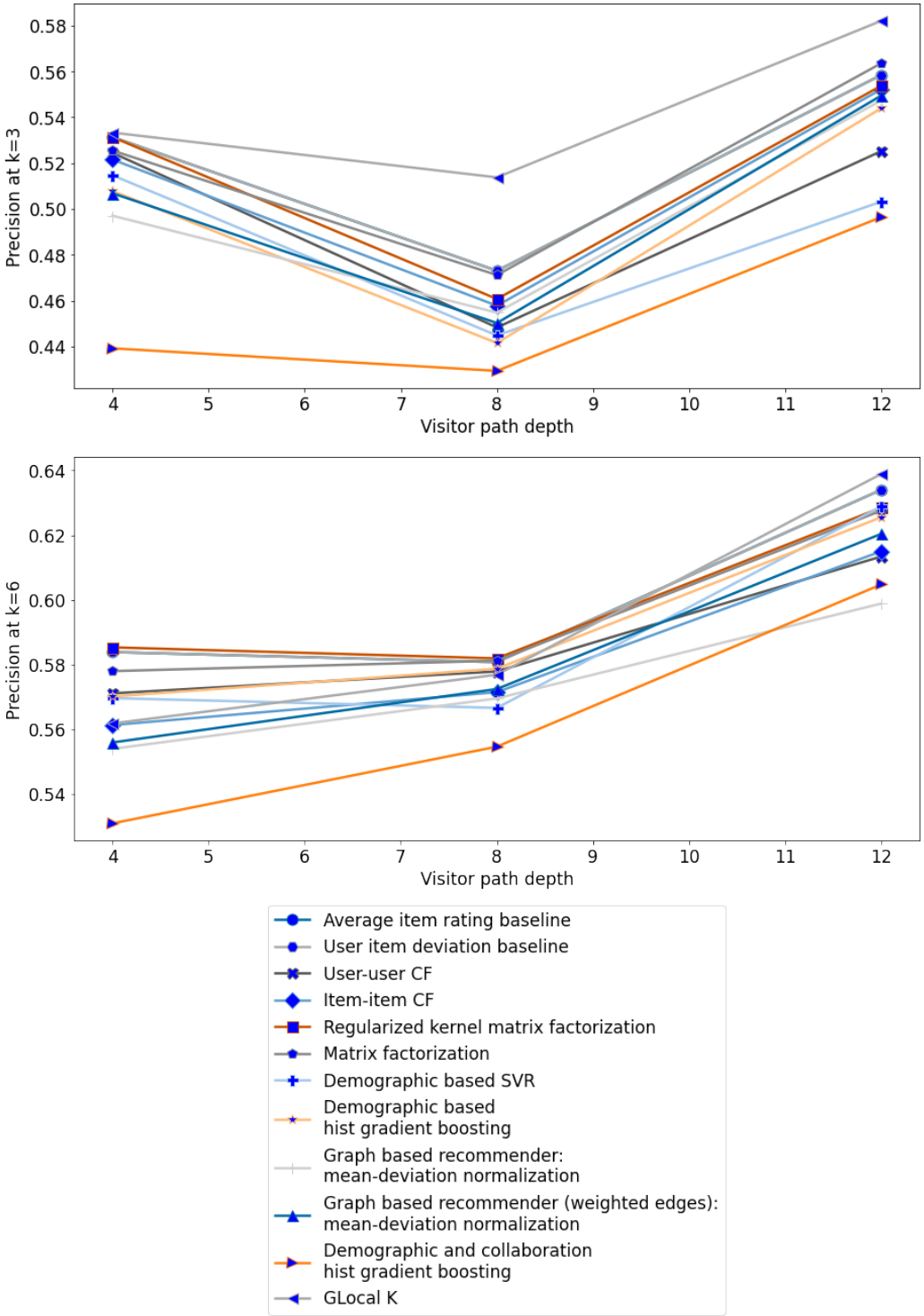

Figure 4.1: Rating prediction evaluation using precision at k=3 and k=6 metrics based on average values from cross-validation at different visitor path depths

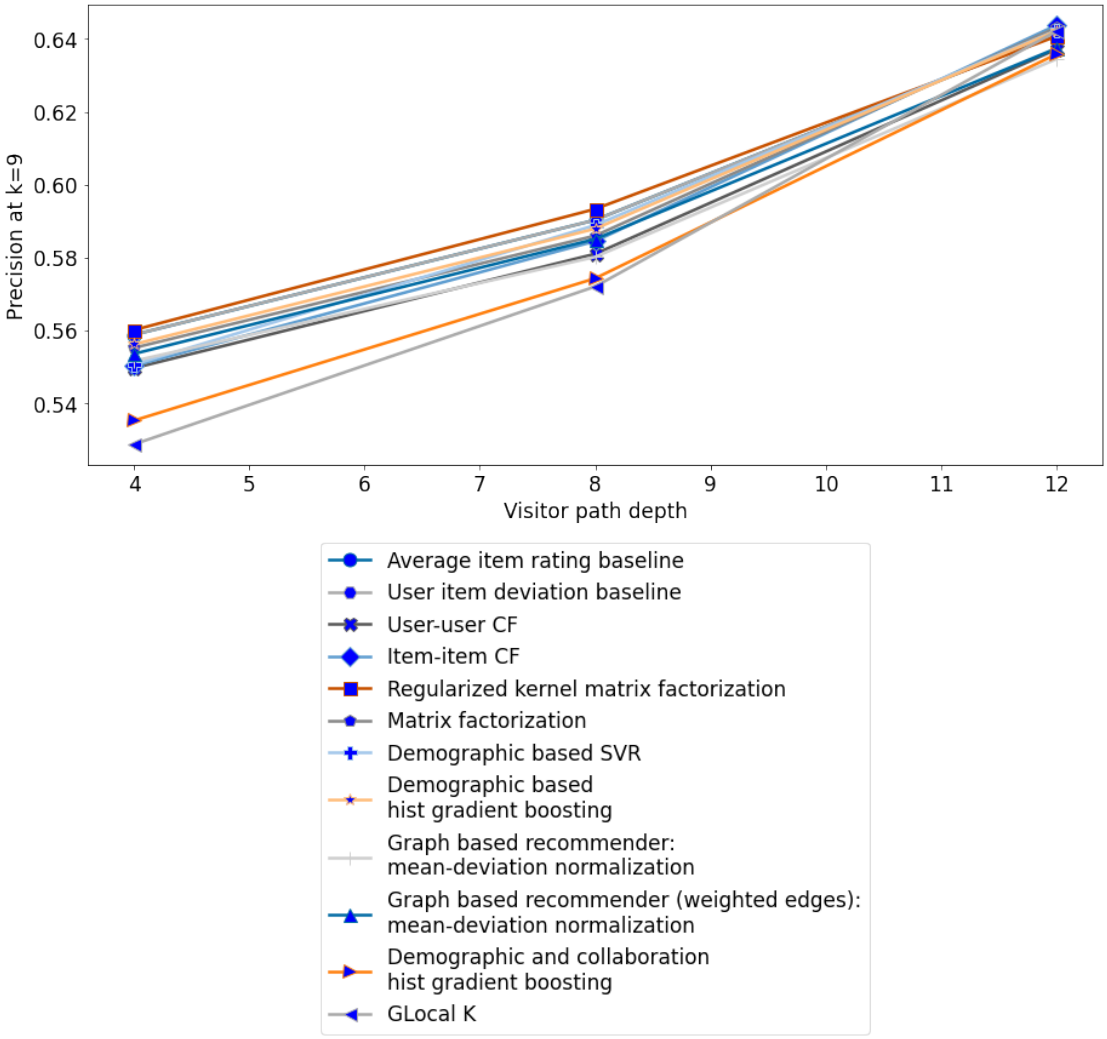

Figure 4.2: Rating prediction evaluation using precision at k=9 metric based on average values from cross-validation at different visitor path depths

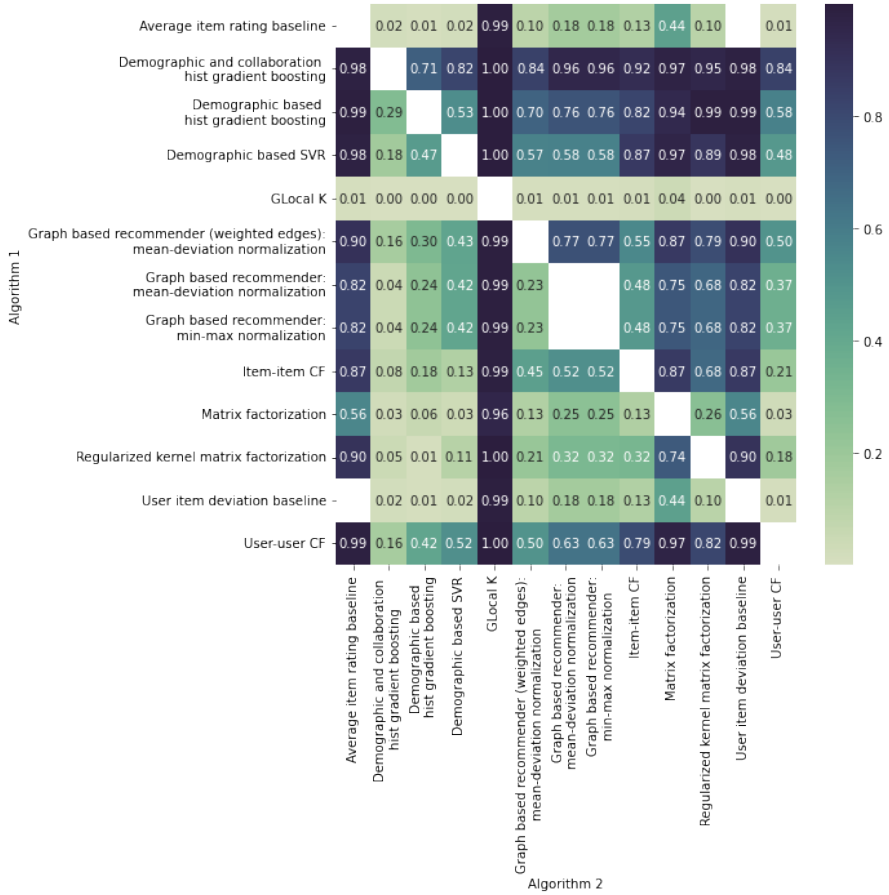

Figure 4.3: P-values from Wilcoxon signed ranked paired tests for the null hypothesis that the mean of the two algorithms is the same and hypothesis 1 that the mean of precision at k=3 for algorithm 1 is greater than metric of algorithm 2 at visitor path depth of 8

alpha=0.05. At a visitor path depth of 12, the GLocal-K algorithm manages to seemingly learn well the user characteristics determined by the larger amount of ratings she/he has given. Unfortunately, this difference is again not significant, when compared to some of the other approaches including the baselines.

For precision at k=6 metric, we can observe a relatively clear general trend of increasing algorithm performance with increasing visitor path depth. This pattern was also slightly present in the case of precision at k=3 in the figure 4.1, at least in the case of some algorithms. However, since we consider the ranking of more items this time, it is no surprise that the pattern is more stable with a higher value of k.

For the metric precision at k=9 in the figure 4.2, we can observe the average metric values are again even closer to each other, without any approach being clearly superior. At

Figure 4.4: P-values from Wilcoxon signed ranked paired tests for the null hypothesis that the mean of the two algorithms is the same and hypothesis 1 that the mean of precision at k=3 for algorithm 1 is greater than metric of algorithm 2 at visitor path depth of 12

visitor path depths of 4 and 8, we can notice the regularized kernel matrix factorization reaches the highest average values, with other approaches closely behind. The differences in performance between the algorithms get even smaller with higher visitor path depth. This is expected, since there are fewer items to choose from, with higher visitor path depth and higher k in precision at k metric. It not surprising the regularized kernel matrix factorization (and no other approach) is not better in terms of this metric, according to statistical significance testing at level alpha=0.05, when being compared to all the other algorithms. At depth of 4, it is not significantly better than the average item rating baseline, demographic-based histogram gradient boosting, graph-based recommender with weighted edges, and user-item deviation baseline. At depth of 8, it is despite the small difference in the mean metric value significantly better than all the other approaches, with the only exception which is the demographic-based SVR (where the p-value is 0.07).

At depth of 12, no approach is significantly better than all the others, according to the statistical significance testing.

If figure 4.5, we can observe which algorithms predict the numerical rating values with the lowest root mean squared error (RMSE). We can see that better numerical prediction does not always guarantee higher precision in rankings of the items. Specifically, the GLocal-K approach is not among the best-performing algorithms in terms of RMSE, despite dominating the ranking metric in some cases. At depths of 4 and 8, the regularized kernel matrix factorization reaches the lowest average RMSE values with standard matrix factorization closely behind. At depth of 12, the standard matrix factorization is slightly better in terms of the average metric value. Interestingly, the user-item deviation baseline is a lot closer to the best performing matrix factorization techniques at depth of 12. We can also notice, the only approach which does not directly predict the metric values, but instead predicts only arbitrary values purposed for item rankings (which is the graph-based recommender) has a lot higher RMSE values despite the normalization technique applied to the predictions. Another observation that can be made based on the RMSE figure 4.5, is that all algorithms perform slightly better with higher visitor depth. The exception is the demographic-based recommender, which is expected, since it is based only on the demographics and not on user rating history therefore there it does not benefit from more user ratings.

4.2 Path prediction

As explained in previous chapters there are two variations of the visitor path data. One at standard exhibit granularity and one where the three smaller rooms with exhibits located very close to each other are aggregated at room level, as explained previously in chapter 2 and illustrated by figure 2.12. Focusing on the version with aggregated rooms is more meaningful, considering the final use case (there is not much value in predicting occupancy of exhibits next to each other). Because of this, we focus mostly on the path prediction interpretation with aggregated rooms. We still briefly comment on the results calculated on non-modified data, to show the algorithms perform rather poorly in case we do not aggregate it appropriately.

In the figure 4.6, we can see the average subsequence accuracy metric values plotted in the same way as in the results of the rating predictions task. We can observe none of the algorithms clearly dominates all the others in terms of this metric. At depth of 4, deep learning with transitions encoding is the best approach by a little, according to the average metric value, at depth of 8 the Markov chain reaches the highest average subsequence accuracy, and finally, at depth of 12, the deep learning with our proposed encoding with states tokens is the best performing approach by a little difference. Our proposed deep learning modifications with demographic tokens do not seem to perform better than their counterparts without the demographic tokens. The interest transition hybrid model seems to reach a very similar performance as the pure Markov chain approach - it is likely the users' selection of exhibits is not very influenced by how much

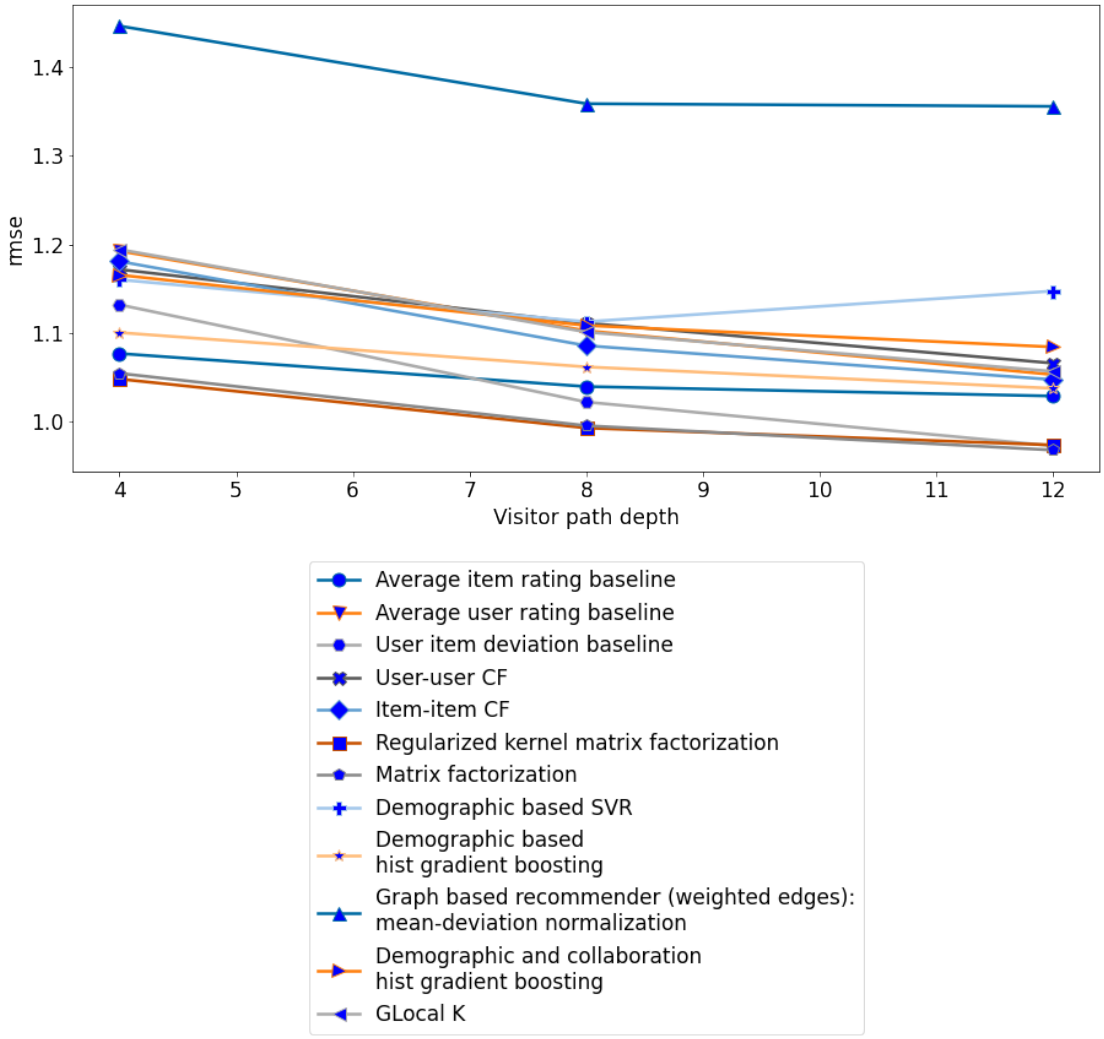

Figure 4.5: Rating prediction evaluation using root mean squared error (RMSE) based on average values from cross-validation at different visitor path depths

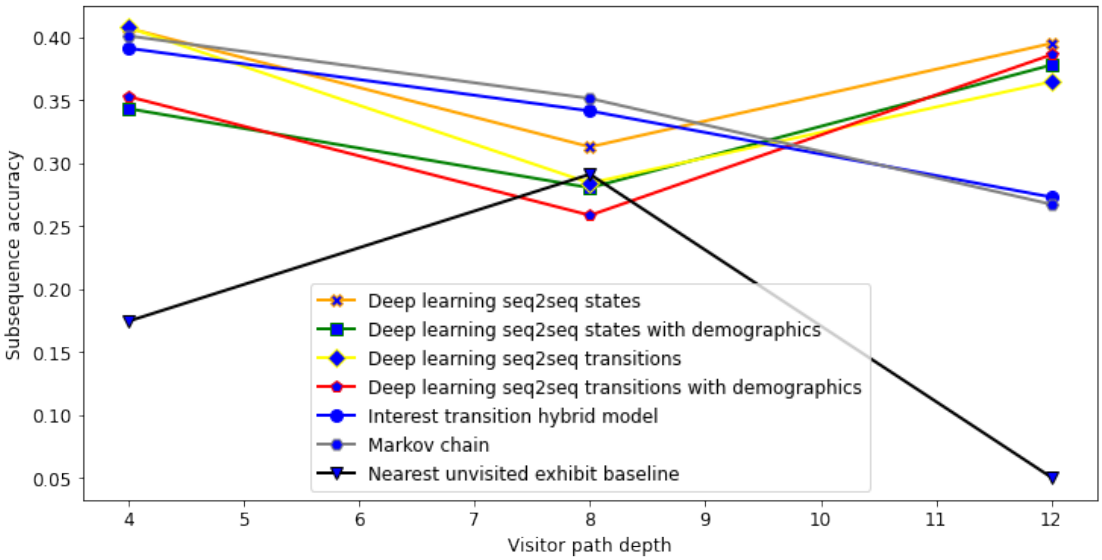

Figure 4.6: Path prediction evaluation using subsequence accuracy based on average values from cross-validation at different visitor path depths with aggregation of small rooms in the exhibition

they will like the exhibit. We can also notice the nearest exhibit baseline approach performs poorly at all depths, outperforming only the demographic approaches at depth of 8 by a little. In the case of the deep learning approaches and the hybrid approach, we would expect to observe a trend of increasing performance with higher visitor path depth, which does not seem to be the case. This might be caused by a bias caused by a different subset of the exhibits being dominant at various depths (as explained in previous chapters illustrated by figure 2.13). For the other approaches, it is no surprise their performance is not increasing with more user history, since they only consider the most recent visited exhibit when making a prediction.

When looking at reported p-values from statistical significance testing for this problem in the figure 4.7, we can see none of the approaches is significantly better than all others. Another observation from the significance testing is all the approaches are significantly better than the nearest exhibit baseline. The deep learning approaches without the demographic tokens are significantly better than their counterparts with the demographics. When comparing the deep learning variants with states encoding and the variant with transition encoding, none of them is significantly better than the other.

In the figure 4.8, which is showing the p-values from significance testing at depth 8 for the subsequence accuracy, we can see that again no approach is significantly better than all the others. However, the Markov chain is superior to all the other approaches with exception of the hybrid model. The model is partially based on the same information as the Markov chain, therefore they are expected to perform relatively similarly. These two models are also the only ones significantly better than the baseline. When comparing

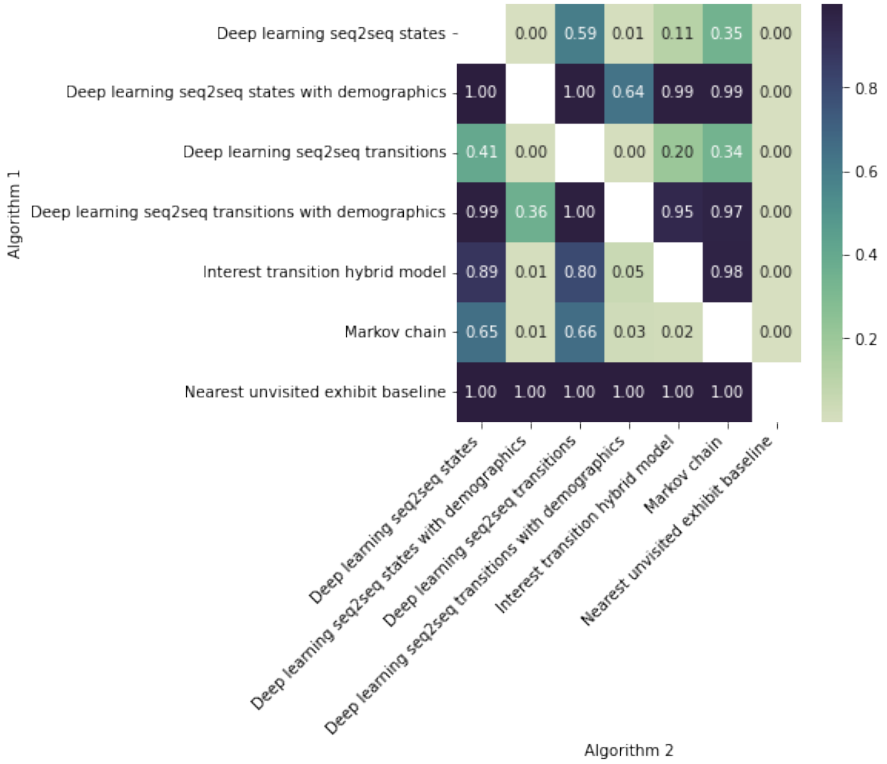

Figure 4.7: P-values from Wilcoxon signed ranked paired tests for the null hypothesis that the mean of the two algorithms is the same and hypothesis 1 that the mean of subsequence accuracy for algorithm 1 is greater than metric of algorithm 2 at visitor path depth of 4

the deep learning approaches, we can see that at this depth there is not a significant difference between the version with and without demographics. However, the version with states encoding is superior to the version with transitions encoding.

At depth of 12 based on the significance testing in the figure 4.9, we can see that all the deep learning approaches are significantly better than the approaches that are not based on deep learning. Among the different variations of deep learning approaches, there is no significant difference at significance level alpha=0.05.

The presence accuracy is a more relaxed version of the subsequence accuracy, therefore it is no surprise that the absolute values of this metric as seen in the figure 4.10 are in general higher compared to subsequence accuracy. Other than that the observed patterns are very similar as in the case of the subsequence accuracy metric. In this case at depth of 4 the Markov chain and deep learning with states encoding exchanged places in the first spot, however according to significance testing there is no significant difference between the two. Again, all approaches are significantly better than the baseline at depth of 4. In the case of the deep learning approaches the versions with transitions encoding are superior

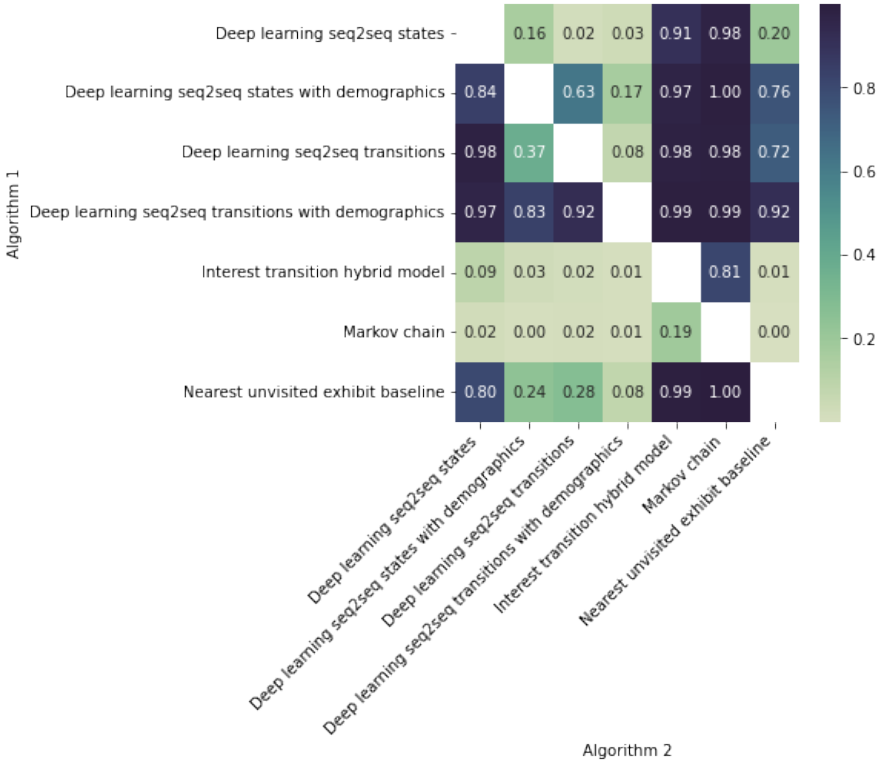

Figure 4.8: P-values from Wilcoxon signed ranked paired tests for the null hypothesis that the mean of the two algorithms is the same and hypothesis 1 that the mean of subsequence accuracy for algorithm 1 is greater than metric of algorithm 2 at visitor path depth of 8

to versions with states encoding. Also again, the versions without the demographics perform significantly better. At depth of 8, the Markov chain reaches the best average metric value however it is not significantly better than all the other algorithms. All the approaches except the demographic ones outperform the baseline significantly at this depth. The versions of deep learning algorithms without the demographics are once again significantly better. Between deep learning with state encoding and transitions encoding there is not a significant difference. At depth of 12, all the deep learning approaches are significantly better than non-deep-learning-based approaches. Among them, there is no significant difference. Also, all the approaches are significantly better than the baseline.

When we look at the subsequence accuracy metric for the data without aggregation of exhibits in small rooms in the figure 4.11, we can notice some differences. Particularly at the depth of 12 where the aggregated rooms are typically visited, we can see all the algorithms (except the baseline) perform worse than in the case of the non-aggregated rooms predictions. As discussed earlier, for the exhibits located next to each other it is very difficult to predict the visitor behavior with reasonable accuracy. The results at

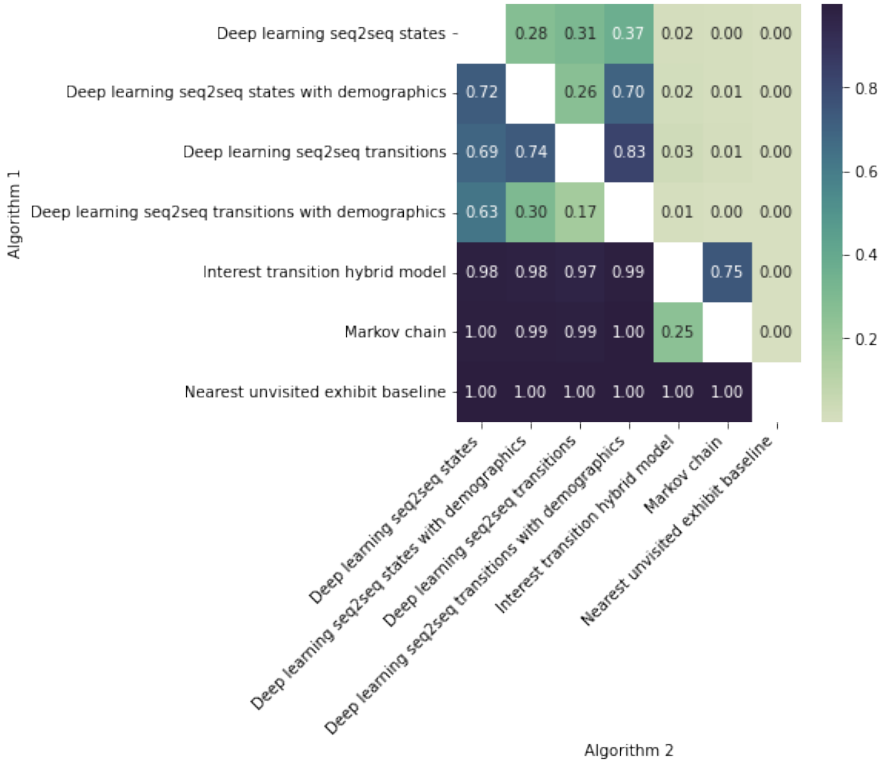

Figure 4.9: P-values from Wilcoxon signed ranked paired tests for the null hypothesis that the mean of the two algorithms is the same and hypothesis 1 that the mean of subsequence accuracy for algorithm 1 is greater than metric of algorithm 2 at visitor path depth of 12

this higher granularity would not be much more useful for the ultimate goal of museum occupancy management. Interestingly, the deep learning approaches are under-performing Markov chain at depths of 8 and 12 in this case, which is the opposite of the situation in the non-aggregated data. It seems the higher granularity complicates the learning process of the deep learning approaches. The deep learning with states encoding without the demographics performs slightly better than other deep learning variants in this case, likely due to the complicated path patterns that are easier to handle for the deep learning algorithm in the case of a simpler encoding.

Similarly, for presence accuracy for data without exhibit aggregations in the figure 4.12, the accuracy of all the algorithms at depth of 12, where the aggregated rooms are usually visited decreased compared to the data with aggregated rooms. Interestingly, in this case, the deep learning with states encoding without demographics manages to predict the complicated sequence decently and reaches a slightly higher average metric value than the Markov chain.

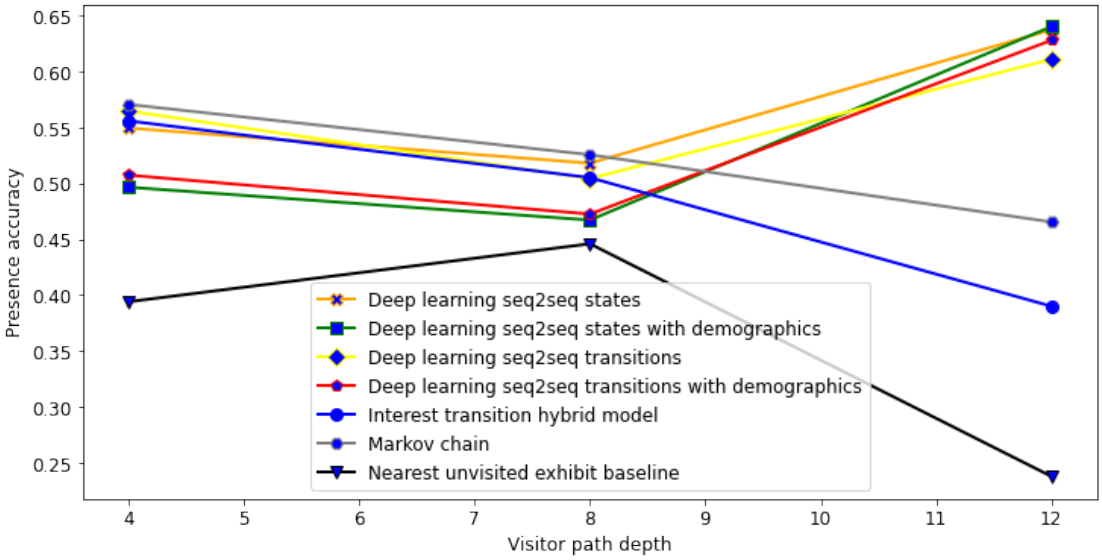

Figure 4.10: Path prediction evaluation using presence accuracy based on average values from cross-validation at different visitor path depths with aggregation of small rooms in the exhibition

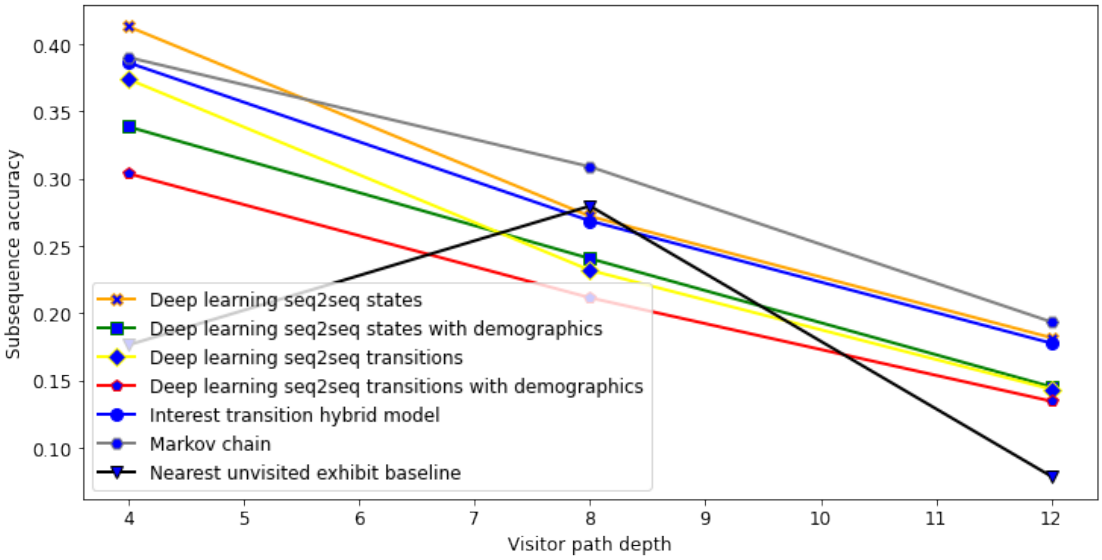

Figure 4.11: Path prediction evaluation using subsequence accuracy based on average values from cross-validation at different visitor path depths without small room aggregation

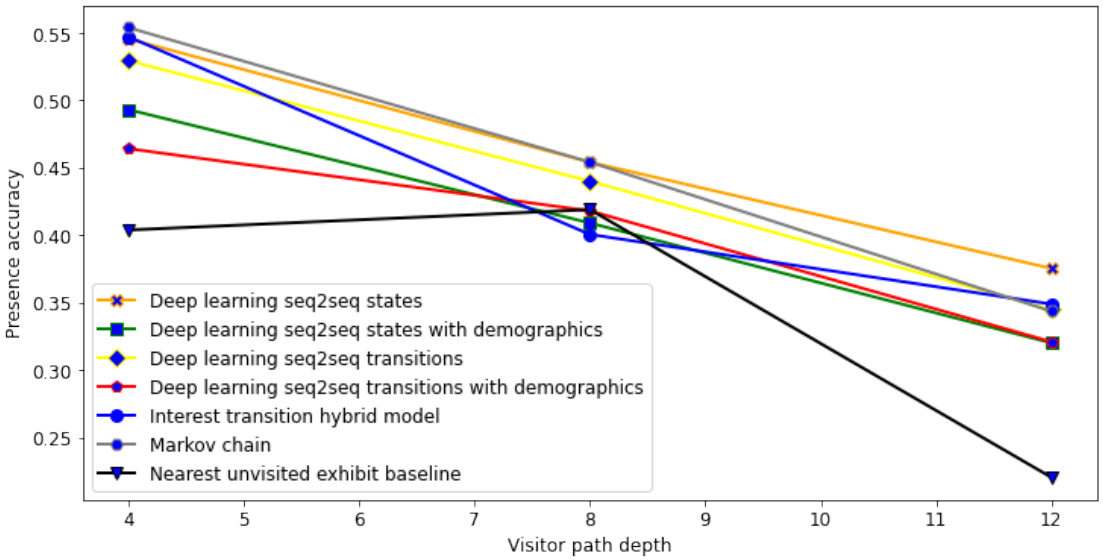

Figure 4.12: Path prediction evaluation using presence accuracy based on average values from cross-validation at different visitor path depths without small room aggregation

4.3 Algorithm sensitivity to the number of users in the dataset

To better understand the performance of the algorithms, we also analyze how well the algorithms perform with different amounts of data. This point of view is useful for choosing an appropriate recommender for an even smaller-sized exhibition/event with fewer visitors and/or using a different algorithm at an early stage of the exhibition when we still have little data. For this purpose, we have applied the same evaluation setting to the dataset with one-third of randomly sampled users, two-thirds of randomly sampled users and compared it together with the performance calculated on the full data. Since we add yet another dimension to the results, we could analyze the performance for different amounts of users for each metric. This would of course result in too many plots. Instead, we only focus on the most relevant metrics - precision at k=3 and k=6 for rating prediction and subsequence accuracy for path prediction.

4.3.1 Rating prediction

In the figure 4.13, we can see precision at k=3 for an under-sampled dataset containing one-third of the users. When we compare these results with the corresponding metric computed on the full dataset as shown in the figure 4.1, we can observe that all the metric values are worse for all of the algorithms what is expected with less data. Interestingly, the GLocal-K approach is not dominating the metric as in the case of the full dataset at depths 8 and 12. This is expected since the approach was designed for a lot larger dataset. Other than that, we can notice the performance of the algorithms seems to be more-less

stable, thus an algorithm performing well at a particular depth is not necessarily among the best-performing ones at another depth.

With two-thirds of users for the metric precision at k=3 in the figure 4.13, we can notice the metric values are in general still slightly smaller in most cases which were expected with less data. When compared to the figure with only one-third of users, we can see the GLocal-K has already sufficient amounts of data to perform decently in contrast with only one-third of the users. With this amount of data at depths of 8 and 12, it already predicts similarly as in the case of the full dataset.

For rating prediction metric precision at k=6 with one-third of users in the figure 4.14, we can see the metric values are surprisingly slightly higher at depths of 8 and 12 when compared to the full dataset metrics in the figure 4.1. This is most likely caused by a random influence of the particular user sample (due to the computational complexity of the evaluation framework it was not possible to test multiple user sub-samples). Similarly, as with full data, no approach is clearly dominant based on this metric, however, we can notice the regularized kernel matrix factorization is still among the best performing at depths of 4 and 8 and GLocal-K leads by a small difference at depth of 12.

For rating prediction metric precision at k=6 with two-thirds of users in the figure 4.14, we can see the metric values are also unexpectedly slightly higher at depths of 8 and 12 when compared to the full dataset metrics in the figure 4.1. The other patterns in the results are very similar, as observed in the results for one-third of the dataset - no approach is clearly dominant while similar approaches are near among the best-performing ones as in the case of full data.

4.3.2 Path prediction

For the path prediction task with the under-sampled dataset, we focus only on the subsequence accuracy metric calculated on the data with aggregated rooms. In the figure 4.15, we can see the results for one-third of users, where there are smaller differences between the deep learning algorithms at depth of 12 with less data when compared with full dataset results in the figure 4.6. At depth of 8, we can again see the deep learning approaches are performing slightly worse so the difference between the first Markov chain-based models and the deep learning approaches is smaller. At depth of 4, the smaller data size does not seem to have changed much, with exception of approaches that use demographics falling behind even more. Surprisingly when we look at the metric values in general we can see they are slightly higher compared to the full dataset, this is most likely just a random influence of the particular user sub-sample. Testing different sub-samples was not feasible due to the long runtimes of the overall evaluation framework.

In the figure 4.16, which is showing the subsequence accuracy results for the sub-sampled dataset with two-thirds of users, we can see similar patterns as in the case of the one-third of users sub-sample. Again, we can notice the deep learning approaches are not outperforming other approaches as much when compared with full dataset results in the figure 4.6. When comparing these results, with the results for one-third of users in the

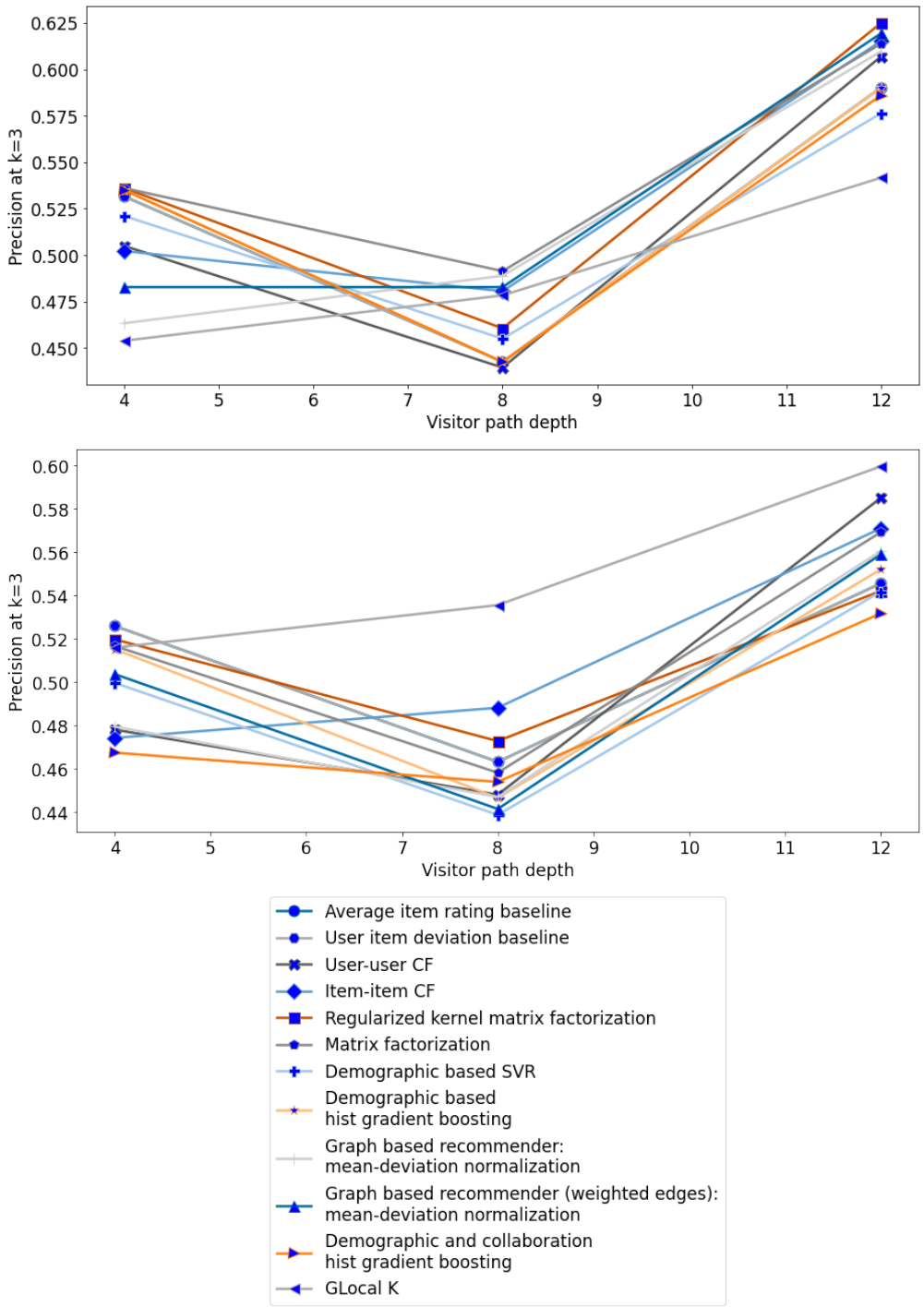

Figure 4.13: Rating prediction with precision at k=3 metric calculated on a under-sampled dataset with one-third of users (57 users) in the upper line-chart and with two-thirds of users (113 users) in the lower line-chart

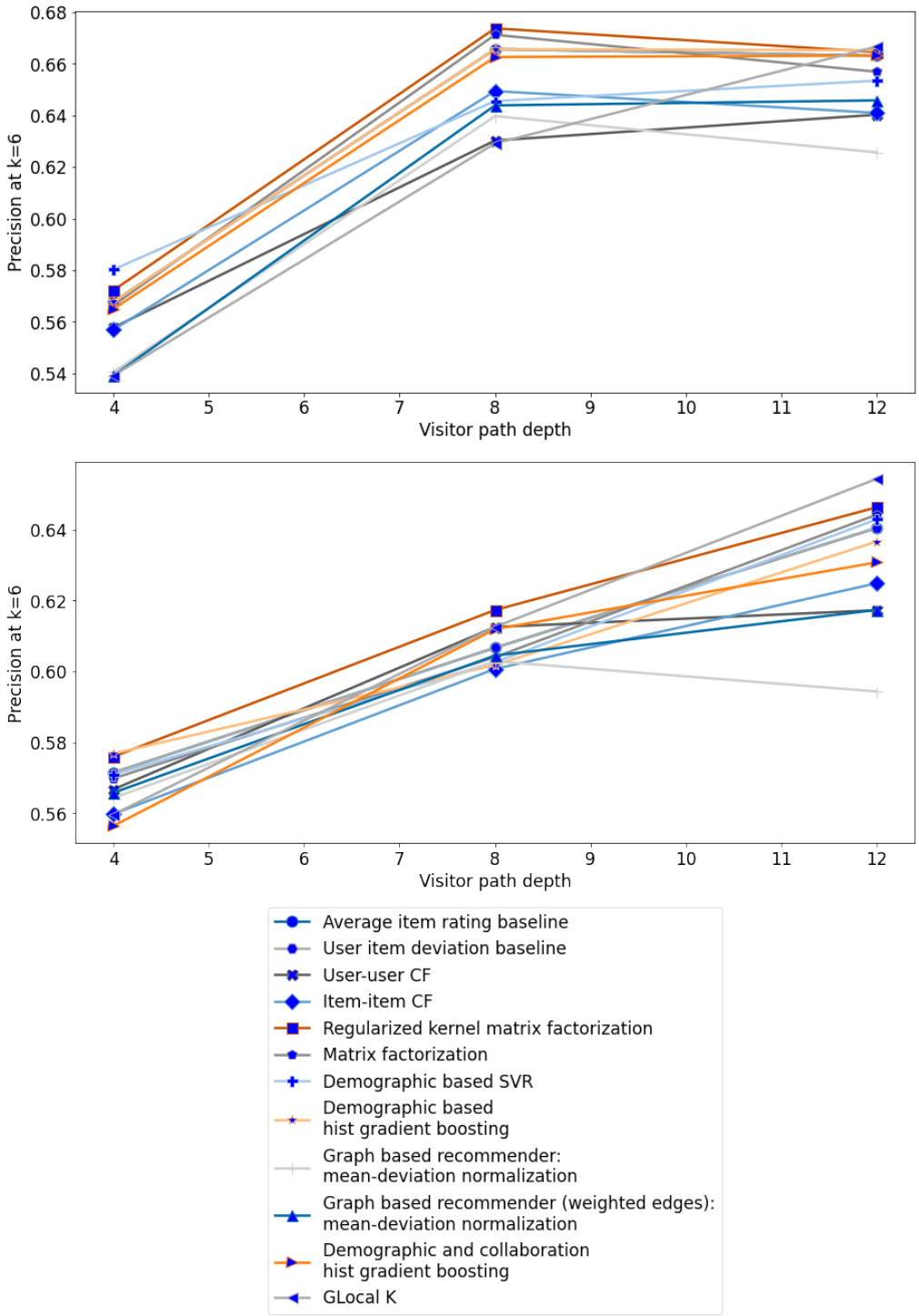

Figure 4.14: Rating prediction with precision at k=6 metric calculated on a under-sampled dataset with one-third of users (57 users) in the upper figure and with two-thirds of users (113 users) in the lower figure

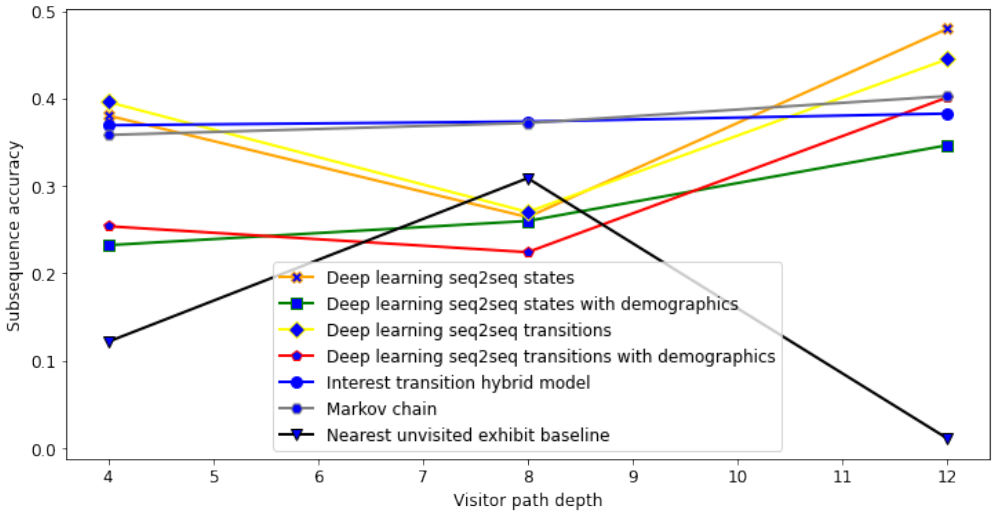

Figure 4.15: Path prediction with subsequence accuracy metric calculated on a under-sampled dataset with with one-third of users (57 users) with aggregated small rooms

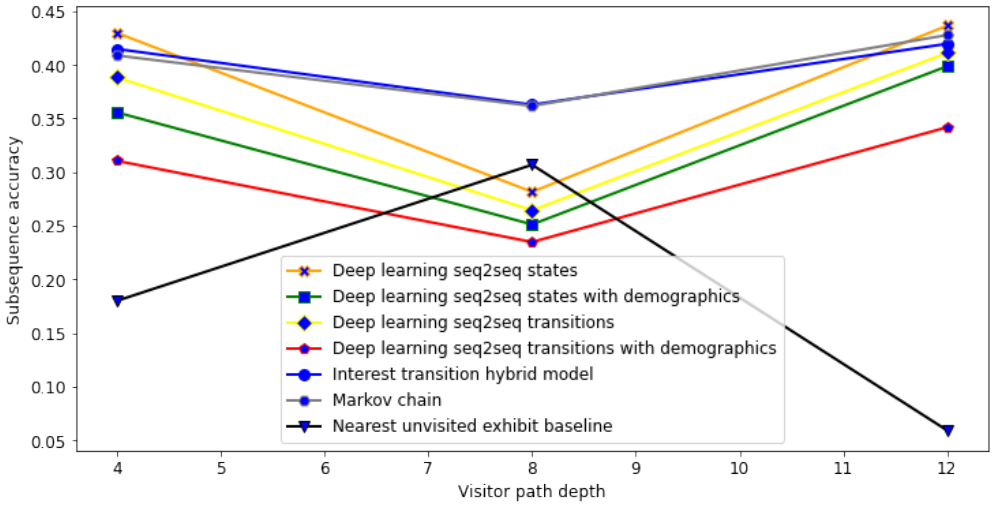

Figure 4.16: Path prediction with subsequence accuracy metric calculated on a under-sampled dataset with with two-thirds of users (113 users) with aggregated small rooms

figure 4.15, we can see the larger amount of data, in this case, helped the deep learning with demographics approaches which are now less behind the other approaches. Also, similarly, as in the case, of the one-third sub-sampled dataset, we can notice the metric values of the algorithms are again a bit higher in general which is most likely caused by the particular random user sub-sample which was easier to predict, therefore all the approaches reached higher metric values.

CHAPTER 5

Summary & Conclusions

In this thesis, we have explored and evaluated recommender system approaches for rating prediction and path prediction problems in a museum/exhibition domain. We propose an evaluation framework, where we respect the order of the exhibits in which they are visited. We argue this is necessary for a physical setting because the selection of the first visited items/exhibits is not random but is influenced by the physical museum layout. The analysis was based on real-life data collected in an exhibition in Rome using the Kouzan web application. Unfortunately, due to the experiment being financed by a private company, the data is proprietary and can't be shared at the time of publishing of this thesis. We hope to make it available to the scientific community in the near future. Until then, the data can be requested from the data owner Vittorio Loreto at Sony Computer Science Labs Paris vittorio.loreto@sony.com.

As the previous research on this topic is very limited, we have adapted approaches proposed for similar settings, selected several commonly used general approaches which were applicable, proposed several modifications of the approaches, and evaluated and compared all the rating and path prediction methods with the most relevant metrics for the domain. Additionally, we have analyzed the approaches for both problems at different visitor path depths which was not done before (the number of exhibits the test user already rated/visited). We have also calculated and briefly discussed the results with a smaller, sub-sampled dataset, to give an idea what is the expected performance at even smaller exhibitions or in the early phases of the exhibition with fewer visitors. The main findings from the results in the context of the initial research questions for the two problems are discussed in the following subsections. An outlook on possible uses of the recommenders, the limitations of this study, and possible future work is also discussed in this chapter.

5.1 Rating prediction

The research question for this problem was stated as: "What are appropriate methods to accurately predict user ratings in an exhibition or museum setting?". The most general answer to this question would be the GLocal-K algorithm, which is the state-of-the-art approach for the popular dataset from a different domain - MovieLens 1M. In the context of ranking metric precision at k=3, this approach was the best performing one, especially with sufficient user rating history (visitor path depths of 8 and 12). For the visitor path depth of 8, the difference was statistically significant when compared to all the other algorithms, while at depth 12 it was not statistically significant when compared to all the other algorithms. At visitor path depth of 4 was the algorithm performing very similarly to the others and as expected based on the small differences, no approach was performing significantly better than the baseline at this depth.

For the metric precision at k=6, where the number of items we wish to recommend to a user is higher, is the performance of all approaches more balanced. The GLocal-K is still among the best performing approaches (and at depth of 12 it is the best performing). However, due to the small differences, none of the approaches is significantly better than all the others. Based only on the average metric value, the regularized kernel matrix factorization is the best approach at depths of 4 and 8.

If we aim to predict even more items, as in the case of metric precision at k=9, the differences in average performance among the algorithms get even smaller. At depths of 4 and 8, the best performing approach is the regularized kernel matrix factorization according to the average metric value, at depth of 12 it is the user-user CF. As expected, the differences were not significant, no approach was outperforming all the other approaches. In the case of a depth of 8, it is almost the opposite case with regularized kernel matrix factorization as it is significantly better when compared to all approaches including baselines with the only exception of demographics-based SVR.

Due to the limited previous research on the topic, we have considered the graph-based recommender as the state-of-the-art approach for this problem, even though the setting was not exactly the same. The other pieces of information about the exhibits which have the authors used in the original work turned out to be important as in our simplified case the approach was not among the best-performing ones.

Unfortunately, none of our proposed modifications for the rating prediction task turned out to be successful. Demographic-based recommender with histogram gradient boosting was not significantly outperforming its original counterpart with SVR. In this case, we hoped a popular powerful machine-learning ensemble-based algorithm could perform well which was unfortunately not the case. The likely reason for all the demographic approaches performing poorly might be the homogeneity in the visitors' demographics. In case of the graph-based recommender with weighted edges was not significantly better than the original without the weights. In the case of the proposed approach combining the demographics and collaboration data, the proposed approach with the additional feature did not perform better either.

5.2 Path prediction

The research question for this problem was stated as: "What are appropriate methods to accurately predict the next visited installation at every point of a user's path?". Based on the subsequence accuracy metric which is our main focus is the answer to this question less straightforward. We observe that at different points in the exhibition can a different approach be the best performing one without a clear connection to the visitor path depth. According to the average metric value, at depth of 4, deep learning with transitions encoding is the best approach, at depth of 8 it is the Markov chain and at depth of 12 it is again the deep learning, but with states encoding instead. At depth of 4, no approach is significantly better than all the others, at depth of 8 is the Markov chain significantly better than all the others with exception of the hybrid model which is not surprising since they are highly related. At depth of 12, all of the deep learning approaches significantly over-perform all the other approaches.

Regarding our proposed modifications of the path prediction approaches, unfortunately in case of the demographic version of deep learning is in most cases significantly worse or at most equally well-performing than the original version. In the case of the two encoding variants of the deep learning algorithm, they are either equally well-performing or the states encoding is significantly better. Therefore, we can consider this proposed modification as a slightly better option.

In conclusion, we can say that either the Markov chain or the deep learning approach without demographics with state encoding are the most appropriate methods while the deep learning approach is expected to make the best predictions more often based on our results. It is important to note we can't expect perfect accuracy for the path prediction task based on such data. There might be other aspects affecting the visitors such as visitors visiting as a part of a group or an organized tour with a guide making them follow a different route than they would choose.

5.3 Possible applications of the recommenders

The rating prediction outputs can be used to select a subset of exhibits out of the unseen exhibits, that the visitor will like the most. Based on our results, we can see the predictions are more accurate with longer user history and with fewer items being predicted. Therefore, it is more suitable for planning the rest of the visitor path later in the exhibition when we already have enough ratings from the user. We can then take the list of the recommended items and plan the shortest path that visits all of the items based on the physical distances. This task is known as the traveling salesman problem [San18]. Another feature that can be based on the rating predictions could predict a few most likable items for the user nearby her/his current location which can be determined by the last item she/he has rated. This feature could also take into account the predicted occupancy of the exhibits and recommend only the ones where we do not expect too many people. Therefore we can recommend subsets of exhibits for the users who do not

wish to visit all the exhibits. However, it is better if the user rates a higher number of exhibits in order for the predictions to be more accurate.

The path prediction outputs can be used to predict the next visited exhibits for a visitor and therefore predict exhibit occupancy. More specifically, at each point in time, we could predict the next visited exhibits for all of the users and calculate the number of occurrences of each exhibit at each time point within the predictions (with the simplification that we assume users spend roughly the same amount of time with each exhibit). For the exhibits where the number of predicted visitors at some point in time would be higher than a set capacity threshold, we could redirect the users. It is desirable to propose alternative exhibits the visitor would like the most, therefore this could be based on the predicted ratings. This results in an optimization problem where we aim to assign the exhibits to users at the next time steps in a way that minimizes the overcrowding and maximizes the predicted likability of the recommended items. The additional constraint could consist of selecting the smallest subset of users who are redirected (because the redirected users might be less happy with their experience). Also, another constraint should take care of the distances within the recommended alternative items so that they are distanced reasonably from each other and the user's last location and they are not spread across the exhibition. Future work could analyze various possible implementations of the visitor redirection mechanism. It is needed to test the algorithms in a real-file setting to see how well it is possible to achieve the initial goal of minimizing the overcrowding. The reason is it is unknown how likely are the visitors to follow the offered suggestions.

5.4 Limitations & Future work

One of the arguable limitations of this work is the used dataset. To our knowledge, the used dataset is the only available dataset for the use case. Therefore, there might arguably be certain biases given by factors such as the topic of the exhibition, the visitor demographics which are dominated by people with a scientific background, and the size of the exhibition. These factors might limit the generalizability of the conclusions made to certain other exhibitions. Future work could explore how consistent are the results at other exhibitions.

The list of approaches in particular for the rating prediction approach is not exhaustive. This work focused on the most relevant and common approaches, however, there are still several approaches we have not tested, that could possibly work well in this setting. For example, for the rating prediction task, future work could focus on the state-of-the-art solutions for other popular datasets such as the Netflix dataset, since using an approach for datasets from other domains with different properties surprisingly turned out to be successful.

In our work we employed parameter tuning of the algorithms, however, due to many approaches being examined, complex evaluation framework, and limited computational resources could the hyper-parameters be explored in greater depth. Another possibility of further improving the performance of the already tested approaches would be to alternate

the architecture of the algorithms where applicable - for example modifying the neural network structure in case of the deep learning path prediction.

Since in the path prediction use case we work with the obvious simplification, that the visitors spend roughly the same amount of time with different exhibits, future studies could analyze whether estimating the time spent with the exhibits is possible, based on the dataset containing only explicit feedback as ours. Further research could also focus on the meta-analysis of the recommenders when applied in the real-life setting - for example, evaluate user satisfaction with the recommendations.

Since museum visitors commonly visit in groups, future work could also focus on the feasibility of a group recommender system (GRS) for a museum. It was previously shown in the related tourism domain that this is a non-trivial task as the characteristics of the groups influence their preferences and there is no single optimal method how to aggregate the individual recommendations [DNW18].

Another possible extension of this work could be if we use a similar application in a conference setting. There, in addition to the item recommendations and path predictions, we could also recommend people with similar interests to talk to, which could then be followed by another meta-analysis of user satisfaction with the recommender.

List of Figures

Bibliography

[AAB+15] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[App] Kouzan expo in rome. `https://www.csh.ac.at/kouzan-app-new-experience-in-rome-museum/`. Accessed: 2021-10-30.

[ben18] bentrevett. pytorch-seq2seq. `https://github.com/bentrevett/pytorch-seq2seq/blob/master/1%20-%20Sequence%20to%20Sequence%20Learning%20with%20Neural%20Networks.ipynb`, 2018.

[BLBDB17] Federico Bartoli, Giuseppe Lisanti, Lamberto Ballan, and Alberto Del Bimbo. Context-aware trajectory prediction. 05 2017.

[BZB+08] Fabian Bohnert, Ingrid Zukerman, Shlomo Berkovsky, Timothy Baldwin, and Liz Sonenberg. Using interest and transition models to predict visitor locations in museums. *AI Communications*, 21:195–202, 04 2008.

[BZL12] Fabian Bohnert, Ingrid Zukerman, and Junaidy Laures. Geckommender: Personalised theme and tour recommendations for museums. volume 7379, 07 2012.

[CCCO21] P. Centorrino, A. Corbetta, E. Cristiani, and E. Onofri. Managing crowded museums: Visitors flow measurement, analysis, modeling, and optimization. *Journal of Computational Science*, 53:101357, 2021.

[CDC14] Pedro G. Campos, Fernando Díez, and Iván Cantador. Time-aware recommender systems: A comprehensive survey and analysis of existing evaluation protocols. *User Modeling and User-Adapted Interaction*, 24(1–2):67–119, feb 2014.

[CLM12] Ka Chan, C. Lenard, and Terence Mills. An introduction to markov chains. 12 2012.

[Dem06] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(1):1–30, 2006.

[Die98] Thomas G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923, 1998.

[DNW18] Amra Delic, Julia Neidhardt, and Hannes Werthner. Group decision making and group recommendations. In *2018 IEEE 20th Conference on Business Informatics (CBI)*, volume 01, pages 79–88, 2018.

[EHR+20] Andrew Emerson, Nathan Henderson, Jonathan Rowe, Wookhee Min, Seung Lee, James Minogue, and James Lester. *Early Prediction of Visitor Engagement in Science Museums with Multimodal Learning Analytics*, page 107–116. Association for Computing Machinery, New York, NY, USA, 2020.

[Exh] Uncertainty. interpreting the present, foreseeing the future. `https://www.palazzoesposizioni.it/mostra/incertezza-interpretare-il-presente-prevedere-il-futuro-eng`. Accessed: 2021-10-30.

[HLL+21] Soyeon Caren Han, Taejun Lim, Siqu Long, Bernd Burgstaller, and Josiah Poon. Glocal-k: Global and local kernels for recommender systems. *CoRR*, abs/2108.12184, 2021.

[HMvdW+20] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[HSSC] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx.

[IMSA22] Md. Ashraful Islam, Mir Mahathir Mohammad, Sarkar Snigdha Sarathi Das, and Mohammed Eunus Ali. A survey on deep learning based point-of-interest (poi) recommendations. *Neurocomputing*, 472:306–325, 2022.

[KBV09] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[KBZ12] Tsvi Kuflik, Zvi Boger, and Massimo Zancanaro. Analysis and prediction of museum visitors' behavioral pattern types. *Cognitive Technologies*, 04 2012.

[KCL21] Jaekyeong Kim, Ilyoung Choi, and Qinglong Li. Customer satisfaction of recommender system: Examining accuracy and diversity in several types of recommendation approaches. *Sustainability*, 13(11):6165, May 2021.

[LdGS11] Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. *Content-based Recommender Systems: State of the Art and Trends*, pages 73–105. 01 2011.

[LGHZ16] Huayu Li, Yong Ge, Richang Hong, and Hengshu Zhu. Point-of-interest recommendations: Learning potential check-ins from friends. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 975–984, New York, NY, USA, 2016. Association for Computing Machinery.

[LH14] Dokyun Lee and Kartik Hosanagar. Impact of recommender systems on sales volume and diversity. In *ICIS*, 2014.

[MAGM20] Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. From goals, waypoints & paths to long term human trajectory forecasting. *CoRR*, abs/2012.01526, 2020.

[MKK17] Einat Minkov, Keren Kahanov, and Tsvi Kuflik. Graph-based recommendation integrating rating history and domain knowledge: Application to on-site guidance of museum visitors. *Journal of the Association for Information Science and Technology*, 68, 06 2017.

[Mov] Movielens 1m dataset. `https://grouplens.org/datasets/movielens/1m/`. Accessed: 2021-10-30.

[pdt20] The pandas development team. pandas-dev/pandas: Pandas, February 2020.

[PGC+20] Francesco Piccialli, Fabio Giampaolo, Giampaolo Casolla, Vincenzo Schiano Di Cola, and Kenli Li. A deep learning approach for path prediction in a location-based iot system. *Pervasive and Mobile Computing*, 66:101210, 2020.

[PS15] S GOPAL PATRO and Dr-Kishore Kumar Sahu. Normalization: A preprocessing stage. *IARJSET*, 03 2015.

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[RHS+13] Tuukka Ruotsalo, Krister Haav, Antony Stoyanov, Sylvain Roche, Elena Fani, Romina Deliai, Eetu Mäkelä, Tomi Kauppinen, and Eero Hyvönen. Smartmuseum: A mobile recommender system for the web of data. *Journal of Web Semantics*, 20:50–67, 2013.

[RRS10] Francesco Ricci, Lior Rokach, and Bracha Shapira. *Recommender Systems Handbook*, volume 1-35, pages 1–35. 10 2010.

[RST08] Steffen Rendle and Lars Schmidt-Thieme. Online-updating regularized kernel matrix factorization models for large-scale recommender systems. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, page 251–258, New York, NY, USA, 2008. Association for Computing Machinery.

[Sac20] Dimitris Sacharidis. Lecture notes in recommender systems, April 2020.

[San18] Shabnam Sangwan. Literature review on travelling salesman problem. *International Journal of Research*, 5:1152, 06 2018.

[VGO+20] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[WCN12] Yuanyuan Wang, Stephen Chi-Fai Chan, and Grace Ngai. Applicability of demographic recommender system to tourist attractions: A case study on trip advisor. In *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, volume 3, pages 97–101, 2012.

[WH00] Rüdiger Wirth and Jochen Hipp. Crisp-dm: towards a standard process modell for data mining. 2000.

[WHW+19] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z. Sheng, and Mehmet Orgun. Sequential recommender systems: Challenges, progress and prospects. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 6332–6338. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[WWN+09] Yiwen Wang, y Wang@tue, Lora Nl, Lora Aroyo, n Stash@tue, Rody Nl, Yuri Sambeek, Guus Schuurmans, and Guus Schreiber. Cultivating personalized museum tours online and on-site. *Interdisciplinary Science Reviews - INTERDISCIPLIN SCI REV*, 34, 04 2009.

[YCS14] Quan Yuan, Gao Cong, and Aixin Sun. Graph-based point-of-interest recommendation with geographical and temporal influences. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, page 659–668, New York, NY, USA, 2014. Association for Computing Machinery.

[ZZLW21] Daniel Zhang, Yang Zhang, Qi Li, and Dong Wang. Sparse user check-in venue prediction by exploring latent decision contexts from location-based social networks. *IEEE Transactions on Big Data*, 7(5):859–872, 2021.