DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF SOCIAL SCIENCES
DEPARTMENT OF BUSINESS ADMINISTRATION
BUSINESS INFORMATION SYSTEMS PROGRAM
MASTER'S THESIS

A LARGE SCALE RECOMMENDER SYSTEM UTILIZING SOCIAL DATA

Ayhan Fuat ÇELİK

Supervisor
Assist. Prof. Dr. Güzin ÖZDAĞOĞLU

İZMİR-2014

MASTER THESIS/PROJECT
APPROVAL PAGE

University	: Dokuz Eylül University
Graduate School	: Graduate School of Social Sciences
Name and Surname	: Ayhan Fuat ÇELİK
Title of Thesis	: A Large Scale Recommender System Utilizing Social Data
Defence Date	: 05.08.2014
Supervisor	: Assist Prof.Dr.Güzin ÖZDAĞOĞLU

EXAMINING COMMITTE MEMBERS

Title, Name and Surname	University	Signature
Assist Prof.Dr.Güzin ÖZDAĞOĞLU	DOKUZ EYLUL UNIVERSITY	.....
Assoc Prof.Dr.Sabri ERDEM	DOKUZ EYLUL UNIVERSITY	.....
Assoc.Prof.Dr. .Ali ÖZDEMİR	DOKUZ EYLUL UNIVERSITY	.....

Unanimity (✓)
Majority of votes ( )

The thesis titled as "**A Large Scale Recommender System Utilizing Social Data**" prepared and presented by Ayhan Fuat ÇELİKis accepted and approved.

Prof.Dr. Utku UTKULU
Director

DECLARATION

I hereby declare that this master's thesis titled as **"A Large Scale Recommender System Utilizing Social Data"** has been written by myself without applying the help that can be contrary to academic rules and ethical conduct. I also declare that all materials benefited in this thesis consist of the mentioned resources in the reference list. I verify all these with my honor.

…/…/…….

Ayhan Fuat ÇELİK

ABSTRACT

Master's Thesis

A Large Scale Recommender System Utilizing Social Data

Ayhan Fuat ÇELİK

Dokuz Eylül University

Graduate School of Social Sciences

Department of Business Administration

Business Information Systems Program

For today's online services, offering the right content to their users has become the primary purpose. Competing on the web, where the alternatives are limitless, requires understanding users' profiles, similarities, and differences. Recommender systems emerged from this need to provide automated help in making every day decisions.

In this study, we evaluated the recommender systems in a location based social network setting. In these platforms people share the places they visited, rate these places or leave tips about them.

In order to build a large scale location recommender system, we collected around 6.7 million check-ins made between March 2014 and June 2014. The data set contains 530 thousand users and 580 thousand venues from Turkey.

We provide a thorough analysis of which types of users visited which types of places and when. We evaluated the performances of user-based and item-based collaborative filtering techniques on the check-in data. We also considered modified versions of these techniques integrating trust and location information, respectively. Finally, we compared these models by their accuracy, computational complexity, and the real value they have in these services.

Keywords: Recommender Systems, Location Based Social Networks, Data Mining, Machine Learning.

ÖZET

Yüksek Lisans Tezi

Sosyal Veri Kullanan Büyük Ölçekli Bir Tavsiye Sistemi

Ayhan Fuat ÇELİK

Dokuz Eylül Üniversitesi

Sosyal Bilimler Enstitüsü

İngilizce İşletme Anabilim Dalı

İngilizce İşletme Bilişim Sistemleri Programı

Günümüzde, kullanıcılara doğru içeriği sunmak çevrimiçi servislerin ana amacı haline gelmiştir. Web'de rekabet edebilmek için kullanıcı profillerini, benzerliklerini ve farklılıklarını anlamak gerekmektedir. Tavsiye sistemleri, bu ihtiyaç üzerine, gündelik kararlara otomatize bir şekilde yardım sağlamak için ortaya çıkmıştır.

Bu çalışmada tavsiye sistemleri, konum tabanlı sosyal ağlar üzerinden değerlendirilmiştir. Kullanıcılar bu platformlarda ziyaret ettikleri yerleri, bu yerlere verdikleri oyları ve bu yerler hakkındaki görüşlerini paylaşabilmektedir.

Büyük ölçekli bir konum tavsiye sistemi oluşturmak amacıyla Mart 2014 – Haziran 2014 tarihleri arasında gerçekleşen 6.7 milyon ziyaret verisi toplanmıştır. Veri seti, Türkiye'den 530 bin kullanıcı ve 580 bin mekan içermektedir.

Çalışmada hangi tür kullanıcıların hangi tür mekanları ne zaman ziyaret ettiği detaylı olarak incelenmiştir. Kullanıcı ve öğe bazlı işbirlikçi filtreleme teknikleri ziyaret verisi üzerinden değerlendirilmiştir. Sonuç olarak, modeller doğruluk, işlem maliyeti ve gerçek değerleri üzerinden kıyaslanmıştır.

Anahtar Kelimeler: Tavsiye Sistemleri, Konum Tabanlı Sosyal Ağlar, Makine Öğrenmesi, Veri Madenciliği.

A LARGE SCALE RECOMMENDER SYSTEM UTILIZING SOCIAL DATA

CONTENTS

CHAPTER TWO
RECOMMENDATION IN LOCATION BASED SOCIAL NETWORKS

CHAPTER THREE
BUILDING THE RECOMMENDER SYSTEM

ABBREVIATION

API	Application Programming Interface
JSON	JavaScript Object Notation
LBSN	Location Based Social Networks
POI	Point of Interest
XML	Extensible Markup Language

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

INTRODUCTION

Internet has transformed our world into a world of abundance. Not very long time ago, we were watching the same TV shows, listening to the same artists, and reading the same best-sellers as they were the only choices we had. Now, we have endless possibilities at our fingertips: Google answers us in a fraction of a second, searching billions of web pages; Amazon has millions of books in its shelves; iTunes' playlist contains millions of songs; eBay has millions of listings at a given time… Out of these limitless alternatives, only a small portion gets to become the *hits*, leaving others to vanish into the thin air.

In his article, "*The Long Tail"*, Anderson (2004) argues that "the future of entertainment is in the millions of niche markets at the shallow end of the bitstream". To this end, he proposes: "Make everything available" – "Help me find it".

From their creation in early 90's to their strong integration to Web 2.0 technologies today, recommender systems have supported Anderson's theory. Amazon's conversion to sales of on-site recommendations is estimated to be as high as 60% (Mangalindan, 2012: 1). 75% of what people watch on Netflix comes from recommendations (Amatriain and Basicilo, 2012: 1). After the deployment of a new recommender system, click through rate in Google News improved by 31% (Liu et al., 2009: 38).

Recommender systems' wide application domain is not limited to e-commerce. While they are vastly used to recommend what to buy, they are also used to increase user engagement. YouTube offers similar videos in order to increase the time spent on the website. Facebook and Twitter recommend users to follow or to become friends with. Google recommends web pages based on a search term. eHarmony recommends people to date with.

In our study, the application domain is locations. Location recommendation problem dates back to the times recommender systems didn't exist. In the absence of a recommender system the alternative is usually a friend: a person who knows you, who understands your expectations and who is aware of your constraints. On a non-personalized level, expert opinions are also used as recommendations. Tour guides and travel guides can help you decide where to go.

Recommender systems have an edge on personalization and dealing with information overload over friends and experts. We, knowingly or unknowingly, feed these systems with a great deal of information. Our profiles, tastes, connections, and reactions are readily available to become an input to these systems.

Location-based social networks (e.g. Foursquare and Gowalla) mainly serve as platforms where users share the places they visit with their friends. However, with the information users provide to these platforms, they also recommend places to visit. In Foursquare's case, the recommendations have become such an integral part of the platform that many users have started to use it mainly for this purpose. In fact, Foursquare recently announced that the platform will serve solely as a recommender system and the social network will move to a new application called Swarm.

While location recommendation attracts many users' attention, research on the subject is limited as the location based social networks only date back to 2009. In this study, we aim at providing insights on people's behavior in location based social networks and how they affect recommendations. Understanding user behavior, figuring out the patterns and spotting the irregularities is a key in recommender systems' success. To this aim, we have built several models from the literature to see their strengths and weaknesses in location recommendation setting.

Our dataset consists of 6.7 million visits occurred in a time period of March 2014 to June 2014. The dataset contains 530 thousand users and 580 thousand venues from Turkey. To the best of our knowledge, this is the first study conducted on Turkish users' mobility behavior on location based social networks.

The scope of our study is collaborative filtering techniques as they mostly focus on finding similarities. We infer that the principal techniques are a good starting point in a local environment.

The thesis consists of three main chapters and is constructed as follows:

In the first chapter, we present the formal definition recommendation problem along with its components. Then, with a focus on collaborative filtering techniques, we explain the state-of-the-art techniques used in recommender systems. We finish the chapter with the evaluation metrics used in recommender systems.

In the second chapter, we introduce location based social networks and give a brief literature review on user behavior and recommender systems in these networks.

In the last chapter of the study, we explain our data collection method and provide descriptive analysis on category, check-in frequency, gender, time and location dimensions of the data. Then, we present our design choices on the model building phase and build user-based and item-based models. We also present two modifications of these models integrated with friends and locations. We finalize the chapter with a comparative evaluation on the performances of the models.

In conclusion, we discuss the importance of results along with the insights we gained from this study. Then, we present the limitations and possible future directions of the study.

CHAPTER ONE
RECOMMENDER SYSTEMS

This chapter defines the definition of recommendation problem considering the areas this problem arises and the components of the problem and follows by the state-of-the-art recommendation techniques from the literature.

1.1. RECOMMENDATION PROBLEM

In its simplest form, recommendation is an aid to solve a decision problem. Therefore, it deals with directing the user to the best set of alternatives based on some criteria. In this section we first define the recommendation problem and then we analyze the "user", "item" and "utility" components of the problem.

1.1.1. Formal Definition

Ricci et al. (2011: 1) define recommender systems as "software tools and techniques providing suggestions for items to be of use of a user." Adomavicius and Tuzhilin (2005: 734) formally formulate the recommendation problem as follows:

C: Set of all users

S: Set of all items that can be recommended

u: A utility function that measures the usefulness of item s to the user c, i. e., $u: C \times S \rightarrow R$, where R is a totally ordered set.

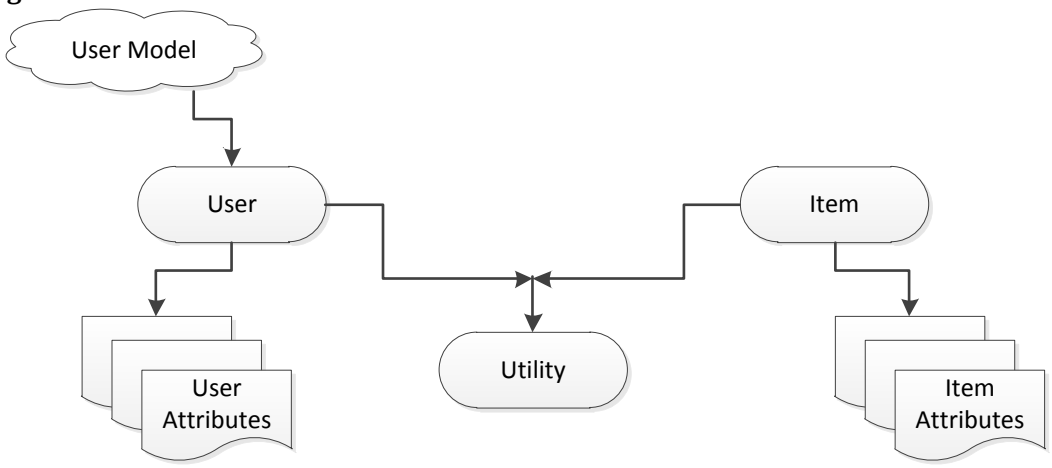
Then, the recommendation problem is *to choose the item that maximizes* user's utility:

$$\forall c \in C, \quad s_c^{'} = \arg\max_{s \in S} u(c, s) \tag{1}$$

1.1.2. Components

Adomavicius and Tuzhilin's definition, along with many others (Burke, 2002: 331; Resnick and Varian, 1997:57; Ricci et al., 2011:1), points out the core components of a recommendation model as *user* and *item*. What links these two components together is the *utility* user gained from the item.

Figure 1: Basic Recommendation Model



Source: Konstan and Ekstrand, 2013

Figure 1 illustrates the main components of a basic recommendation model. In a basic recommendation model, the output of the system is a set of item recommendations maximizing user's utility. Inputs of the system, on the other hand, can include user attributes (e.g. demographics) or item attributes (e.g. genre of a movie, type of a song, size of a mobile phone, etc.) User model, then, can be built upon those item properties by profiling the user with user attributes and user's preferences on item attributes. Ricci et al. (2011: 7) analyze the roles of these components as the data and knowledge sources of a recommender system and classify into three categories:

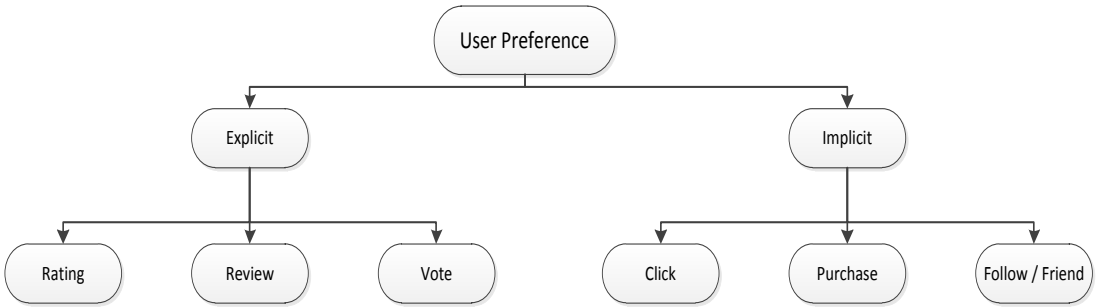
1) *User*: User component of a recommender system is the person who gets the recommendations from the system. For personalized recommender systems, user information plays a key role as personalization is not possible without a viable user model. In user profiling, user data can further be subdivided into three data sources:

a. *Demographics*: Demographics can be seen as the bottom layer in the personalization process. While they are very powerful at filtering in the earlier stages of the process, these properties can help understand the user needs and expectations only to a certain degree.
b. *Trust*: In social networks, connections among the users of the community can help identify and measure a level of trust which in turn can be used as inputs in recommender systems. Rather than considering the whole community, trust-based recommender systems are useful when a variety of items is in consideration.
c. *User-Item Interactions*: A more direct way to get a sense of how much of a use an item to a user is to investigate the history of user-item interactions. These interactions are typically stored as logs in the database and can answer questions like: Is the user interested in the item? Has she already bought the item? If yes, has she expressed any opinions? Since these interactions constitute a history for the user, when user-item interactions are in focus, the systems have the assumption that user preferences are stable and do not change over time.

2) *Item*: Items are the objects that are recommended to users by the system. The type of the item is central to the recommender system as it affects both the choice of data source and the technique utilized by the system. Items can be products – digital or physical, or services. Their cost and complexity are also used in item categorization. For example, books, movies, web pages, songs are of low complexity and cost category. Cars, mobile phones, houses, travel plans, on the other hand, are of high complexity and cost category. The items in the high complexity and cost category are not frequently bought items and the decision process may require a higher level of interaction with the system.

3) *Transactions*: As we have previously mentioned in the "user-item interactions" category, a great deal of information about user's preferences comes from human-computer interactions. These transactions can help us determine the utility of an item. Users have many ways to express their opinions about their interest, gained utility or usefulness of an item. Figure 2 summarizes how users express their preferences.

Figure 2: Users' Preference Expressions



Source: Konstan and Ekstrand, 2013

The main way for a user to share her opinion about an item is to explicitly evaluate that item. This can be in the form of ratings, generally on a Likert scale. User may also choose to express her opinions in words to give a more detailed feedback. A less informative category, votes, is commonly used in today's Web 2.0 products. Votes can either be in binary form (like/dislike, good/bad) or in unary form where only negative opinions are shared. While explicit preference sharing has many advantages, it also has disadvantages since only a limited proportion of users use these evaluation tools.

Implicit preference sharing is in the unary form and shows if the user has clicked on a link, purchased an item or is a friend with another user. While the amount of data of implicit preferences is certainly larger than the explicit preferences, they are harder to evaluate. There might be the cases where the user deliberately chooses not to click on a link or not to purchase a product. There might also be cases where the user purchased a product but did not like it.

The last consideration about users' preference sharing is the time when it is provided. Users may share their opinions at the time of consumption, where the

radical opinions are more likely to surface or they may share from memory. It is also known that users may share opinions about items they have not used. These can provide information about users' expectations and preferences about item attributes but also raise difficulty of differentiating from experience-based opinions.

1.2. RECOMMENDATION TECHNIQUES

In this section we present most commonly used personalized recommendation techniques following Jannach et al.'s categorization (2010: 2): Collaborative recommendation, content-based recommendation, knowledge-based recommendation and hybrid approaches.

1.2.1. Collaborative Recommendation

Collaborative recommendation, the most popular and widely implemented technique in recommender systems (Ricci et al., 2011: 12), considers only the past history of user-item interactions and proposes a model based on the similarities between users.

In collaborative recommendation approaches, the utility of an item for a particular user is predicted by the expressed utility of the other users who are, in several ways, similar to the user and believed to have similar taste in that particular item as well.

In its purest form, the only input of the system is the user-item ratings and the system produces a prediction on the utility function value and a list of *n* recommended items where the items user has already bought are excluded (Jannach, 2011: 13). This type of similarities between users in pure collaborative approaches are also called people-to-people correlations (Burke, 2002: 333).

The first known collaborative recommendation approach was used in Tapestry system, which was used to filter mails in the newsgroups (Goldberg et al., 1992). The study is also known as the first study to use the term "collaborative filtering". Several other collaborative approaches followed the Tapestry system in the early 90's. GroupLens (Resnick et al., 1994) was also a document filtering

system in an open community and automated the recommendation process. Ringo (Shardanand and Maes, 1995) was one of the first systems to apply recommendation techniques to recommend songs and artists. At the same year, Hill et al. proposed the Bellcore Video Recommender to suggest movies to the users with similar taste.

The simplistic nature of collaborative filtering along with its high performance results made this approach highly popular. Also, since this approach only uses the user-item ratings, the application domain is limitless as no attribute data for user or the item is required.

Collaborative filtering approaches are mainly studied under two categories (Jannach et al., 2011: 26):

- Memory based approach
- Model based approach

The main difference between these two approaches is that the former requires all the rating information to be held in memory while the latter requires only the learned model produced from an offline processing beforehand which allows for good scalability (Symeonidis et al., 2014: 53).

1.2.1.1. Memory Based Approaches

In memory based approaches, the user-item rating matrix is required at the time of the creation of the recommendation in order to form the neighborhoods of either the user or the item. Therefore, they are classified as user-based and item-based neighbor recommendation techniques.

a) User-Based Nearest Neighbor Recommendation

In user-based nearest neighbor recommendation, the utility values that the user has not yet evaluated are predicted with the help of other users' evaluations on that item. Here, other users are called the neighbors of the user and this neighborhood is constructed via several similarity measures.

User-based nearest neighbor recommendation has two assumptions: "if users had similar tastes in the past, they will have similar tastes in the future and user preferences remain stable and consistent over time." (Jannach, 2011: 14)

Let r_{ui} be the rating of user *u* for item *i*. Then, we can use the ratings from the users who are most similar the user *u.* These users are called the k-nearest neighbors (*k*-NN*)* of the user and denoted by $\mathrm{N}(\mathrm{u})$. The subset of these users who also rated the item *i* are denoted by $\mathrm{N}_i(\mathrm{u})$, so r_{ui} can be estimated by averaging the ratings given by these neighbors (Desrosiers and Karypis, 2011: 115):

$$\hat{r}_{ui} = \frac{1}{\left|\mathrm{N}_i(u)\right|} \sum_{v \in \mathrm{N}_i(u)} r_{vi} \tag{2}$$

Although the approach in (2) predicts the rating based on neighbors' ratings, it does not take into account that user similarities can have different levels. A common approach to fix this issue is to give different weights on the neighbors' ratings based on the similarity function.

Assuming the similarity between user *u* and user *v* is w_{uv}, we can calculate the weighted average for the rating estimation as:

$$\hat{r}_{ui} = \frac{\sum_{v \in \mathrm{N}_i(u)} w_{uv} r_{vi}}{\sum_{v \in \mathrm{N}_i(u)} \left|w_{uv}\right|} \tag{3}$$

where the absolute value function ensures that the estimated rating is in the allowed range (Desrosiers and Karypis, 2011: 115).

One important flaw in equation (3) is that it does not take into account the fact that users have different rating behaviors. For example, one user may only give 5 stars to only a few items, but another may give 5 stars for most of the items she like. This issue can be overcome by applying a variance weighting factor to increase the influence of the items that have a high variance, a significance weighting to eliminate the effect of few commonly rated items, or case amplification to emphasize the values close to +1 and -1 (Jannach, 2011: 17). Simpler approaches include mean-centering and z-score normalization (Desrosiers and Karypis, 2011: 123).

Another issue in user-based nearest neighbor recommendation is to select a subset of users to do the calculations. It is possible to put a threshold on the *k* value

to get the best neighbors based on similarity and also a threshold on correlation value can be used (Herlocker et al., 2002: 299).

b) Item-Based Nearest Neighbor Recommendation

In large-scale recommender systems, the number of rated items by users is very small compared to the total number of users and the items, which causes the sparsity problem. For user-bases nearest recommendation algorithms, this can cause the system to recommend items based on only a few common ratings (Adomavicius and Tuzhilin, 2005: 740). The problem of sparsity also makes it impossible to compute the similarities in real time and calls for different techniques (Jannach et al., 2010: 18).

In item-based nearest neighbor recommendation, instead of computing the similarities between users, item similarities are computed. Then, the set of items $\mathrm{N}_u(\mathrm{i})$, the nearest neighbors of item *i* rated by user *u*, can be used to calculate the weighted predicted rating as follows (Desrosiers and Karypis, 2011: 117):

$$\hat{r}_{ui} = \frac{\sum_{j \in \mathrm{N}_u(i)} w_{ij} r_{uj}}{\sum_{j \in \mathrm{N}_u(i)} |w_{ij}|} \tag{4}$$

For both user and item-based nearest neighbor recommendation, the weights of the ratings are computed by means of a similarity function. One of the most commonly used similarity function is the Pearson correlation coefficient and for user similarity it can be calculated as follows:

$$w_{u,v} = \frac{\sum_{i \in I_u \cap I_v} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_u \cap I_v} (r_{\mathrm{v},i} - \bar{r}_v)^2}} \tag{5}$$

where I_u and I_v are the items rated by users *u* and *v*, respectively. Pearson correlation coefficient can also be used to compute the item similarities; however, it does not show good performance in this setting (Sarwar et al., 2001: 292).

Another similarity measure is the cosine similarity and it is computed in the vector space:

$$w_{u,v} = \frac{\mathbf{r}_u \cdot \mathbf{r}_v}{\|\mathbf{r}_u\|_2 \|\mathbf{r}_v\|_2} \tag{6}$$

The advantage of cosine similarity is that since it is computed in the vector space it can be used for unary data as well.

While Pearson correlation is widely used in recommender systems, since most of the systems deal with rank data, Spearman's rank correlation and Kendall's Tau were also suggested to compute similarities; however, their performances were very similar to the Pearson correlation (Herlocker et al., 2002: 294; Herlocker et al. 2004: 33).

1.2.1.2. Model Based Approaches

In model based approaches, instead of creating the model for every recommendation, the raw data is first processed offline to create a single model and then this model has the ability to make predictions without the use of historical data. Model based approaches include dimension reduction techniques (singular value decomposition and principal component analysis) and data mining techniques (association rule mining and machine learning methods).

a) Singular Value Decomposition

In singular value decomposition (SVD), the rating matrix is decomposed into three matrices such that:

$$\mathbf{R} = \mathbf{U}\Sigma\mathbf{T}^{\mathbf{T}} \tag{7}$$

where the elements of the diagonal matrix Σ are the singular values and **U** and **T** are orthogonal. Using the largest k singular values, the rating matrix can be approximated by using the intermediate vector space Σ_k instead of Σ. It, therefore, decreases the dimensionality which reduces the required storage and computational complexity and also since the smaller singular values are dropped, only the strongest effects appear in the model (Ekstrand et al., 2011: 102).

The algorithms that deploy singular value decomposition generate predictions by first computing the topic-relevance factor, *u*:

$$\mathbf{u} = (\mathbf{\Sigma T}^T)^{-1}\mathbf{r}_u \tag{8}$$

After the topic-relevance factors for users are calculated, the user's preferences for all items p_u can be calculated as follows:

$$\mathbf{p}_u = \mathbf{u\Sigma T}^T \tag{9}$$

Examples of SVD in recommender systems can be seen in several works (Sarwar et al., 2000; Goldberg et al., 2001; Hoffman et al., 2005).

b) Principal Component Analysis

Another approach to reduce dimensionality of the rating matrix was used by Goldberg et al. (2001) in Eigentaste algorithm to recommend jokes.

They used principal component analysis (PCA) on the correlation matrix computed from the normalized matrix to decompose it into three components:

$$\mathbf{C} = \mathbf{E}^T \Lambda \mathbf{E} \tag{10}$$

where $\mathbf{\Lambda}$ is a diagonal matrix with eigenvalues of the matrix $\mathbf{C}$. Taking only the largest eigenvalues into account, it is possible to represent most of the variation stored in the correlation matrix. After the users are clustered into k-dimensional space, the algorithm makes the recommendations based on the cluster of the user.

c) Association Rule Mining

The main idea behind association rule mining to exploit relationships between commonly occurring transactions. Lots of web sites deploy association rule mining to recommend items to their users via statements like "customers who bought this product also purchased these products". Personalization in association rule mining is very limited as it is for all of the collaborative recommendation techniques: it only considers the user-item interactions. However, it has the advantage of capturing surprising patterns by its support and confidence metrics.

Following the notation from Jannach et al. (2010: 32), let I be the item set with the subsets of X and Y. The rule $X \Rightarrow Y$ implies that if the user interacts with

item set X, then she interacts with item set Y as well. The support of this rule is the ratio of the transactions where both X and Y appear:

$$\text{support} = \frac{\text{number of transactions containing } X \cup Y}{\text{number of transactions}} \tag{11}$$

The confidence of the rule, on the other hand, is computed on a reduced sample space of transactions containing X:

$$\text{confidence} = \frac{\text{number of transactions containing } X \cup Y}{\text{number of transactions containing } X} \tag{12}$$

The support metric of the rule shows how frequently these two items appear in the transactions whereas the confidence metric adjusts this value considering the frequency of the premise. The confidence metric allows eliminating the rules for items that appear frequently by themselves. The adaptations of association rules in recommender systems can be seen in (Sarwar et al., 2000; Mobasher et al., 2001).

d) Machine Learning Methods

As the recommendation system has its roots in information retrieval, several machine learning methods were also used in recommendation algorithms: clustering, Bayesian networks, artificial neural networks and decision trees are among the techniques applied in recommender systems.

.

1.2.1.3. Limitations

The basic collaborative recommendation approaches are powerful in the sense that they only require the user-item interactions to make the recommendations. However, they have several shortcomings as pointed out by Adomavicius and Tuzhilin (2005: 740):

a) New User Problem

When a new user enters the system, since the data about the user will be limited, the collaborative approaches fail to recommend items to the user. As we mentioned earlier, both the thresholds on similarity values and the neighborhood size prevent the system from making recommendations. Non-personalized recommendation techniques can be deployed in this situation along with many other hybrid approaches.

b) New Item Problem

Like the new user problem, new item problem arises in collaborative approaches as the user-item interactions for these items are limited. Since the system is not able to find meaningful similarities between the items or between the users who rate them, until a group of users rate the new item it will stay out of the system. Approaches for solving new user problem can also be applied for this issue.

c) Sparsity

For many recommender systems, the number of users is substantially larger than the number of items. Therefore, sparsity problem surfaces mostly on user-based neighbor recommendation techniques. Most common ways to address this problem is to deploy dimensionality reduction techniques on the user-item rating matrices.

1.2.2. Content-Based Recommendation

Collaborative recommendation techniques only use user-item interactions for recommendation. However, it is possible to integrate additional information about the items to the recommender system. Content-based recommendation works by building a *keyword* or a *taste vector* from users' ratings. Going back to the basic recommender model in Figure 1, the vector of preferences is built on the item attributes. These item attributes, for the case of movies, can be the genre of a movie

(i.e. action, comedy, and drama), actors or actresses starring in it, director or the author of the movie. The content-based recommendation uses these attributes to build user profiles: How much does the user like action movies? Does the user prefer Hitchcock's movies over Kubrick's movies?

Having its roots in information retrieval, most content based systems focus on recommending items whose attributes are usually represented by keywords (Adomavicius and Tuzhilin, 2005: 736). This representation can be in binary form: 1 if the keyword is in the document, 0 otherwise. To overcome the shortcomings of this approach, documents are generally described using *term frequency-inverse document frequency* (TF-IDF) format. Term frequency shows how commonly a keyword appears in a document. Regardless of the context, some keywords appear frequently in every document (i.e. "the", "I", and "you"). Inverse document frequency reduces the weights of these keywords and put more weight on less commonly appearing keywords.

Letting f_{ij} be the number of times keyword k_i appears in document d_j, the normalized frequency is computed as follows (Adomavicius and Tuzhilin, 2005: 736):

$$TF_{i,j} = \frac{f_{i,j}}{\max_z f_{z,j}} \tag{13}$$

For a total of N documents, where k_i appears in n_i of them, the inverse document frequency is:

$$IDF_i = \log \frac{N}{n_i} \tag{14}$$

Putting together (13) and (14), the TF-IDF weight for keyword k_i in document d_j is:

$$w_{i,j} = TF_{i,j} \times IDF_i \tag{15}$$

The content of the document d_j is a vector of the weights from (15):

$$Content(d_j) = (w_{1j}, \ldots, w_{kj}) \tag{16}$$

The content based profile of user *c, ContentBasedProfile(c),* is also a vector of weights where w_{ci} indicates the preference of user *c* towards the keyword k_i.

After constructing the content based profile of the user via Rocchio's algorithm (Balabanovic and Shoham, 1997), decision trees (Pazzani et al., 1996), Bayesian classifiers (Pazzani and Billsus, 1997) or Winnow algorithm (Pazzani, 1999); the predicted score of the item for the user can be found by measuring the cosine similarity of *ContentBasedProfile(c)* and *Content(s)* (Adomavicius and Tuzhilin, 2005: 736):

$$\text{score}(\text{ContentBasedProfile}(c),\text{Content}(s)) = \frac{\mathbf{w}_c \cdot \mathbf{w}_s}{\|\mathbf{w}_c\|_2 \times \|\mathbf{w}_s\|_2} \quad (17)$$

The advantages and disadvantages of content-based recommendations can be summarized as follows (Lops et al., 2011: 78):

Advantages:

- *User Independence*: Content-based systems only interact with the active user without requiring data from the *neighbors*.
- *Transparency*: Users can easily be provided with the information on how the recommendations are constructed.
- *New Item*: The only requirement for the new items in content-based systems is the vector of keywords which can immediately be constructed.

Disadvantages:

- *Limited Content Analysis*: There are only a limited number of features of the items that can be represented as keywords.
- *Over-Specialization*: Content-based recommenders tend to offer same kind of items all the time. They are not suitable for finding something unexpected.
- *New User*: As content-based recommenders build user profiles based on the items they rate, in order to recommend items user needs to rate some items first.

1.2.3. Knowledge-Based Recommendation

Sometimes classified under content-based recommendation, knowledge-based recommendation techniques aim to provide solutions in the cases where user or item similarities cannot help. For example, we may only buy a house once in a lifetime, we may only visit other countries a few times; by the time we need a new mobile phone, our needs, expectations may have changed as well as the features of the new mobile phones.

Knowledge-based recommender systems work interactively with users. Users can provide constraints on their end, and the system builds cases upon these constraints.

1.2.4. Hybrid Approaches

Hybrid recommender systems combine several recommendation techniques to improve performance and to eliminate the drawbacks of each technique. Burke (2002: 339) summarizes the hybrid approaches in seven categories:

1) *Weighted*: In weighted hybrid recommender systems, the scores are combined based on a weighing schema. This is generally a linear combination of several techniques.
2) *Switching*: These systems employ several recommender systems and based on some condition, they switch between the alternatives. Generally, when a recommender system fails to recommend an item (because of new user and new item problems) another technique comes into play.
3) *Mixed*: In mixed recommender systems recommendations comes from different techniques. Instead of combining the scores, these systems present the output from all the systems they have.
4) *Feature Combination*: In feature combination, the similarities come from the collaborative approach are represented as features in the content-based systems.

5) *Cascade*: In cascade systems, one of the recommender systems presents the recommendation and the other one is used for refining the outputs coming from the first system.
6) *Feature Augmentation*: Feature augmentation systems leverage the output coming from other recommender systems as new features.
7) *Meta-Level*: Like feature augmentation, in meta-level systems the first recommender system becomes the input for the second one. The difference is, meta-level systems use the models as inputs whereas the feature augmentation systems use the outputs of the models.

1.3. EVALUATING RECOMMENDER SYSTEMS

There are several evaluation metrics coming from other areas that applied to recommender systems.

Mean absolute error (MAE) calculates the average absolute deviations in predicted ratings from actual ratings:

$$MAE = \frac{\sum_{u \in U} \sum_{i \in testset_u} |\hat{r}_{ui} - r_{ui}|}{\sum_{u \in U} |testset_u|} \tag{18}$$

Another metric, mean squared error (MSE) calculates the squared deviations from the actual ratings:

$$MSE = \frac{\sum_{u \in U} \sum_{i \in testset_u} (\hat{r}_{ui} - r_{ui})^2}{\sum_{u \in U} |testset_u|} \tag{19}$$

As MSE is in squared units, in order to have a unit in line with the ratings, root mean squared error (RMSE) can also be calculated:

$$RMSE = \sqrt{\frac{\sum_{u \in U} \sum_{i \in testset_u} (\hat{r}_{ui} - r_{ui})^2}{\sum_{u \in U} |testset_u|}} \tag{20}$$

All these three metrics require numeric data and treat the rating data as numeric. As decision support metrics, precision and recall metrics from information retrieval field are also used in recommender systems.

$$precision = \frac{\left|\{relevant\ documents\} \cap \{retrieved\ documents\}\right|}{\left|\{retrieved\ documents\}\right|} \tag{21}$$

$$recall = \frac{\left|\{relevant\ documents\} \cap \{retrieved\ documents\}\right|}{\left|\{relevant\ documents\}\right|} \tag{22}$$

(21) and (22) give the definition of precision and recall in the information retrieval context. For their implications in the recommender systems we can interpret the relevant documents as "good recommendations" such that the system is able to recover the actual values excluded from the train set. Retrieved documents, on the other hand, refers to all recommendations.

Usually, precision and recall are computed taking all recommendations into account, but it is also possible to compute them at different cutoff (*N*) values. In this case, they are denoted as *precision@N* and *recall@N*.

In the classification context, precision refers to the ratio of true positives to all positives and recall refers to the ratio of true positives to the sum of true positives and false negatives (sensitivity).

Another measure proposed to combine these two metrics is F_β:

$$F_\beta = (1+\beta^2)\frac{precision \cdot recall}{\beta^2 \cdot precision + recall} \tag{23}$$

As the value of β increases, F_β puts more weight on recall. When $\beta = 1$ both metrics have equal weight.

CHAPTER TWO
RECOMMENDATION IN LOCATION BASED SOCIAL NETWORKS

This chapter starts with the definition and the structure of location based social networks and introduces the recommendation problem in these networks. Following the recommendation types, it finalizes with a brief review on the recommender systems for location based social networks.

2.1. LOCATION BASED SOCIAL NETWORKS

Location based social networks emerged from the advances in mobile and GPS technologies. Nowadays, most smartphones have the capability of detecting locations. The integration of location information can be seen in many social networks (e.g. Facebook, Twitter, and Google Plus).

Location based social networks, while carrying out the same features with social networks, put the location at the center of their structure (Symeonidis et al., 2014: 35). It is a relatively new tool for users sharing information with their social circle.

The biggest location based social network, Foursquare, was launched in 2009. Today, Foursquare has more than 50 million users with over 6 billion check-ins (Foursquare Inc., 2014). Other location based social networks, like Gowalla and Whrrl, didn't live long. Whrrl was acquired by Groupon in 2011, and Gowalla was acquired by Facebook in 2011. While Foursquare is the leading service in this area, social networks like Facebook, Twitter and Instagram have also integrated location info in their services. Facebook announced an application called Facebook Places where people can share the places they visit like Foursquare. The integration is not limited to sharing the location as a post, but these platforms have also made it possible to store location info in stories, tweets or pictures.

The strong attention of users to these services has caused many restaurants, bars, and hotels to try to become more visible in these services. The ratings, likes,

comments and tips in these services have had an effect on users' behavior in choosing where to go.

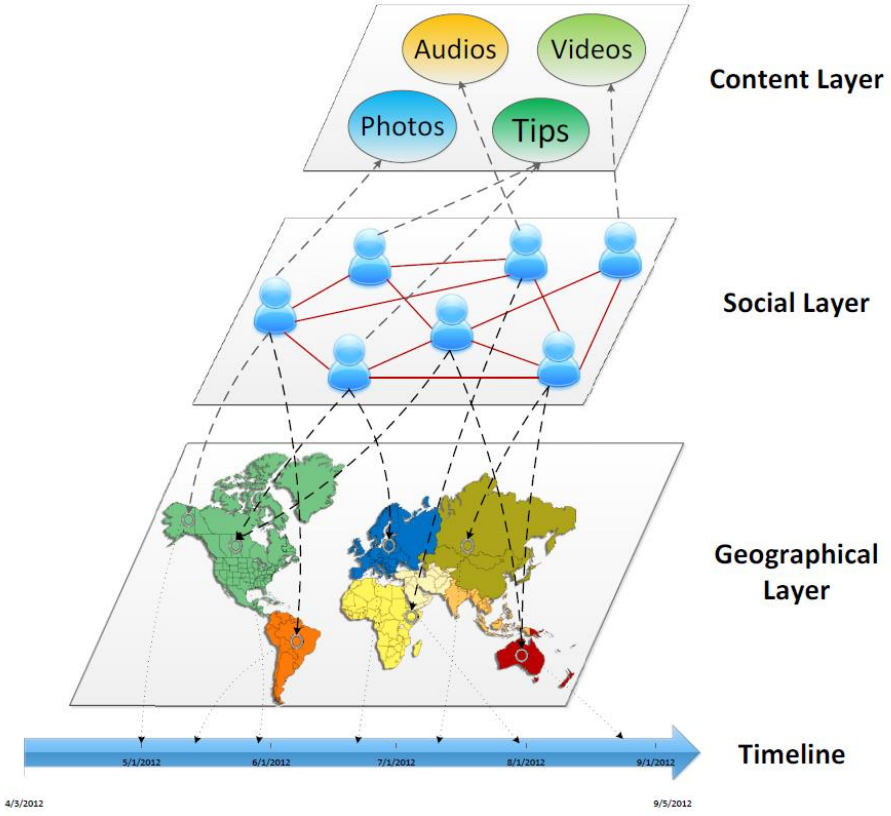
2.1.1. Structure of Location Based Social Networks

Location based social networks consist of three layers: geographical layer, social layer, and content layer (Symeonidis et al., 2014: 16). At the bottom of the structure is the geographical layer which consists of places, spots, or points-of-interests (POIs). On top of the geographical layer is the social layer. Social layer consists of users and their connections. The content these users share is at the top of the structure as the content layer.

The interconnections between these layers form the location based social networks. Users stand at the heart of these interactions with their connections to places and contents.

Gao and Liu (2014: 3) adds timeline to this structure as shown in Figure 3.

Figure 3: Structure of Location Based Social Networks



Source: Gao and Liu, 2014: 3

2.2. RECOMMENDATION IN LOCATION BASED SOCIAL NETWORKS

Recommendation in location based social networks is a new era due to the young history of these services. Much of the research conducted in this area uses publicly shared check-in data. Although it is possible for researchers to crawl these data, privacy concerns and regulations of the services prevents constructing and studying public datasets. This, in turn, becomes an obstacle in reproducible research.

Earlier studies in this area used GPS trajectories of users to model the mobility behavior of users and to recommend locations based on this analysis. Takeuchi and Sugimoto (2006) proposed a recommender system that uses GPS-based location histories of users to recommend shopping places. Zheng et al (2010) used collaborative filtering on GPS data to recommend both locations and activities. Yoon et al. (2010) recommended itineraries using GPS trajectories and user interactions.

2.2.1. Recommendation Types in Location Based Social Networks

Based on the layers in the structure of location based social networks, there can be four types of recommendations: Friend, location, activity, and event recommendations (Symeonidis et al., 2014: 59; Gao and Liu, 2014:11).

Friend recommendations in the location based social networks usually deals with recommending people user may know. This recommendation works on common friends in the social graph of the user.

Location recommendations mostly deal with restaurants, bars and holiday spots. Focused research on this area mostly investigates the geographical influence among places.

Activity recommendation is generally linked to the location recommendation as the context of the location sometimes implies an activity (e.g. sports, shopping, or eating).

Event recommendation is the one where time plays a role as dominant as geographical location. These types of recommendations usually require content information.

2.2.2. Recommender Systems for Location Based Social Networks

As mentioned earlier, within the five year history of location based social networks there have not been many studies on recommender systems. In this subsection, we present these studies by their recommendation types and the techniques they use.

Quercia and Capra (2009) proposed the algorithm *FriendSensing* to detect the people user may already know utilizing geographical proximity and link prediction. Symeonidis et al. (2011) developed an algorithm called FriendLink, which performs a local path traversal on the social circle to recommend friends. Scellatoet al. (2011) combined social, place and global features to predict the links in the social network resulting satisfactory recommendations even for people who do not share any friend or place.

Eventer algorithm (Kayaalp et al., 2009) recommends events based on user's location which is extracted from the IP address. The algorithm deploys a hybrid approach combining content-based and collaborative filtering techniques.

Zheng et al. (2010) combined location and activity recommendations utilizing collective matrix factorization to find interesting locations and activities. Sattari et al. (2012) also generated activity and location recommendations with their technique called Improved Feature Combination which integrates location-activity, activity-activity and location-feature matrices. Symeonidis et al.'s (2011) FriendSensing algorithm was part of their Incremental Tensor Reduction algorithm to provide both location and activity recommendations. They applied singular value decomposition to decompose a tensor to users, locations, and activities.

Noulas et al. (2012) built a random walk model for location recommendation. Ye et al. (2011) combined trust and geo-location information with user and item similarities to build a collaborative filtering recommender system. Cheng et al. (2013) proposed a location recommender system that utilizes embedded Markov chains in a matrix factorization method to recommend places to go next, after a certain venue is visited.

CHAPTER THREE
BUILDING THE RECOMMENDER SYSTEM

In this chapter we first present the data collection process and some preliminary results of descriptive analysis on the data. Then we build the recommender system alternatives and compare them on different metrics. We finalize our discussion with overall evaluation of the models.

3.1. DATA

This section summarizes the data collection process and gives the summary statistics on the research problem.

3.1.1. Data Collection

Our data collection process spans a period of 3 months from March 2014 to June 2014. The data is collected through Python programming language scripts connecting several application programming interfaces (APIs).

Our sample consists of Foursquare users who publicly share their check-in information through Twitter. We have limited the streaming response coming from Twitter API only to include the check-ins to the venues in Turkey.

The Twitter API response to the keyword "4sq.com" is a collection of JavaScript Object Notation (JSON) [1] objects filtering the real-time tweets containing the string "4sq.com". A sample JSON representation of a tweet can be seen in Appendix 1.

Due to the 140 character limitation of Twitter, many applications, including Foursquare, shorten the URLs sent to Twitter. Therefore, after the JSON object is

[1] JSON objects are semi-structured, dictionary like objects containing key-value pairs. It is a data-interchange format as an alternative to Comma Separated Values (CSV) files and Extensible Markup Language (XML). Details can be found in http://json.org/

parsed, if the {“entities”: {“urls”}} key of the object contains at least one valid URL, we connect to the Bitly API in order to expand that link. The expanded link, then, is stored under {“entities”: {“4sq_expanded”}} key of the object. This URL contains a “check-in ID” and a signature which allows retrieving the details of that check-in.

The main API that our data crawler connects to is the Foursquare API. /check-ins endpoint of the Foursquare API returns a JSON object containing the details of the check-in (see Appendix 2 for an example check-in object). We, then, use {checkin: {user: {id}}} and {checkin: {venue: {id}}} keys of the object to retrieve user and venue ID’s. These fields are then used to retrieve the data from /users, /users/friends and /venues endpoints.

As the streaming data collection requires a stable connection, we utilized several virtual private servers (VPSs) on Amazon Web Services (AWS) Cloud. For online data processing, we stored the data as raw text files on AWS S3 Cloud storage. For offline data processing, we used a NoSQL document database, MongoDB, as a local storage.

3.1.2. Descriptive Analysis

Our data collection process resulted in 6.7 million check-ins by 530 thousand users to 580 thousand venues. We also gathered a graph of 22 million user connections.

3.1.2.1. Venue Categories

Foursquare categorizes the venues into 10 main groups. In the second level of categorization, there are 42 categories with a total of 611 sub-categories in the third level.

In Table 1 we list the main categories along with the number of sub-categories they have.

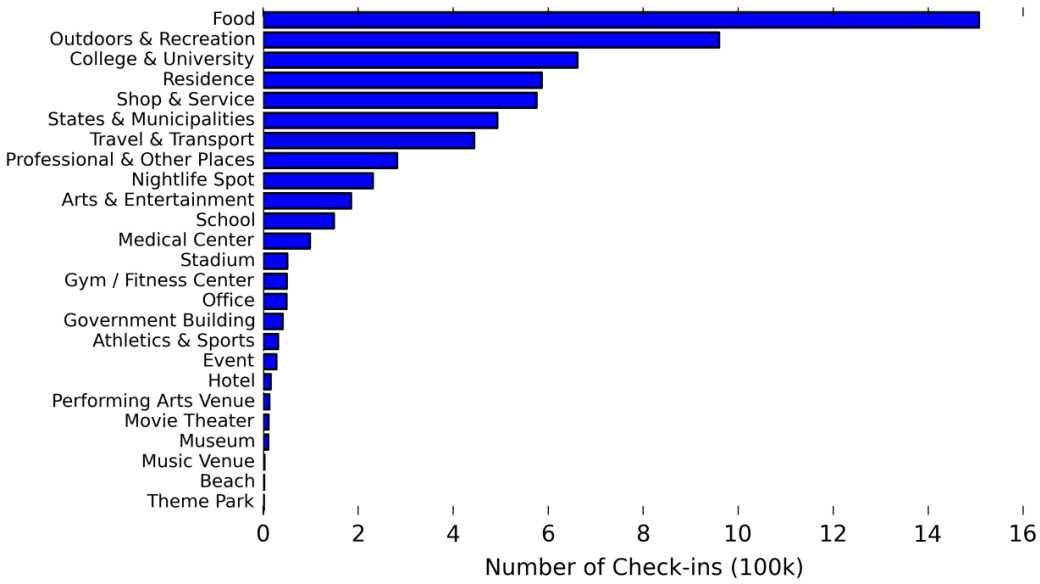
Here, it is important to note that as a social network, Foursquare lets its users to submit the categories of the venues when they are created by users. Therefore, it is possible to have misclassified venues.

Table 1: Venue Categories

Category Name	**Number of sub-categories**
Arts & Entertainment	53
College & University	37
Event	8
Food	141
Nightlife Spot	22
Outdoors & Recreation	77
Professional & Other Places	77
Residence	6
Shop & Service	146
Travel & Transport	44

For this study, we used the second level of categorization for the summary statistics as they represent important categories like medical centers, offices, movie theaters, stadiums, etc. Figure 4 gives a summary on the number of check-ins to each category in the second level.

Figure 4: Number of Check-ins by Categories



The most visited venue group is "Food" with Café and Restaurant subcategories. The next category, "Outdoors & Recreation" is mostly dominated by city and county check-ins. Third and fourth categories, "Universities" and "Residences" are not in the scope of recommender systems; however, they provide additional information about the user profile. For example, Foursquare does not provide age information of the users but our analysis shows that there are 89 thousand users who checked in to venues classified under "Universities" category at least three times. When we apply the same analysis on the "School" category (elementary, middle and high schools), we see that the number of users who checked in to schools at least three times is 44 thousand.

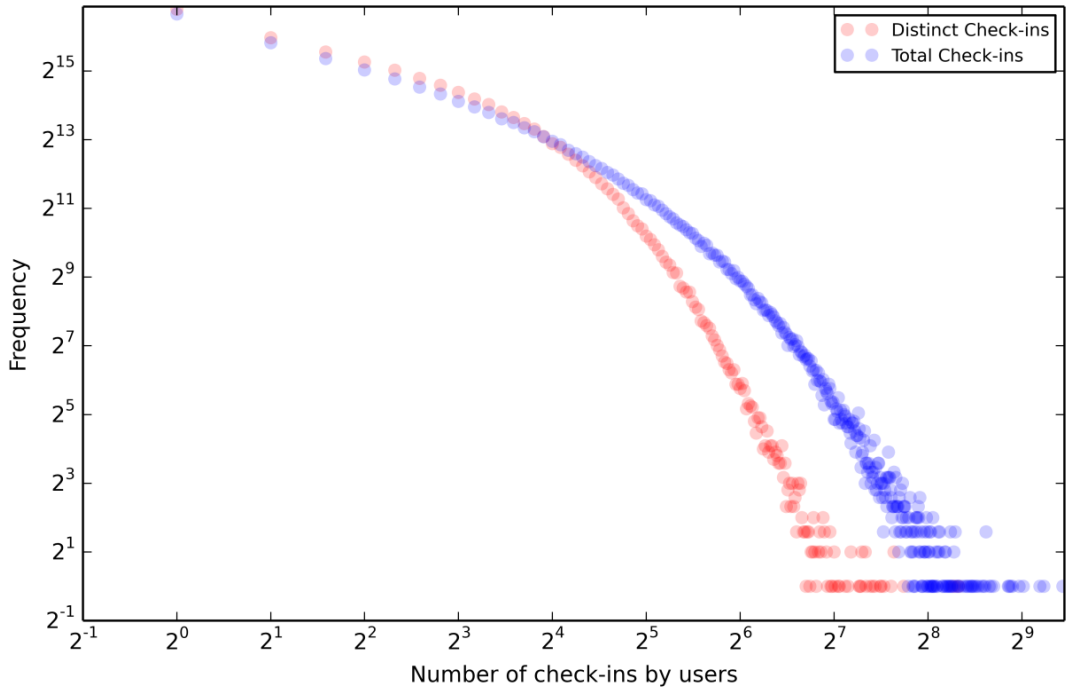
Another important category in this figure is the "Travel & Transport" category, with a total check-in count around 500 thousand. It shows that location-based social networks are also used when people are planning to visit other cities, possibly to inform their friends who live there.

"Professional & Other Places" category mostly contains the categories of offices, government buildings and medical centers. Although these groups have their own categories, since it is user-generated content, people may leave the sub-categories empty.

3.1.2.2. Check-in Frequency

In order to evaluate the user check-in behavior we also computed user check-in frequency distributions (see Figure 5 and Table 3).

Figure 5: User Check-in Frequency Distribution



As it is common in social networks, user check-in frequency follows a power-low distribution with peaks at single digit numbers and a very heavy tail. We present the distribution in log scale (base 2) to clearly see the behavior of the distribution at the tail. One important observation that can be drawn from this figure is that the ratio of number distinct check-ins (different venues visited by the user) to the total number of check-ins gets smaller and smaller as we approach to the tail. It shows that, as the total number of check-ins increases, venue diversity will be smaller. It is important to note that even though logarithmic transformation is applied on both scales, the relationship between the number of check-ins and their frequency is still non-linear.

Table 2: Summary Statistics for User Check-in Frequency Distribution

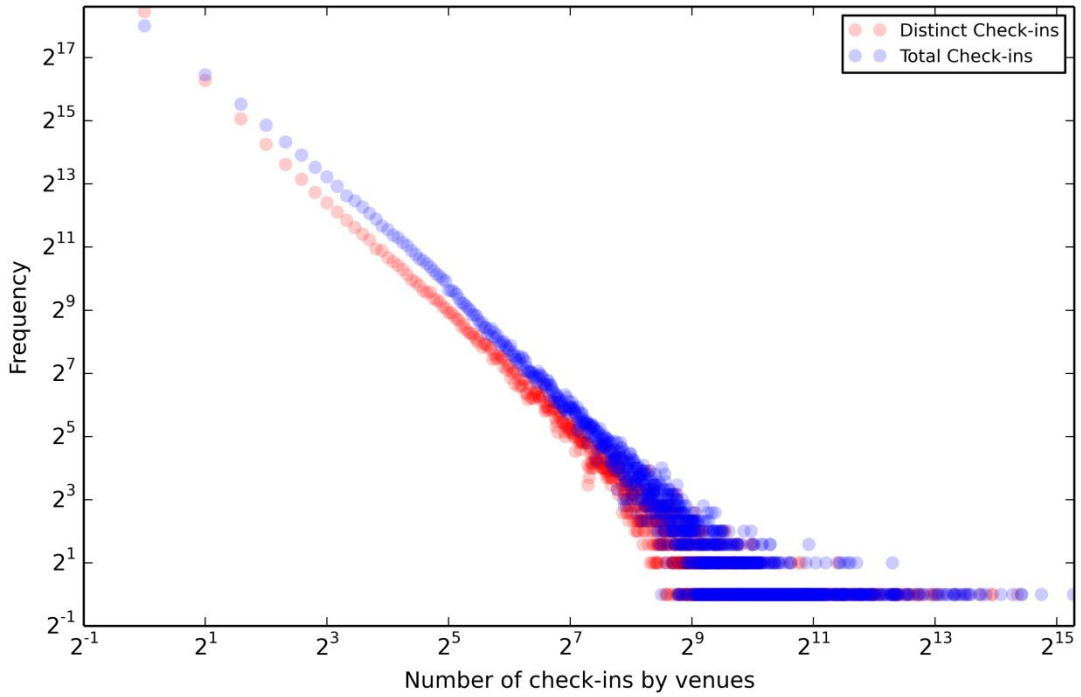
Statistic	Distinct Check-ins	Total Check-ins
Mean	7.58	12.12
Standard Deviation	8.67	18.20
Minimum	1	1
25th Percentile	2	2
Median	5	6
75th Percentile	10	15
Maximum	321	692

The skewness of the distribution can be seen in Table 2 more clearly. The 75th percentiles of distinct and total check-ins are 10 and 15, respectively. Maximum values show that 25% of the check-ins are spread on ranges 31 and 45 times larger than the ranges of check-ins in the first 75th percentiles.

The distribution of the check-ins suggests that most users use the service scarcely and a 3 month period of data collection may not be enough to build a successful recommender system. If we turn to our discussion on thresholds in Chapter 1, we can see that nearly half of the users will be treated as new users by the recommender system.

Venue check-in distribution follows a similar behavior to the user check-in distribution (see Figure 6).

Figure 6: Venue Check-in Frequency Distribution



The tail in the venue check-in distribution is a lot heavier than the user check-in distribution as expected. Also, the relationship between number of check-ins and frequency is linear for the venues.

Table 3: Summary Statistics for Venue Check-in Frequency Distribution

Statistic	Distinct Check-ins	Total Check-ins
Mean	7.03	11.23
Standard Deviation	79.79	130.759
Minimum	1	1
25th Percentile	1	1
Median	1	2
75th Percentile	2	5
Maximum	20885	39645

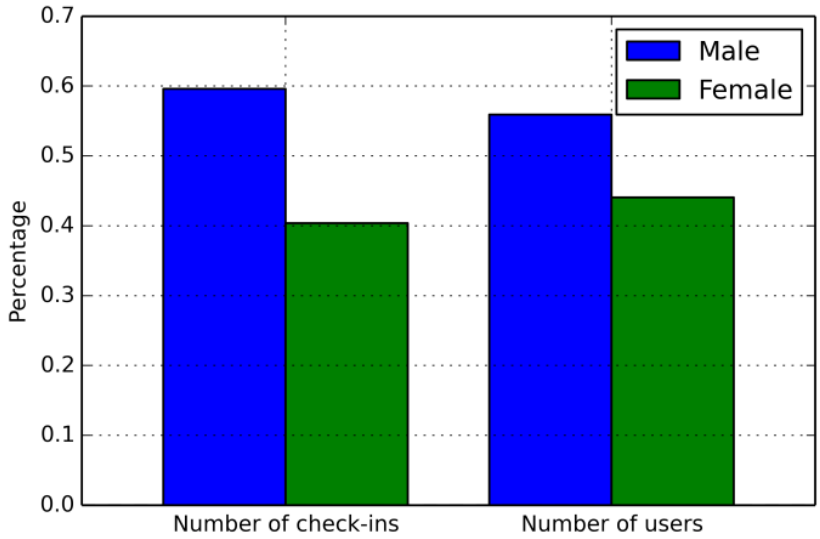
The skewness in venue check-in distribution is a lot higher due to the heavier tail it has. Considering the 75th percentile of the distribution we see that only to 25% of the venues more than 2 different users checked in. Venue check-in frequency

distribution also raises problems in the system as the recommendations for most of the venues will be prevented by the thresholds.

3.1.2.3. Gender

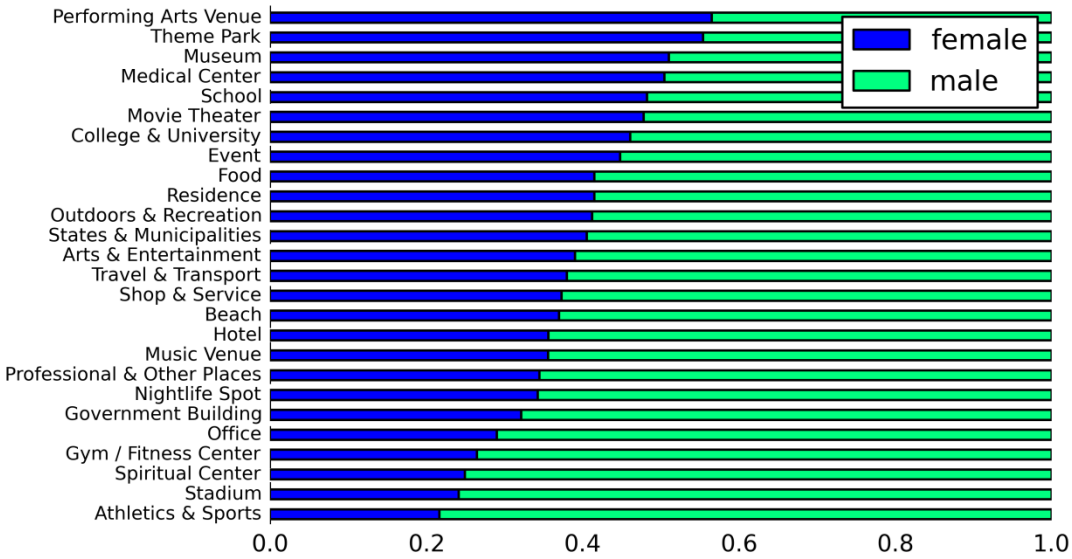
When we analyze the gender distribution, we see that 55% of the users are male and 45% are female (see Figure 7). Adding this to the number of check-ins by these users, it is clear that males use foursquare services more frequently than females (60-40%) although the average number of check-ins for each category seems to stay the same.

Figure 7: Number of Check-ins and Number of Users by Gender



In recommender systems, based on the application domain, gender can play an important role in the user profiling process. Because of that, in Figure 8 we investigated the percentage check-ins to each category from both genders.

Figure 8: Check-in Percentages by Category and Gender



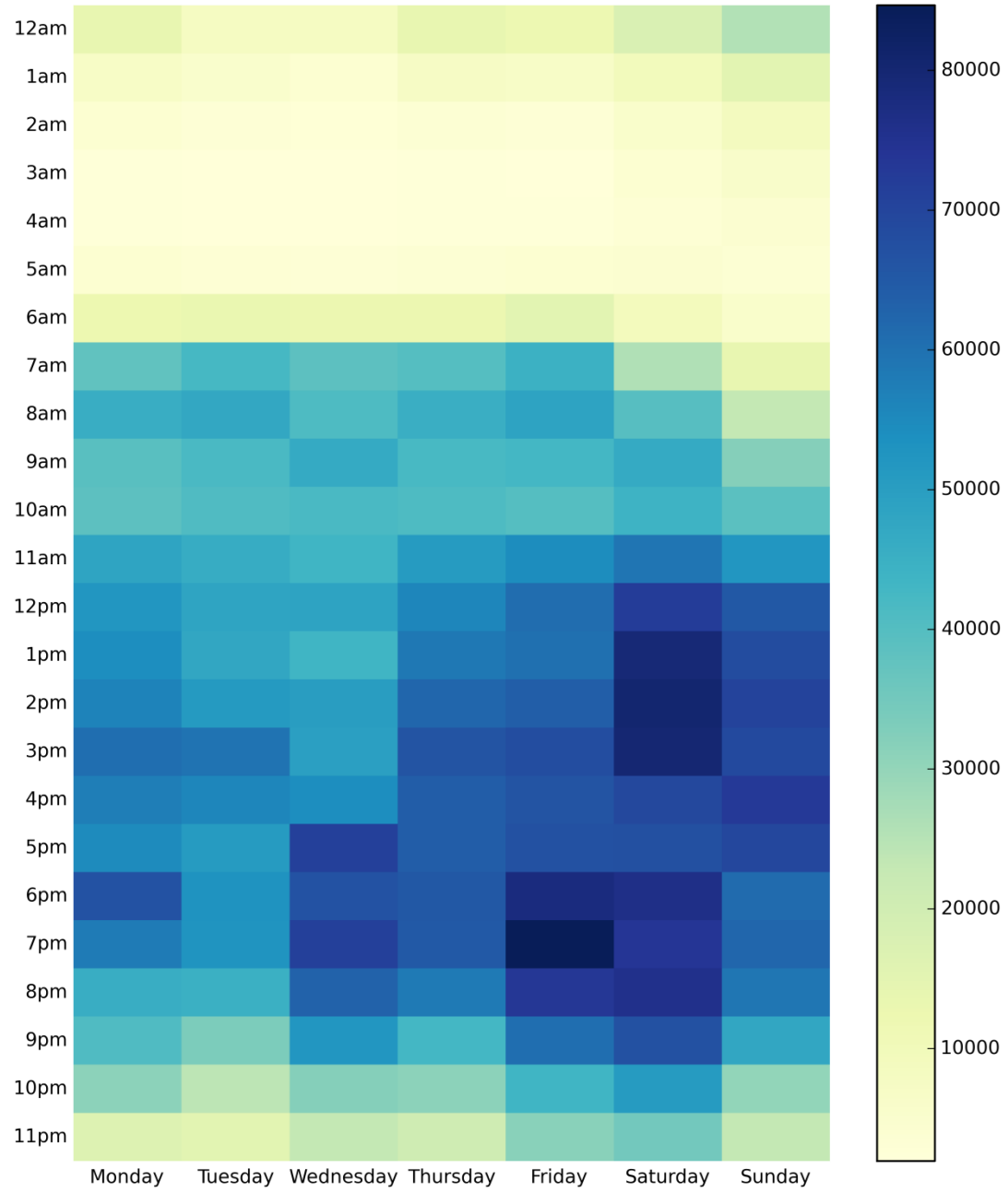
Female majority becomes prominent in “Performing Arts and Venue” and “Theme Park” categories. They also seem to be dominant in “Museum” and “Medical Center” categories. Adjusting for the overall percentages of the female users, the categories “School”, “Movie Theater”, “College and University” and “Event” appear to be in line with the general distribution. However, the remaining categories, especially the ones involving sports activities are mostly visited by male users.

In location based social networks, the suitability of recommending a venue may have dependencies on time and context as we see from these categories. Although the recommender systems are traditionally treated as the systems recommending the items user is not aware of, these systems can also work as intelligent systems recommending activities and events. Therefore, their domain is not constrained on the question “where to go”, but has the flexibility to answer questions like “where to go and what to do”. It is, then, important to capture the information resulting from these kinds of break-downs.

3.1.2.4. Time

Following the reasoning from the previous paragraph, we computed the check-in frequency by day and time in Figure 9.

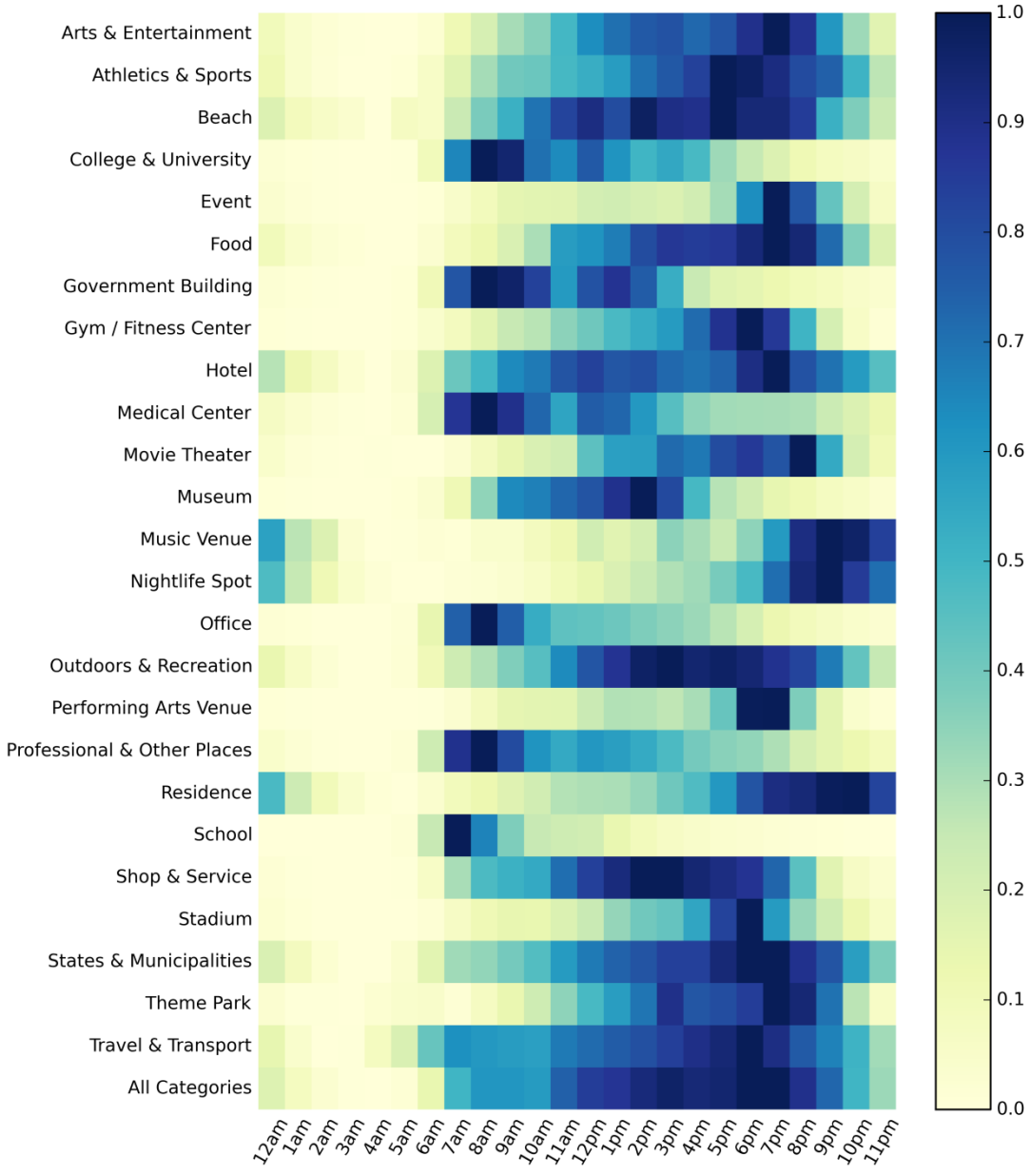
Figure 9: Total Number of Check-ins Grouped by Day and Time



From the figure we see that on weekdays, the density starts to reach the average values around 7-8 a.m. Except for Wednesday, all weekdays appear to have strong density around 3pm. Wednesday check-ins show a different pattern both at the start of the day and around the times people usually leave their work. The densest days are Friday (with a peak at 7 p.m.) and Saturday (with a peak around 1-3 p.m.). Both days also have high frequencies around midnight. Sunday shows a clear distinction from both weekdays and Saturday with the lowest frequency around 7-8

a.m. and highest frequency around 1-2 a.m. Most of these observations are not surprising as they are in line with our daily life patterns. We, therefore, made a further classification for categories to see the distribution of hourly check-ins for each category in Figure 10.

Figure 10: Hourly Check-in Distributions of Categories



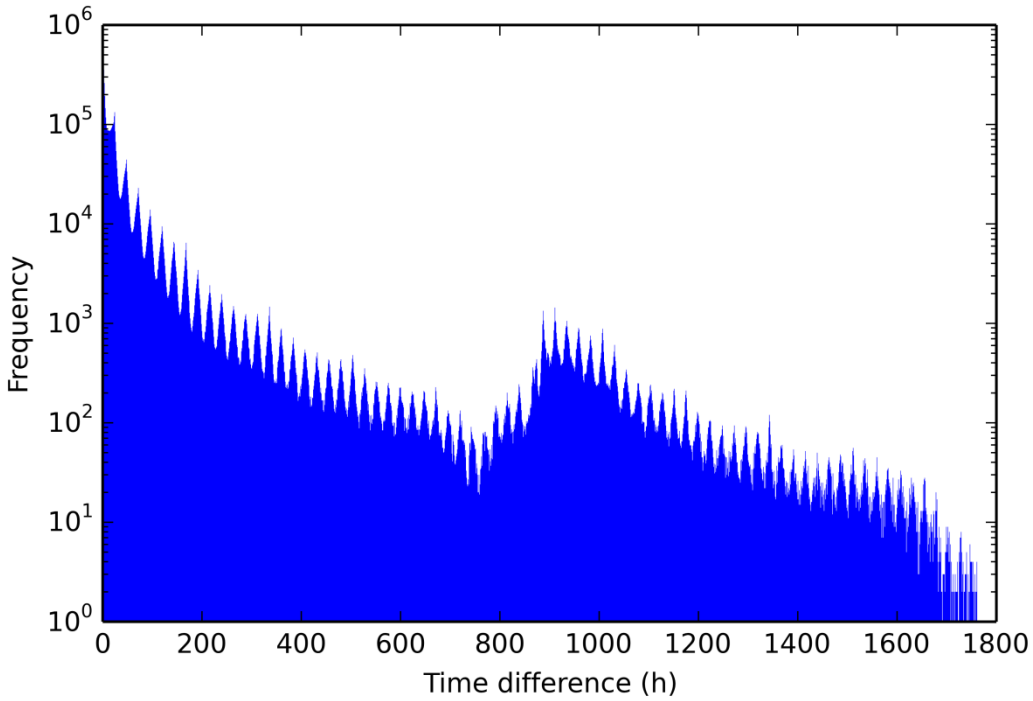
The figure is constructed by applying min-max normalization for each category. Therefore it shows the distribution of check-ins around different hours of the day and not suitable for comparison among the categories.

When all categories are considered, we see a distribution with two peaks around 3 p.m. and around 6-7 p.m. This implies that we have two different groups: possibly students and employees. Regular activities show themselves in School, Office, Professional & Other Places, Government Building and College & University categories with highest frequencies around 7-9 a.m. The places people choose to check-in around midnight are generally of Hotel, Music Venue, Nightlife Spot or Residence categories.

Other categories, in accordance with Figure 9, generally show peaks at times when students leave their schools or employees leave their work. Both figures show the need for time-awareness of a recommender system. As we previously mentioned in gender-category break-down of the check-ins, time-awareness can also be utilized for activity recommendation.

Figure 11 shows the distribution of inter-check-in times.

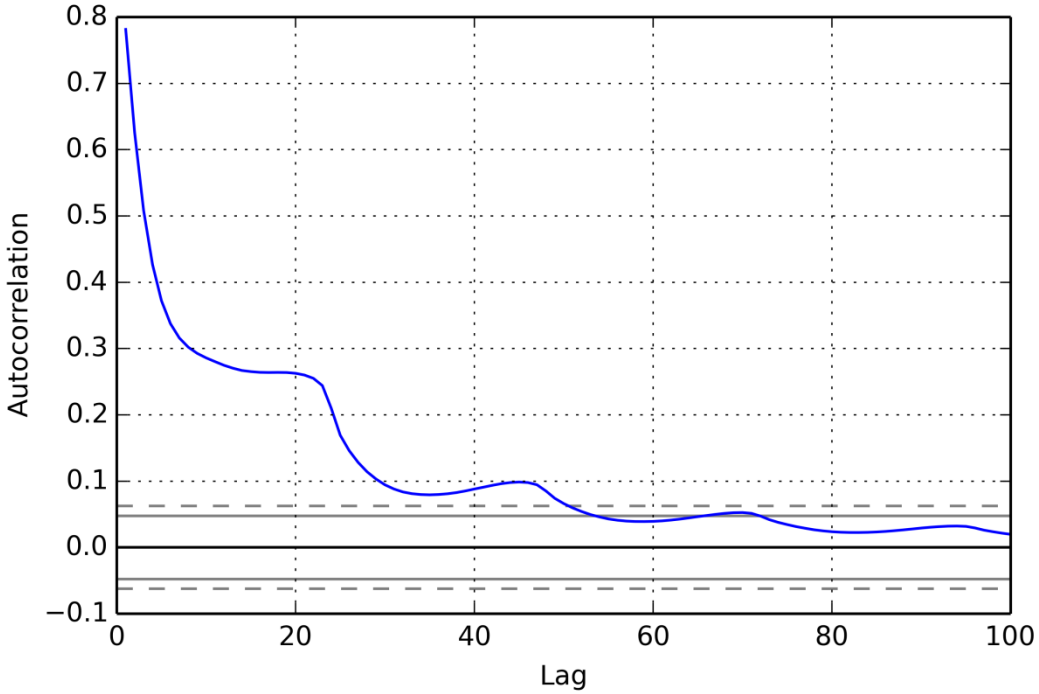
Figure 11: Time Difference Between Two Successive Check-ins



From the figure, we see that inter-check-in times also follow a right skewed distribution. There are two important aspects of this figure to consider: First, while the frequency decreases steadily until 800 hours, it starts to show an increase of more

than 10 times. We see that for non-regular users of the service, inter-check-in times come from another distribution yielding another peak around 950 hours. Second, for each day we see jumps on the distribution at certain time differences. To investigate this behavior more clearly, we constructed the correlogram of the time differences in Figure 12.

Figure 12: Correlogram of Time Differences



In the correlogram we can clearly see the daily seasonality effect on the time differences. While the autocorrelation function decreases steadily up to a certain point, the decrease stops around the 16th lag and stays on that level until 24th lag. This behavior repeats itself for the next lags and we see jumps around 48th and 72nd lags[2]. What this tells us is that even though there are many people who use the service at different times of the day, most users do their check-ins at the same time of the day independent of whether it is in the morning, or in the afternoon and whether there is one day between the check-ins or two days.

[2] The only significant partial autocorrelation value, except for the 1st lag, is at the 24th lag with a value of 0.08.

3.1.2.5. Location

For the location distribution of the check-ins, we considered two administrative areas: cities and counties. As the city and county information coming from the Foursquare API have many missing values, we used Global Administrative Areas Database (GADM) to determine the boundaries of each latitude and longitude pair. The choropleth maps for city and county levels can be seen in Figure 13 and Figure 14, respectively. A geographical heat map of the check-in distribution can also be seen in Appendix 3.

Figures show that the cities where the most check-ins occur are: İstanbul, İzmir, Ankara, Antalya, and Bursa. After that, Kocaeli, Muğla, Balıkesir, Manisa, Adana, Sakarya, Aydın, Mersin and Tekirdağ follow.

In Black Sea region, Samsun and Trabzon are the cities where people generally check-in. Ordu, Giresun and Rize come after these cities in the check-in density.

In Southeastern region, we see Gaziantep and Şanlıurfa as dense cities. The Eastern region seems to be the region where check-in density is at the lowest.

If we investigate the counties, we see that most of these check-ins occur in city centers. For İstanbul, the densest counties are Beşiktaş and Kadıköy and followed by Bakırköy, Fatih, Maltepe and Pendik. For İzmir, the densest county is Konak. Karşıyaka and Balçova follow Konak very closely. In Ankara, the prominent counties are Çankaya and Yenimahalle.

Figure 13: Choropleth Map of the Check-in Counts by Cities

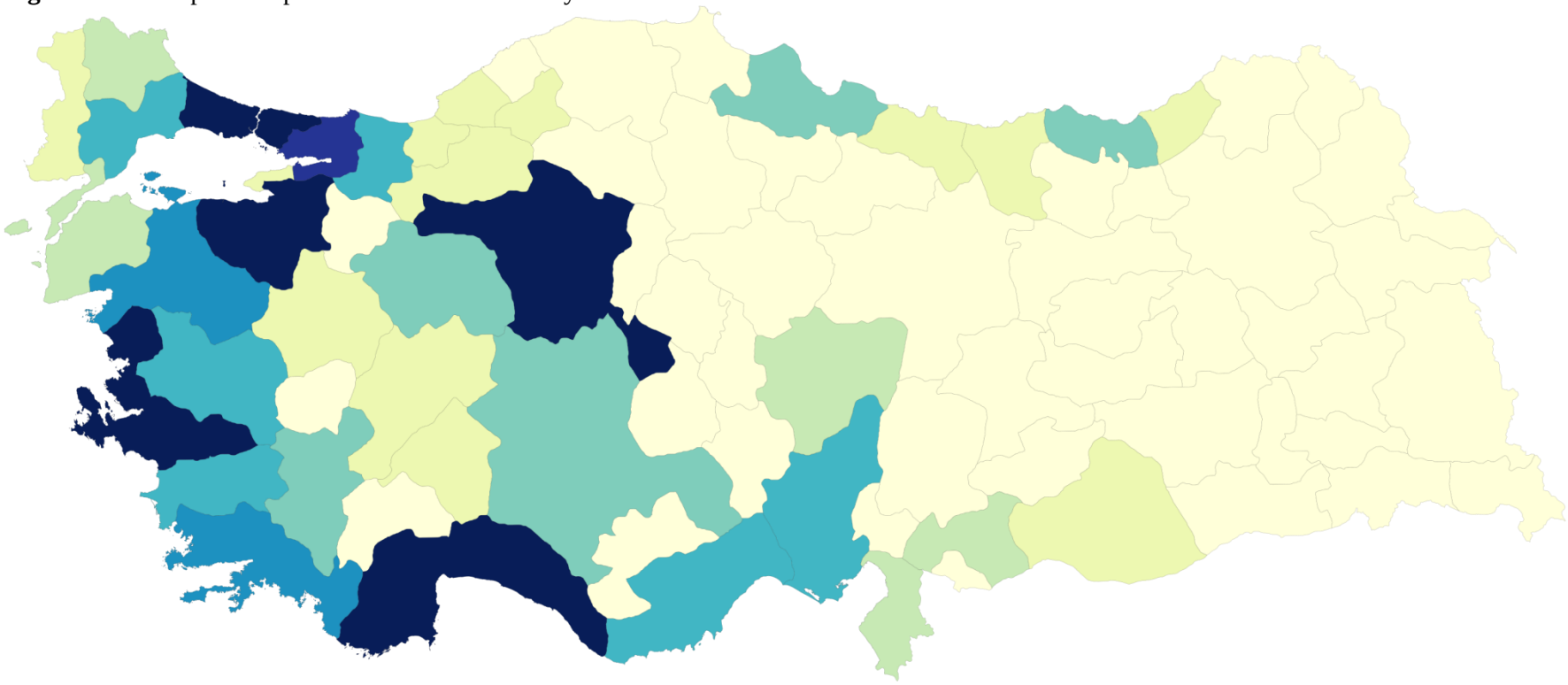
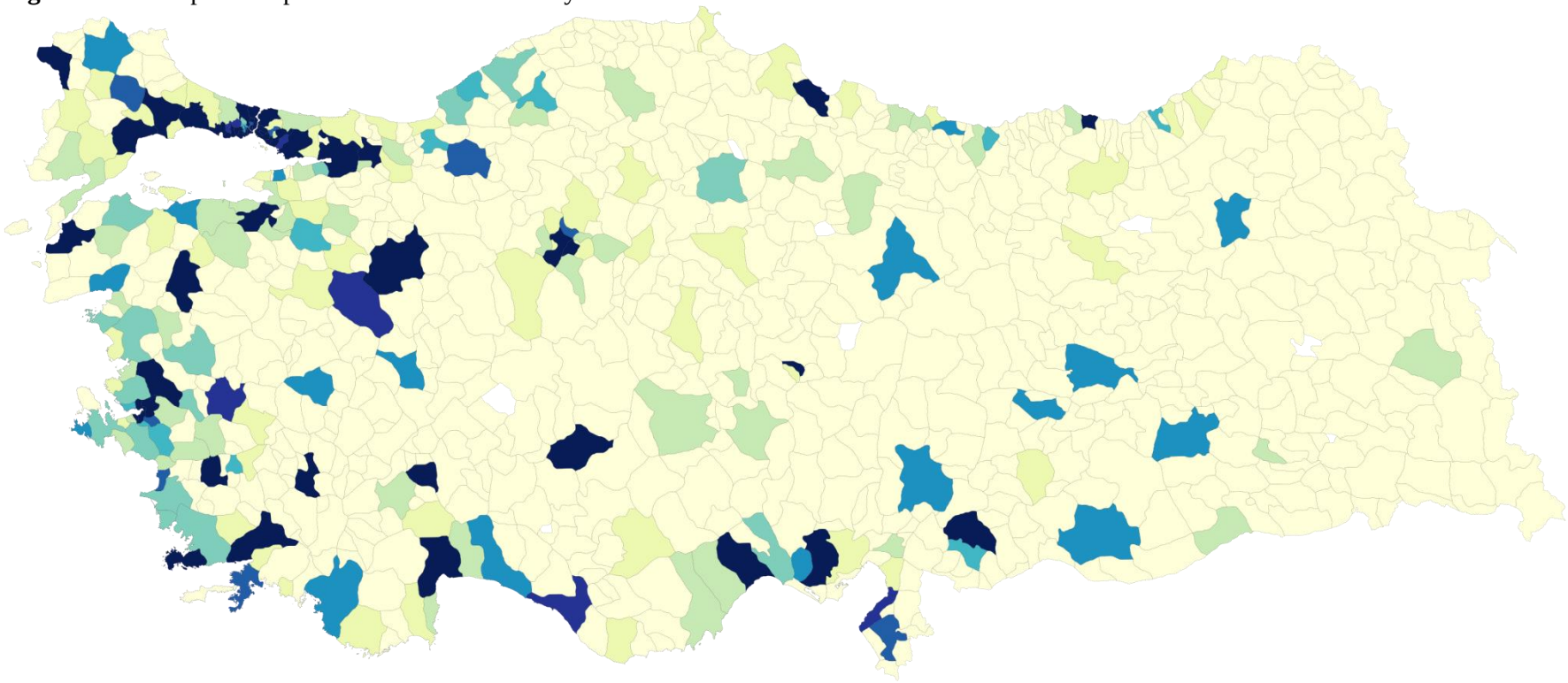


Figure 14: Choropleth Map of the Check-in Counts by Counties



3.2. MODEL BUILDING

In this section, our aim is to evaluate the performances of collaborative filtering techniques on recommending locations. We also aim to see the difference in the performance integrating a level of trust and geo-location information to the collaborative filtering.

3.2.1. Design

Check-in data is implicit and unary. Whether a person checks in to place does not imply that she enjoys spending time at that place. The principle also stands for the opposite: not having checked in to a place does not mean dislike towards that place. Because of that, the choices are very limited than the explicit rating data.

We choose collaborative filtering technique as it provides a good approach on handling unary data.

a) **Similarity:** For both user and item similarities, we used Cosine similarity measure as defined in (6). Cosine similarity can work on unary data as opposed to Pearson correlation. Another similarity measure that can work on unary data is Jaccard similarity. Jaccard similarity is a simple metric that counts the co-occurrences in two vectors. We deployed Jaccard similarity on friend-based collaborative filtering due to its simplicity in manual coding.

b) **Performance Measures:** In order to measure the performance, we used precision@N and recall@N along with their combined F_1 score. For unary data, we cannot calculate mean absolute error or mean squared error.

c) **Thresholds on Training Set:** As we mentioned in the previous chapters, both user and venue check-in frequencies are highly skewed. In order to prevent very high similarities with very low common points, we excluded

the users with less than three distinct check-ins and the venues with less than three distinct users. We included all categories in the training set as to capture the similarities even though many categories cannot be used in the recommendation.

d) Thresholds on Test Set: On the test set, we set higher threshold values in order to be able to calculate precision@N and recall@N values at larger N values. We randomly excluded half of the check-ins of some users and venues with more than five check-ins to evaluate as the test set. This yielded a ratio of 85%-15% for the training and test sets. Although we did use all categories in the performance calculations as to see the consistency of the system, we excluded the categories that are not relevant to our context (i.e., Residences, Universities, and Offices...)

e) Deployment and Algorithms: We utilized GraphLab parallel machine learning framework (Low et al., 2012). The parallelized computations were done on a cloud machine on AWS with 16 virtual CPUs and 30 GiB RAM. We used user-based collaborative filtering and item-based collaborative filtering techniques for the recommendations (1.2.1).

3.2.2. User-Based Collaborative Filtering

The first method we deploy is user-based collaborative filtering. This method has the highest complexity among the others since the similarities are calculated on the entire set.

Let $c_{i,u}$ be the binary variable indicating whether user i has checked in to the venue u and let I_i and I_j be the set of venues users i and j have checked in, respectively. Then, the cosine similarity between users i and j can be calculated as follows:

$$w_{\mathrm{i,j}} = \frac{\mathbf{c}_i \cdot \mathbf{c}_j}{\left\|\mathbf{c}_i\right\|_2 \left\|\mathbf{c}_j\right\|_2} = \frac{\sum_{u \in I_i \cup I_j} c_{i,u} c_{j,u}}{\sqrt{\sum_{u \in I_i \cup I_j} c_{i,u}^2} \sqrt{\sum_{u \in I_i \cup I_j} c_{\mathrm{j},u}^2}} \tag{24}$$

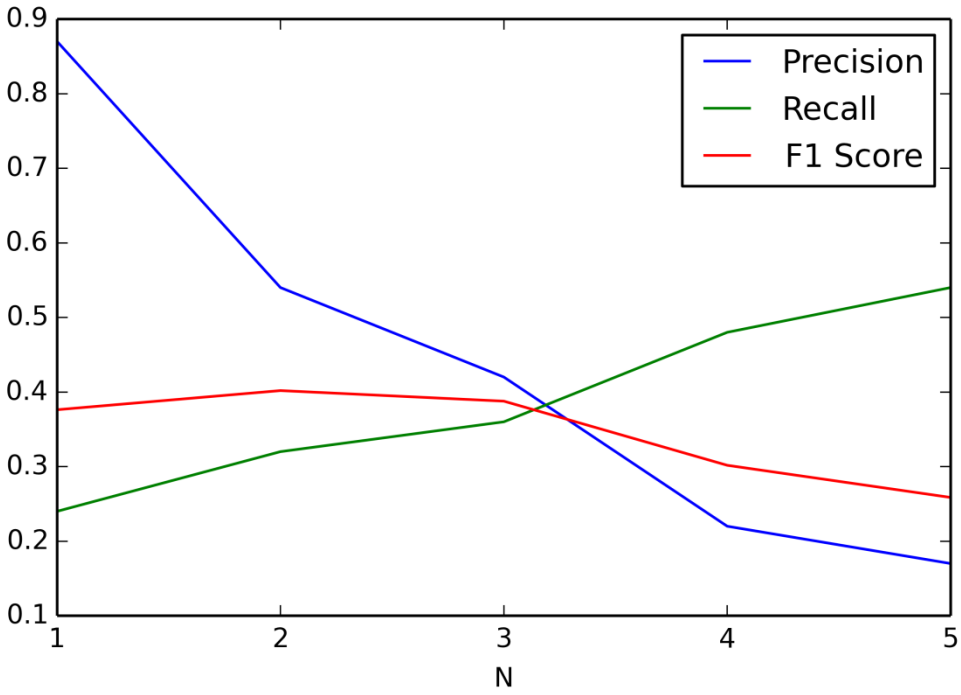
Note that we use the unions of the venues instead of the intersections as we do not have explicit ratings. For explicit ratings similarities can be calculated on the sets on the items users commonly rated. However, since the ratings we consider are 0 and 1, this would yield a similarity of 1 for all cases. Instead, we compute the similarities on the venues where either of the users has checked in.

After calculating the similarities, the estimation of the check-in score can be calculated as follows:

$$\hat{c}_{i,u} = \frac{\sum_{j \in N_i(u)} w_{i,j} c_{j,u}}{\sum_{j \in N_i(u)} w_{i,j}} \tag{25}$$

We present the performance metrics for user-based collaborative filtering in Figure 15.

Figure 15: Performance Metrics for User-Based Collaborative Filtering



Precision value in this setting refers to the ratio of venues, which were removed from the data set to the number of recommended venues. More specifically, precision@1 = 0.87 means that, if we only make one recommendation 87% of these recommendations will be the places these users visited and removed from the system.

Recall value, on the other hand, is the ratio of the removed locations that appear in the recommendation to the all removed locations of that user. Therefore, precision can be interpreted as the proportion of good recommendations in all recommendations; and recall can be interpreted as the proportion of good recommendations in all good candidate recommendations. Recall@1 = 0.24, then, means that for 24% of the deleted locations could be recovered by the recommender system.

The increase in recall as the number of recommendations increases is expected since the test set mostly has more than five venues for each user. Therefore, as we make more recommendations, it is possible to recover more of these deleted venues. However, this also decreases the precision value as we become more prone to the false positives.

In the figure we see a sharp decrease in precision from $N = 1$ to $N = 2$. Precision and recall balance each other around $N = 3$. The highest F_1 score we gained is at $N = 2$ as the increase in recall cannot account for the decrease in precision at larger N values.

3.2.3. Item-Based Collaborative Filtering

Similar to the user-based collaborative filtering, we compute the similarity between venue u and venue v as follows:

$$w_{\mathrm{u,v}} = \frac{\mathbf{c}_u \cdot \mathbf{c}_v}{\|\mathbf{c}_u\|_2 \|\mathbf{c}_v\|_2} = \frac{\sum_{i \in U_u \cup U_v} c_{i,u} c_{\mathrm{i,v}}}{\sqrt{\sum_{i \in U_u \cup U_v} c_{i,u}^2} \sqrt{\sum_{i \in U_u \cup U_v} c_{\mathrm{i,v}}^2}} \tag{26}$$

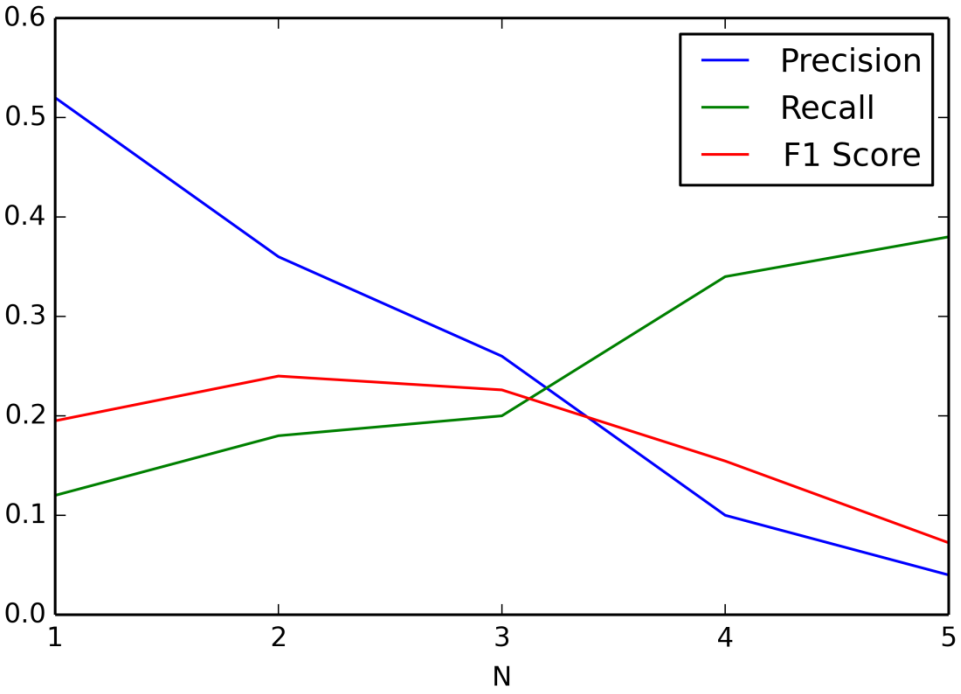
where U_u and U_v are the set of users who visited venue u and v, respectively.

Then, the check-in score can be calculated as follows:

$$\hat{c}_{i,u} = \frac{\sum_{v \in \mathrm{N}_u(i)} w_{u,v} c_{i,v}}{\sum_{v \in \mathrm{N}_u(i)} w_{u,v}} \tag{27}$$

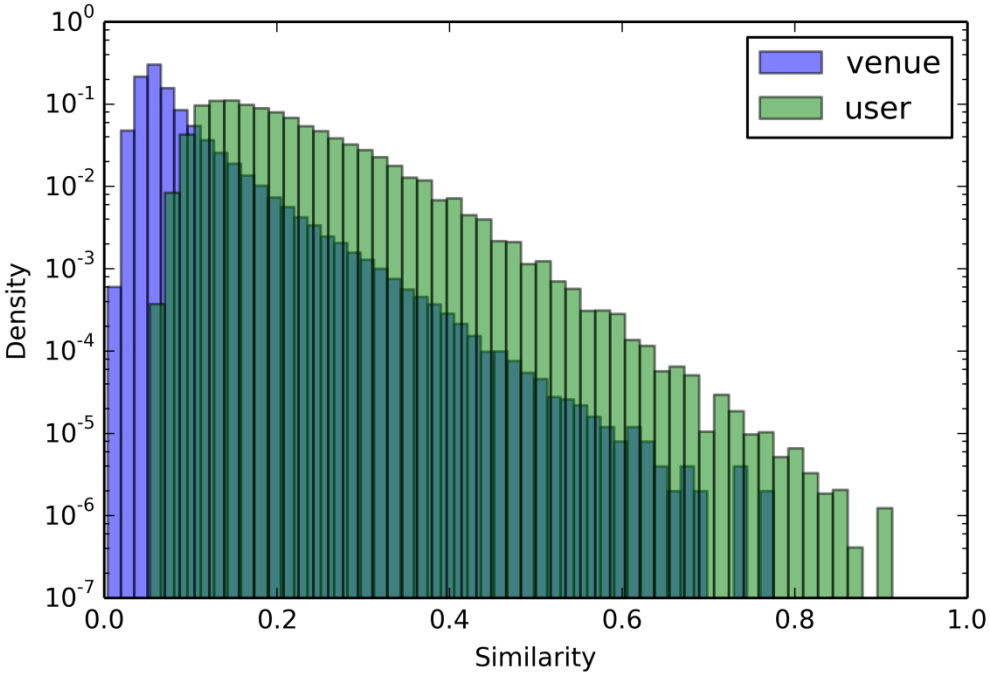
The summation is computed on the venues user i has already visited. Performance measures for item-based collaborative filtering are presented in Figure 16.

Figure 16: Performance Metrics for Item-Based Collaborative Filtering



We see from the figure that item-based collaborative filtering has poorer performance than user-based collaborative filtering. To see why this is the case, we computed the distributions of similarities for both methods in Figure 17.

Figure 17: Similarity Distributions



Venue similarities have very high frequencies at very low similarities. User similarities, on the other hand, span on a range with higher values. This is probably because the venues have a diverse set of users. For users, on the other hand, the set of venues is not so diverse. This can also be justified with user and venue check-in distributions we previously mentioned. The tail on the venue check-in distribution is a lot heavier than the user check-in distribution producing a larger set to compute similarities in. It is also important to note that these values can highly be affected by the design choices. Therefore, in order to generalize this result in location based social networks, further analysis on the threshold values may be required.

3.2.4. Friend-Based Collaborative Filtering

Trust plays an important role in recommendations. Especially in the location based social networks context, it is reasonable to expect friends to have more commonly visited locations than non-friends. Foursquare reports that the median user check-ins to a place that their social circle has been to is larger than 60% (Lee, 2011). Ye et al. (2010) also reports improvement on both performance and efficiency by using a friend-based collaborative filtering algorithm.

The similarity between user i and j can be computed as follows (Cheng et al. 2013):

$$w_{i,j} = \lambda \frac{\left|F_i \cap F_j\right|}{\left|F_i \cup F_j\right|} + (1-\lambda) \frac{\left|L_i \cap L_j\right|}{\left|L_i \cup L_j\right|} \tag{28}$$

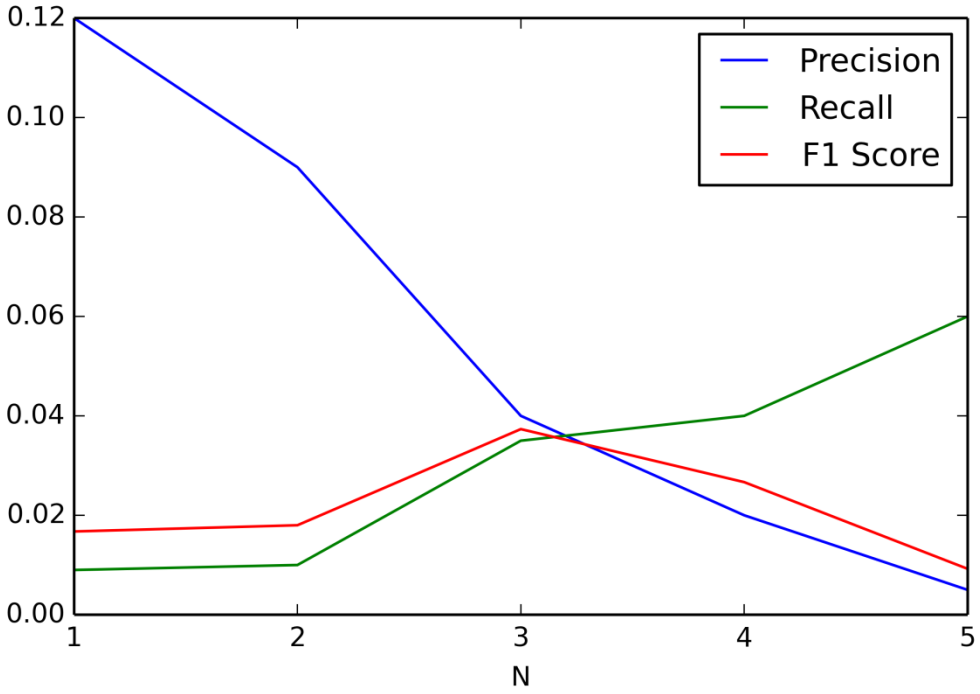
where F_i is the friend set of user i, L_i is the venue set of user i and λ is the weighing factor.

The check-in score, then, can be computed on a reduced set of users as follows:

$$\hat{c}_{i,u} = \frac{\sum_{j \in F_i} w_{i,j} c_{j,u}}{\sum_{j \in F_i} w_{i,j}} \tag{29}$$

The huge reduction in the set of users reduces the computational cost radically. We present the performance metrics of the algorithm in Figure 18.

Figure 18: Performance Metrics for Friend-Based Collaborative Filtering

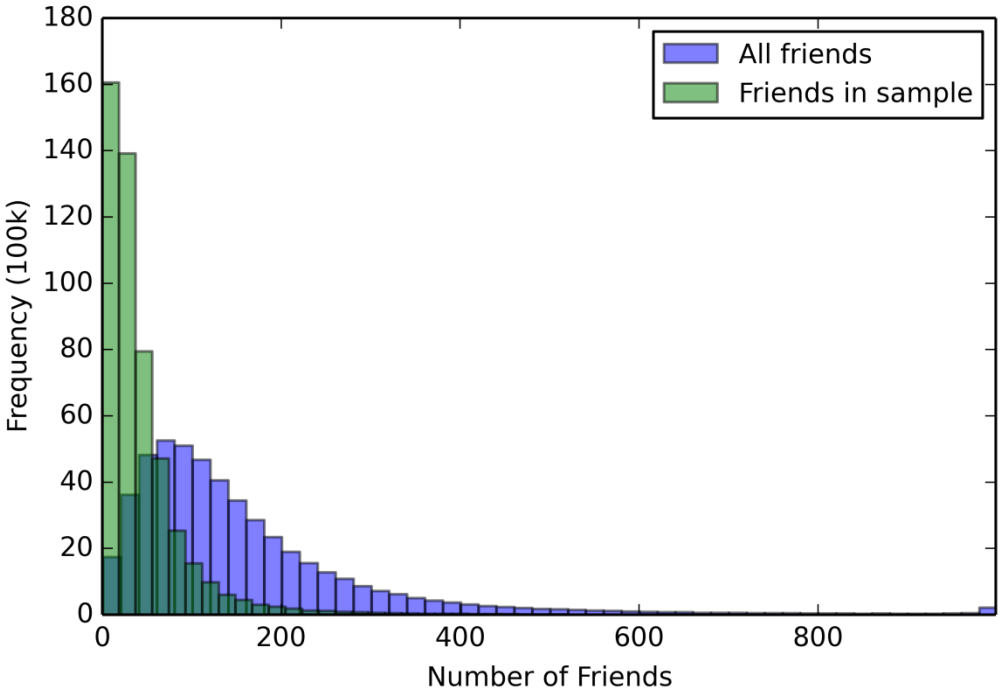


We see from the figure that the performance of this algorithm is very poor compared to the other two methods we considered. Since common check-ins of the friends in our dataset were a lot lower than the ones reported by Foursquare, we investigated the social graph in more detail (see Table 4 and Figure 19).

Table 4: Summary Statistics for Friend Graph

Statistic	Sample	All Friends
Mean	43.67	159.78
Standard Deviation	49.54	146.16
Minimum	0	1
25th Percentile	15	69
Median	30	120
75th Percentile	55	198
Maximum	930	1000

Figure 19: Frequency Distribution of the Number of Friends



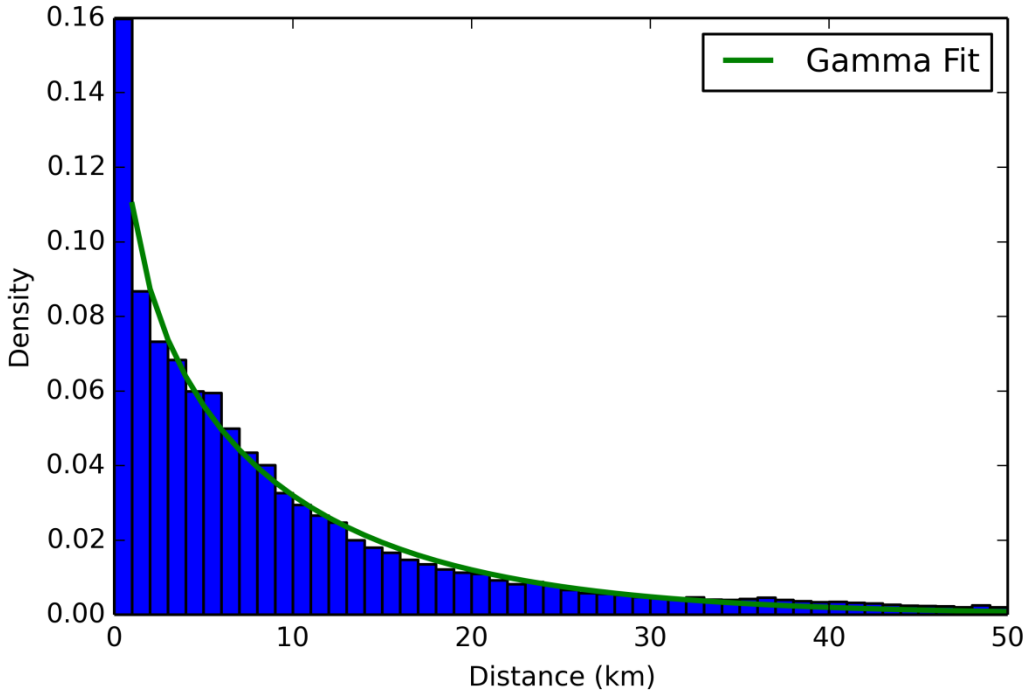
Due to our data collection method, we were only able to extract the check-ins of the users who share their check-ins publicly on another platform, Twitter. Both the table and the figure suggest that there are many other users in our users' social graph that we couldn't account for. The average number of friends of a user is almost four times higher than the ones in our dataset. It is also important to note that Ye et al.'s study (2010) had a much smaller coverage (with around 59 thousand users and 96 thousand venues). Despite of that, since at the time of their study it was possible to extract the check-ins of a user by the user id[3], they were able to construct a more representative social graph.

3.2.5. Location Based Collaborative Filtering

The last method we consider is based on the observation by Ye et al. (2010) that people tend to visit nearby places more often and it is possible to compute item similarity by venue distance. First, we investigate the distribution of the location between the two successive check-ins of users (see Figure 20).

[3] Foursquare now allows retrieving the check-in information only if it is shared publicly.

Figure 20: Distance Distribution of Successive Check-ins



We see in the figure that the distances also follow a right-skewed distribution with very low densities as the distance increases. We truncated the distribution at 50 km as there were jumps after that point caused by the people who visit other cities. Our aim here is to construct a model to recommend places within user's hometown.

The gamma distribution fits well on the data with the shape parameter $k = 0.787$ and a scale parameter $b = 12.06$.

With a strong assumption that these check-ins are independent, the probability that user i checked in to a set of venues L_i can be computed as follows (Cheng et al., 2013):

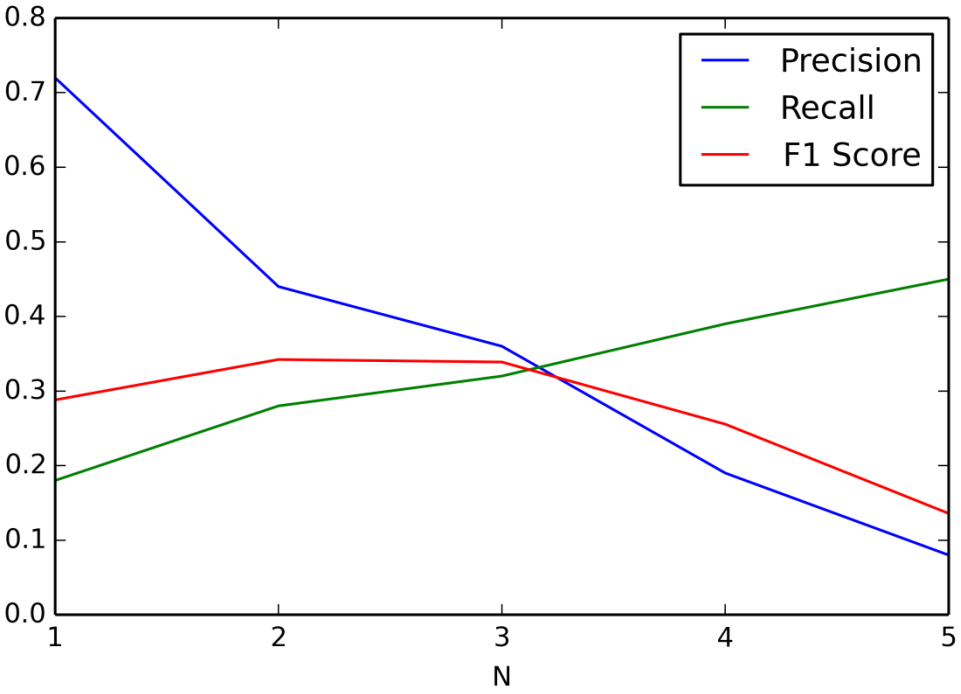
$$P(L_i) = \prod_{u,v \in L_i, m \neq n} P(d(\mathrm{u},\mathrm{v})) \tag{30}$$

Then, the probability that user i visits location v can be computed by means of conditional probabilities:

$$\log P(u \mid L_i) = \log \frac{P(L_i) \prod_{v \in L_i} P(d(u,v))}{P(L_i)} = \sum_{v \in L_i} \log P(d(u,v)) \tag{31}$$

This conditional probability can serve as the check-in score of venue u for user i. The performance metrics of this model can be seen in Figure 21.

Figure 21: Performance Metrics for Location-Based Collaborative Filtering



We see from the figure that, although this model requires an assumption that is hard to justify, it outperforms both item-based collaborative filtering and friend-based collaborative filtering. Also, since it does not take into account the users, it has a substantial improvement on the computational complexity with a competitive performance to user-based collaborative filtering.

3.3. OVERALL EVALUATION

Based on the performance metrics, user-based collaborative filtering outperforms all the other models. Taking computational complexity into account, we can say that user-based collaborative filtering has a very higher computational cost than the other methods. While item-based and location based collaborative filtering methods has poorer performances, they have competitive advantage due to their lower computational complexity.

We present the implications and limitations of the models, along with possible future directions in the conclusion section.

CONCLUSION

In the age of information overload, social data has become a substantial source for researchers and businesses. It is spontaneous. It is not a response to a questionnaire. It is easily accessible. It is trustworthy. It is big.

Today, it is possible to answer the questions that we could not imagine to ask before. It is leading to a future where computers will know us better than our friends. Companies spend more and more money on frameworks where they can store, access and analyze this data. What links do they click? What songs do they skip? Whose picture are they looking at? How much time do they spend on the site? What makes them leave? What makes them stay?

Struggling with these questions, much of their effort goes into personalization. Out of millions of alternatives they search the right ones for you, before you even ask.

At the heart of this race lie the recommender systems. From their creation in early 90's they have been an integral part in many domains: E-commerce, entertainment, service, and content retrieval.

In this study, we evaluated the recommender systems in the location based social networks setting. These networks are relatively new but they attract the attention of many users. Our aim was to provide some insights on user behavior in these networks and then building on those insights to develop a recommender system.

After a thorough introduction on the state-of-the-art of recommender systems with a focus on collaborative techniques, we gave a brief review on the research conducted in location based social networks.

In the main chapter of our study we focused on data analysis and building the recommender system. We analyzed 6.7 million check-ins of 530 thousand users on category, time, check-in frequency, gender and location dimensions. Studying user behavior is crucial for building a successful recommender system. Uncovering hidden patterns and spotting irregularities play an important role for recommender systems. Identifying regular, expected results is also important as they need to be quantified.

Throughout the analysis, we saw that the venue categories, with their interaction to other dimensions, provide useful information about user profiles. Gender and time dimensions clearly show the need for profile-aware recommender systems. In check-in frequency distributions, we confirmed the results from the literature that power law appear in every aspect of a social network. In the location dimension, we concluded that being largely populated does not always mean being popular.

As the user mobility behavior in Turkey is studied for the first time in the recommender system setting, we inferred that the fundamental techniques are a good start and set the scope our study to collaborative filtering techniques.

We built two main models of collaborative filtering: user-based and item-based models. Then, in order to investigate the effect of trust and geolocation, we built two modified models. While we failed to verify the role of trust, we confirmed that geolocation can be as effective as the item similarity in predicting users' check-ins.

User-based collaborative filtering technique outperformed the other techniques. However, the computational cost of this technique is much higher than the other methods. In recommender systems, where real time recommendations are provided to millions of users, computational complexity is as important as the accuracy of the system.

Throughout the study we faced with the problems arising from implicit data. As much as we lay emphasis on the importance of implicit data and how it has changed the directions of e-commerce, we dealt with the lack of power in the techniques utilizing implicit data. While the amount of data is huge, it is harder to interpret and harder to quantify. This leads to difficulties in the evaluation process as well.

In the evaluation of the performances of the models we used two metrics: precision and recall. These metrics have their roots in information retrieval where the problem can easily be formulated as a prediction problem. However, this is not the case for recommender system. A good recommender system is not the one that predicts where user would normally go. A good recommender system is the one that recommends a place that the user is not aware of, that the user interested or excited to

know. Unfortunately, offline evaluation has its limitations in this area. In order to truly measure the success of a recommender system, experimental design can be conducted.

Returning to our discussion on the model that utilizes the geolocation information, we can claim that while it shows good performance, it has no real value in the recommendation systems. What that model basically does is to predict that users usually visit nearby places. Rather than focusing on nearby places that the user probably is aware of, or has already visited, the performance of the recommender system can be measured by users' reactions to unexpected recommendations.

User and item based collaborative filtering techniques, on the other hand, are capable of recommending new places to a certain degree. However, their simplistic structure does not account for most of the readily available information.

In location based social networks, users do not only share the places they visited. They have to-do lists, they explicitly like venues, they give tips about venues, and they comment on each other's activities.

These can be integrated in a content-based recommender system along with venue categories, popular hours of the venues, males and females preferences towards those categories, etc.

Another major limitation of the study is the dataset. It only covers for three months of data and misses the seasonal characteristics of users and venues. Also, the data consists of publicly shared check-ins which introduces a certain bias to the study. First, it requires the use of another platform. Many users do not choose to share this information on a platform where their social circle may contain people they do not know. Twitter's social graph is a little different than Foursquare as it is constructed on "following" instead of "becoming friends". The latter requires mutual agreement.

Overall, this study should be evaluated as a starting point with many possible future directions. Our aim is to improve this study on a larger dataset evaluating the performances of other algorithms in an experimental design setting.

REFERENCES

Adomavicius, G., and Tuzhilin, A. (2005). Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*. *17*(6): 734–749.

Amatriain, X. and Basilico J. (06.04.2012). *Netflix Recommendations: Beyond the 5 Stars.* http://techblog.netflix.com/2012/04/netflix-recommendations-beyond-5-stars.html, (20.07.2014).

Lee, Ben (28.07.2011). *Foursquare's Data and the Explore Recommendation Engine* http://engineering.foursquare.com/2011/08/03/foursquares-data-and-the-explore-recommendation-engine, (20.07.2014).

Burke, R. (2002). Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*. *12*(4): 331–370.

Hofmann, T., & Hartmann, D. (2005). Collaborative Filtering with Privacy via Factor Analysis. *Proceedings of the 2005 ACM Symposium on Applied Computing* (pp. 791-795).

Cheng, C., Yang, H., Lyu, M. R., and King, I. (2013). Where You Like to Go Next: Successive Point-of-interest Recommendation. *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence* (pp. 2605–2611). Beijing, China: AAAI Press. August 3-9, 2013.

Cheng, Y., Fang, Y., and Yuan, Y. (2013). Recommendation System for Location-based Social Network. (Unpublished Technical Report).

Desrosiers, C., and Karypis, G. (2011). A Comprehensive Survey of Neighborhood-based Recommendation Methods. *Recommender Systems Handbook* (pp. 107–144). Springer US.

Ekstrand, M. D., Riedl, J. T., and Konstan, J. A. (2011). Collaborative Filtering Recommender Systems. *Foundations and Trends in Human-Computer Interaction*. *4*(2): 81-173.

Gao, H., and Liu, H. (2014). Data Analysis on Location-Based Social Networks., *Mobile Social Networking* (pp. 165–194). Springer New York.

Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. (1992). Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM*. *35*(12): 61-70.

Goldberg, K., Roeder, T., Gupta, D., and Perkins, C. (2001). Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*. *4*(2): 133-151.

Herlocker, J., Konstan, J. A., and Riedl, J. (2002). An Empirical Analysis of Design Choices in Neighborhood-based Collaborative Filtering Algorithms. *Information Retrieval*. *5*(4): 287-310.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. (2004). Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems (TOIS)*. *22*(1): 5-53.

Jannach, D., Zanker, M., Felfernig, A., and Friedrich, G. (2010). *Recommender Systems: An Introduction*. Cambridge University Press.

Kayaalp, M., Özyer, T., and Özyer, S. T. (2009). A Collaborative and Content Based Event Recommendation System Integrated with Data Collection Scrapers and Services at a Social Networking Site. *Proceedings of the 2009 International Conference on Advances in Social Network Analysis and Mining* (pp. 113–118). Washington, DC, USA: IEEE Computer Society. July 20-22, 2009.

Konstan, J. A., and Ekstrand, M. D. (2013). *Introduction to Recommender Systems*. (Unpublished Lecture Notes). https://www.coursera.org/course/recsys. (20.07.2014).

Liu, J., Dolan, P., and Pedersen, E. R. (2010). Personalized News Recommendation Based on Click Behavior. *Proceedings of the 15th International Conference on Intelligent User Interfaces* (pp. 31-40). Washington, DC, USA: ACM. February 7-10, 2010.

Lops, P., de Gemmis, M., and Semeraro, G. (2011). Content-based Recommender Systems: State of the Art and Trends. *Recommender Systems Handbook* (pp. 73–105). Springer US.

Mangalindan, J. P. (30.07.2012) *Amazon's Recommendation Secret*. http://fortune.com/2012/07/30/amazons-recommendation-secret/, (20.07.2014)

Mobasher, B., Dai, H., Luo, T., and Nakagawa, M. (2001). Effective Personalization Based on Association Rule Discovery from Web Usage Data. *Proceedings of the 3rd International Workshop on Web Information and Data Management* (pp. 9–15). New York, NY, USA: ACM. November 5-10, 2001.

Noulas, A., Scellato, S., Lathia, N., and Mascolo, C. (2012). A Random Walk Around the City: New Venue Recommendation in Location-Based Social Networks. In *Proceedings of the 2012 ASE/IEEE International Conference on Social Computing and 2012 ASE/IEEE International Conference on Privacy, Security, Risk and Trust* (pp. 144–153). Washington, DC, USA: IEEE Computer Society. September 3-6, 2012.

Quercia, D., and Capra, L. (2009). FriendSensing: Recommending Friends Using Mobile Phones. *Proceedings of the Third ACM Conference on Recommender Systems* (pp. 273–276). New York, NY, USA: ACM. October 22-25, 2009.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews. *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work* (pp. 175–186). New York, NY, USA: ACM. October 22-26, 1994.

Resnick, P., and Varian, H. R. (1997). Recommender systems. *Communications of the ACM*. *40*(3): 56–58.

Ricci, F., Rokach, L., and Shapira, B. (2011). Introduction to Recommender Systems Handbook. *Recommender Systems Handbook* (pp. 1–35). Springer US.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). *Application of Dimensionality Reduction in Recommender System-A Case Study* (No. TR-00-043). Minnesota University Minneapolis, Department of Computer Science.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001) *Item-based Collaborative Filtering Recommendation Algorithms*. Proceedings of the 10th International Conference on WorldWideWeb (WWW '01), (pp. 285–295) Hong Kong: ACM. May 1-5, 2001.

Sattari, M., Manguoglu, M., Toroslu, I. H., Symeonidis, P., Senkul, P., and Manolopoulos, Y. (2012). Geo-activity Recommendations by Using Improved Feature Combination. *Proceedings of the 2012 ACM Conference on Ubiquitous Computing* (pp. 996–1003). New York, NY, USA: ACM. September 5-8, 2012.

Scellato, S., Noulas, A., and Mascolo, C. (2011). Exploiting Place Features in Link Prediction on Location-based Social Networks. *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1046–1054). New York, NY, USA: ACM. July 23-26, 2012.

Shardanand, U., and Maes, P. (1995). Social Information Filtering: Algorithms for Automating Word of Mouth. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 210–217). New York, NY, USA: ACM Press/Addison-Wesley Publishing Co. May 7-11, 1995.

Symeonidis, P., Ntempos, D., and Manolopoulos, Y. (2014). *Recommender Systems for Location-based Social Networks*. New York, NY: Springer New York.

Symeonidis, P., Papadimitriou, A., Manolopoulos, Y., Senkul, P., and Toroslu, I. (2011). Geo-social Recommendations Based on Incremental Tensor Reduction and Local Path Traversal. *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks* (pp. 89–96). New York, NY, USA: ACM. November 1, 2011.

Takeuchi, Y., and Sugimoto, M. (2006). CityVoyager: An Outdoor Recommendation System Based on User Location History. *Ubiquitous Intelligence and Computing* (pp. 625–636). Springer Berlin Heidelberg.

Ye, M., Yin, P., Lee, W.-C., and Lee, D.-L. (2011). Exploiting Geographical Influence for Collaborative Point-of-interest Recommendation. *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 325–334). New York, NY, USA: ACM. July 24-28, 2011.

Yoon, H., Zheng, Y., Xie, X., and Woo, W. (2010). Smart Itinerary Recommendation Based on User-Generated GPS Trajectories. *Ubiquitous Intelligence and Computing* (pp. 19–34). Springer Berlin Heidelberg.

Zheng, V. W., Zheng, Y., Xie, X., and Yang, Q. (2010). Collaborative Location and Activity Recommendations with GPS History Data. *Proceedings of the 19th International Conference on World Wide Web* (pp. 1029–1038). New York, NY, USA: ACM. April 26-30, 2010.

APPENDICES

APPENDIX 1: Sample Tweet Object in JSON Format


```
{
   "text": "I'm at Gözde Sitesi http://t.co/pXc3bHzMJN",
   "id": xxx,
   "favorite_count": 0,
   "source": "<a href=\"http://foursquare.com\" rel=\"nofollow\">foursquare</a>",
   "retweeted": false,
   "entities": {
      "user_mentions": [],
      "symbols": [],
      "trends": [],
      "hashtags": [],
      "urls": [
         {
            "url": "http://t.co/pXxc3bHzMJN",
            "expanded_url": "http://4sq.com/1pKx9k6a",
            "display_url": "4sq.com/1pKx9k6a"
         }
      ]
   },
   "retweet_count": 0,
   "favorited": false,
   "user": {
      "id": xxx,
      "followers_count": 40,
      "statuses_count": 1143,
      "description": "★Trance/Electronic/HipHop★  /Fenerbahçe/",
      "friends_count": 71,
      "location": "ANKARA",
      "name": "Serkan",
      "lang": "tr",
      "favourites_count": 125,
      "screen_name": "xxx",
      "url": "http://instagram.com/xxx",
      "created_at": "Sun Feb 02 13:48:32 +0000 2014",
   },
   "lang": "tr",
   "created_at": "Thu Jul 24 09:13:19 +0000 2014",
   }
```

APPENDIX 2: Sample Check-in Object in JSON Format


```
{
   "twId": "xxx",
   "checkin": {
      "likes": {
         "count": 1,
         "groups": [
            {
               "count": 1,
               "items": [
                  {
                     "lastName": "xxx",
                     "photo": {
                        "prefix": "https://irs3.4sqi.net/img/user/",
                        "suffix": "/83657760-5RBLJYW2WMUNTJWT.jpg"
                     },
                     "id": "xxx",
                     "firstName": "xxx",
                     "gender": "male"
                  }
               ],
               "type": "others"
            }
         ],
         "summary": "xxx xxx"
      },
      "like": false,
      "isMayor": true,
      "reasonCannotAddComments": "notfriends",
      "venue": {
         "verified": false,
         "name": "xs home",
         "specials": {
            "count": 0
         },
         "contact": {},
         "location": {
            "lat": 39.91398698853621,
            "cc": "TR",
            "lng": 32.8930440805031,
            "isFuzzed": true,
            "country": "Türkiye"
         },
         "stats": {
            "tipCount": 0,
            "checkinsCount": 28,
            "usersCount": 6
         },
         "id": "xxx",
         "categories": [
            {
               "pluralName": "Homes (private)",
               "primary": true,
               "name": "Home (private)",
               "shortName": "Home",
               "id": "4bf58dd8d48988d103941735",
```

```
            "icon": {
               "prefix": "https://ss1.4sqi.net/img/categories_v2/building/home_",
               "suffix": ".png"
            }
         }
      ]
   },
   "photos": {
      "count": 0,
      "items": []
   },
   "source": {
      "url": "https://foursquare.com/download/#/android",
      "name": "foursquare for Android"
   },
   "shout": "Kilo almak için güzel aksam yemekleri.",
   "score": {
      "total": 7,
      "scores": [
         {
            "message": "First check-in at x's home.",
            "points": 5,
            "icon": "https://ss1.4sqi.net/img/points/swarm-discoveryvenue.png"
         },
         {
            "message": "First of friends to check in at x's home.",
            "points": 2,
            "icon": "https://ss1.4sqi.net/img/points/swarm-discoveryvenue.png"
         }
      ]
   },
   "createdAt": 1397668690,
   "reasonCannotSeeComments": "notfriends",
   "type": "checkin",
   "id": "534ebb52498e2f287dc225f1",
   "timeZoneOffset": 180,
   "user": {
      "lastName": "xxx",
      "photo": {
         "prefix": "https://irs2.4sqi.net/img/user/",
         "suffix": "/72345075-YQYCMMHN2W1WRHWA.jpg"
      },
      "id": "xxx",
      "firstName": "xxx",
      "gender": "xxx"
   }
  }
}
```

APPENDIX 3: Geographical Heatmap of the Check-ins