Latent Feature Models for Uncovering Human Mobility Patterns from Anonymized User Location Traces with Metadata

Dissertation by

Basma Mohammed Alharbi

In Partial Fulfillment of the Requirements

For the Degree of

Doctor of Philosophy

King Abdullah University of Science and Technology
Thuwal, Kingdom of Saudi Arabia

March, 2017
The dissertation of Basma Alharbi is approved by the examination committee

Committee Chairperson: Xiangliang Zhang
Committee Members: Hui Xiong - Professor at Rutgers University, Mikhail Moshkov - Professor at KAUST, Xin Gao - Associate Professor at KAUST
ABSTRACT

Latent Feature Models for Uncovering Human Mobility Patterns from Anonymized User Location Traces with Metadata

Basma Mohammed Alharbi

In the mobile era, data capturing individuals’ locations have become unprecedently available. Data from Location-Based Social Networks is one example of large-scale user-location data. Such data provide a valuable source for understanding patterns governing human mobility, and thus enable a wide range of research. However, mining and utilizing raw user-location data is a challenging task. This is mainly due to the sparsity of data (at the user level), the imbalance of data with power-law users and locations check-ins degree (at the global level), and more importantly the lack of a uniform low-dimensional feature space describing users.

Three latent feature models are proposed in this dissertation. Each proposed model takes as an input a collection of user-location check-ins, and outputs a new representation space for users and locations respectively. To avoid invading users privacy, the proposed models are designed to learn from anonymized location data where only IDs - not geophysical positioning or category - of locations are utilized. To enrich the inferred mobility patterns, the proposed models incorporate metadata, often associated with user-location data, into the inference process.

In this dissertation, two types of metadata are utilized to enrich the inferred patterns, timestamps and social ties. Time adds context to the inferred patterns, while social ties amplifies incomplete user-location check-ins. The first proposed model incorporates timestamps by learning from collections of users’ locations sharing the same discretized time. The second proposed model also incorporates time into the
learning model, yet takes a further step by considering time at different scales (hour of a day, day of a week, month, and so on). This change in modeling time allows for capturing meaningful patterns over different times scales. The last proposed model incorporates social ties into the learning process to compensate for inactive users who contribute a large volume of incomplete user-location check-ins. To assess the quality of the new representation spaces for each model, evaluation is done using an external application, social link prediction, in addition to case studies and analysis of inferred patterns. Each proposed model is compared to baseline models, where results show significant improvements.
ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and appreciation to the people who helped me accomplish my PhD degree. This chapter of my life would not have been completed without the presence and the inspiration I continuously receive from each member of my family. In particular, to my parents, Suzan Shams and Mohammed Alharbi, I will forever be in your debt for the support, guidance and sacrifices you made for me. I will forever be grateful to my sisters, Reem and Rasha, for their continuous emotional support and uplifting during my journey. To my brothers, Badr, Sari and Bahar, without them I would not have become the person I am now. I stand speechless in front of the sacrifices, dedication and support endowed by my other half, Alaa Alharbi. Last but not least, to my son, Mohammed, you will forever be my number one motivation.

I am filled with gratitude to my professor, Dr. Xiangliang Zhang, who paved the way for my learning process and made my PhD joinery enriching and fulfilling. I am grateful for her patience, guidance and endless support. I will forever be in debt to Professor Mikhail Moshkov for his continuous support during my journey at KAUST. I sincerely thank my committee members, Professor Hui Xiong, Professor Mikhail Moshkov, and Professor Xin Gao for their valuable comments and feedback on this thesis. I am thankful to my fellow students, members of the MINE group and my friends at KAUST.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examination Committee Page</td>
<td>2</td>
</tr>
<tr>
<td>Copyright</td>
<td>3</td>
</tr>
<tr>
<td>Abstract</td>
<td>4</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>6</td>
</tr>
<tr>
<td>List of Figures</td>
<td>9</td>
</tr>
<tr>
<td>List of Tables</td>
<td>11</td>
</tr>
</tbody>
</table>

## 1 Introduction

- 1.1 Attributes of User-Location Data | 17 |
- 1.2 Feature Extraction | 21 |
- 1.3 Mixed Membership Models | 23 |
- 1.4 Contributions and Thesis Organization | 27 |

## 2 Background

- 2.1 Preliminary | 31 |
  - 2.1.1 Notations and Problem Definition | 31 |
  - 2.1.2 Social Link Prediction | 37 |
- 2.2 Related Work | 46 |
  - 2.2.1 on Inferring Human Mobility Patterns | 46 |
  - 2.2.2 on Link Prediction with Mobility Patterns | 50 |

## 3 Baseline Studies

- 3.1 Using Feature Engineering | 52 |
- 3.2 Using Latent Feature Models | 55 |
  - 3.2.1 Principal Component Analysis (PCA) | 55 |
  - 3.2.2 Latent Dirichlet Allocation (LDA) | 56 |
  - 3.2.3 Biterm Topic Model (BTM) | 60 |
  - 3.2.4 Compact Representation Model (CRM) | 64 |
3.3 Evaluation ........................................................................ 70
3.4 Discussion ........................................................................ 78

4 Model I: Incorporating Timestamps .................................. 80
  4.1 The Proposed Model ...................................................... 81
  4.2 Estimating Model Parameters ........................................ 86
    4.2.1 Estimating the Mixture Assignments ....................... 86
    4.2.2 Computing the New Representations ...................... 88
  4.3 Evaluation ...................................................................... 89
  4.4 Discussion ...................................................................... 94

5 Model II: Incorporating Multiple Time-Scales .................. 96
  5.1 The Proposed Model ...................................................... 97
  5.2 Estimating Model Parameters ........................................ 101
    5.2.1 Estimating the Mixture Assignments ....................... 102
    5.2.2 Optimizing the Hyperparameters ............................ 106
    5.2.3 Computing the New Representations ...................... 108
  5.3 Evaluation ...................................................................... 109
  5.4 Discussion ...................................................................... 114

6 Model III: Incorporating Social Ties ............................... 115
  6.1 Motivation ..................................................................... 116
  6.2 The Proposed Model ...................................................... 117
  6.3 Estimating Model Parameters ........................................ 120
  6.4 Evaluation ..................................................................... 123
  6.5 Discussion ..................................................................... 128

7 Concluding Remarks ......................................................... 129
  7.1 Summary of Contributions ............................................ 129
  7.2 Future Research Directions .......................................... 131

References ............................................................................ 148

Appendices ............................................................................ 149
LIST OF FIGURES

1.1 An Illustrative Example of User-Location Data .................................. 13
1.2 An Illustrative Example of User-Location Check-ins ............................. 14
1.3 An Illustrative Example of User-Location Matrix ................................. 15
1.4 An Illustrative Example of User-Feature Matrix ................................. 17
1.5 Data Distributions: Users’ Check-ins Distributions ............................ 18
1.6 Data Distributions: Locations’ Check-ins Distributions ........................ 20
1.7 An Illustration of the Feature Engineering Process ............................. 22
1.8 An Illustrative Example of Mixed-Membership Models ........................ 24

2.1 A Sample of User Location Check-ins .............................................. 32
2.2 A Sample of User Location Matrix .................................................. 34
2.3 A Sample of User Adjacency Matrix .............................................. 35
2.4 An Illustration of Social Link Prediction .......................................... 38
2.5 An Illustration of the ROC curve and AUC metric .............................. 40
2.6 Distributions of Basic Network Measures ......................................... 41

3.1 Baseline Studies: Graphical Representation of LDA ............................ 57
3.2 Baseline Studies: Parameters of the Dirichlet Distribution ................... 61
3.3 Baseline Studies: Graphical Representations of BTM and CRM .......... 62
3.4 Baseline Studies: Graphical Representations of CRM ........................ 64
3.5 Baseline Studies: Illustration of CRM ............................................. 67
3.6 Baseline Studies: Visualization of the Learned Representations ............ 74
3.7 Baseline Studies: Evaluation Results on Social Link Prediction ............ 76
3.8 Baseline Studies: Visualization of Link Prediction Results .................. 77

4.1 Model I: Graphical Representation ............................................... 84
4.2 Model I: Visualization of the Learned Representations ...................... 91
4.3 Model I: Sensitivity Analysis ....................................................... 94

5.1 Model II: Graphical Representation ............................................... 98
5.2 Model II: Visualization of the Inferred Patterns ................................ 111
5.3 Model II: Word Cloud of Popular Locations .................................... 112
5.4 Model II: Run Time Analysis ........................................ 112
5.5 Model II: Sensitivity Analysis ................................. 113

6.1 Model III: Analysis of Link and Mobility ...................... 117
6.2 Model III: Graphical Representation ......................... 118
6.3 Model III: Visualization of the Learned Representations ... 125
6.4 Model III: Case Study ........................................ 126
6.5 Model III: Sensitivity Analysis ............................... 127
LIST OF TABLES

2.1 Sample of User-Location Check-in Records from two datasets. . . . . . 31
2.2 Summary of Related Work . . . . . . . . . . . . . . . . . . . . . . . 47
3.1 Baseline Studies: Datasets Statistics . . . . . . . . . . . . . . . . 71
3.2 Baseline Studies: Evaluation Results on Social Link Prediction . . . . 75
4.1 Model I: Datasets Statistics . . . . . . . . . . . . . . . . . . . . . . 90
4.2 Model I: Evaluation Results on Social Link Prediction . . . . . . . . . 92
5.1 Model II: Datasets Statistics . . . . . . . . . . . . . . . . . . . . . 110
5.2 Model II: List of Top Locations Associated with Inferred Patterns . . 111
6.1 Model III: Datasets Statistics . . . . . . . . . . . . . . . . . . . . . 123
6.2 Model III: Evaluation Measures on Social Link Prediction . . . . . . . 127
Chapter 1

Introduction

Background. Given the recent technological advances and the increasing popularity of GPS-enabled devices, many activities now leave digital traces, and many of these traces capture individuals locations. In the digital world for example, people intentionally share their locations when they post content on social media, such as when posting images of themselves visiting new restaurants on Instagram with location tags. Collectively, these location-based posts provide large-scale data capturing users’ locations. Additionally, locations of users are automatically collected by service providers such as mobile network operators, e.g., when a call is made, and credit card providers when an item is bought, for billing purposes. Again, collectively, these recorded transactions provide user-location data at an unprecedented scale. Such user location data capture some of the locations that a user visits over a long period of time, and for a large portion of the population.

For one user, the collection of his records, along with metadata, can reveal a lot about him, including his daily (regular) movements patterns and irregular ones. More interestingly, the collection of records of all users in a dataset facilitate understand-

\footnote{In trajectory data, the complete path of a moving object (e.g., user) is captured. Locations along the path are thus continuous and not discrete values. User-location data, on the other hand, only capture users’ locations at certain times of a day (i.e., when a transaction or a post is recorded), often recording location IDs which allows treating observed locations as a discrete variable. Though user-location data only records few locations per a user, it covers a larger portion of the population. Figure 1.1 illustrates the difference between both datasources, where the blue path denotes what trajectory data look like and the labeled points represent user-location records. This key difference between the two datasources prevents utilizing popular trajectory data mining techniques here. Hence, trajectory data mining \[1\] - with its popular research areas including trajectory analysis \[2, 3\], representation \[4, 5\], indexing \[6, 7\], matching \[8\], clustering \[9\], recommendation \[10, 11\] and travel mode detection \[12, 13\] - is out of the scope of this dissertation.}
ings of human mobility [14, 15], and has proven to be beneficial in varying areas, ranging from sociology to urban computing, and from disease control to customized services and targeted advertisement, where *user profiling* [16] and *location profiling* are essential. Examples of interesting applications include Point-of-Interest recommendation [17, 18, 19, 20, 21, 22] and characterization [23], public transportation planning [24, 25], travel prediction [26], job change analysis [27], land use and region function prediction [28, 29], route clustering and prediction [30], demographic inference [31, 32], urban pattern extraction [33, 34], emergency detection [35], location advertisement [36], and community detection [37].

![Image of user location data](image.png)

**Figure 1.1:** An example of a user location data from the Reality Mining dataset [38]

**User-Location Data.** To understand what user-location data consist of, consider as an example data automatically logged by mobile network operators, formally known as Call-Detail Records. In this data, and for every user, a record is automatically created every time a user initiates/receives a phone call, message or even connects to the internet. This record, captures among others, the user ID, location, and other details required for billing purposes. Figure 1.1 visualizes five records cap-
Figure 1.2: A toy example of user location check-ins.

<table>
<thead>
<tr>
<th>Time of Day</th>
<th>Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 am</td>
<td>![Image of a person]</td>
</tr>
<tr>
<td>8 am</td>
<td>![Image of a school]</td>
</tr>
<tr>
<td></td>
<td>![Image of a restaurant]</td>
</tr>
<tr>
<td>5 pm</td>
<td>![Image of a gym]</td>
</tr>
<tr>
<td></td>
<td>![Image of a home]</td>
</tr>
<tr>
<td>7 pm</td>
<td>![Image of a store]</td>
</tr>
</tbody>
</table>

In addition to the user ID and location data, time is often recorded as a metadata in user location data, and is very valuable to such data as it provides context and semantics to the otherwise raw collections of users records. Additionally, depending on the data source, other metadata are often recorded as well, and they often provide valuable details and insights about the recorded data. Metadata includes, among others, 1) location specific metadata such as the geographical positioning of the location (latitude and longitude), the name and category of the location, and 2)
user specific metadata such as user demographic details (age, gender, nationality) and user’s friends (links/ties in the social network). These metadata enrich user-location data and allow for deeper understanding of human movement.

![Locations](image)

Figure 1.3: A toy example of user location data represented as vector space model.

**Problem Statement.** User-location data is often represented as a vector space model; a high dimensional matrix of user location check-ins counts over a given time period. Figure 1.3 illustrates this concept using a toy example, with five users visiting a total of six locations over a period of one week. In the Figure, each entry in the $5 \times 6$ matrix denotes the number of times a user visited a location in a week. This simple representation ignores the timestamp of check-ins and only accounts for the number of times a user visited each location in the entire dataset. Additionally, it does not incorporate any of the useful metadata often associated with user-location data, such as social links for example.

More importantly, in real datasets, the number of users and locations is large, making the user-location matrix a high-dimensional one. This high-dimensional matrix is sparse, with less than 0.003% non-zero entries in two real datasets. Both the
high-dimensionality and sparsity of the matrix indicate that such representation cannot be used directly by machine learning methods, given that the performance of the later is heavily dependent on the feature space of the data. This is especially true when the task requires computing user (or location) distance measures, as distance measures computed in a high-dimensional space suffer from the curse of dimensionality [39, 40]. This is because, in a high-dimensional and sparse space, the distance between an object (e.g., a user) and his nearest neighbor approaches the distance to his furthest neighbor [41]. Consequently, this reduces the usability of the measure (e.g., Euclidean distance) for discriminating between near and far neighbors [42].

Given the above discussion, the high-level objective of this dissertation is to represent users in user-location data by a low-dimensional feature space while utilizing metadata to enrich the new feature space. That is, the objective is to infer a set of representative features that can group collections of locations together, as well as describe users using the inferred feature space. Figure 1.4 illustrates this concept, showing the users’ new feature space, illustrated in Subfigure 1.4(a), where users are described by two features in a $5 \times 2$ matrix. Additionally, features themselves are groupings of locations, as illustrated in Subfigure 1.4(b) and are described by a $2 \times 6$ matrix. The new low-dimensional feature space for users enable utilizing user-location data in diverse applications, and facilitates exploiting existing machine learning methods, where the outcome is highly dependent on the quality of the extracted feature space. This being said, the main contribution in this dissertation lies in proposing three models for feature inference designed for user-location data (a detailed list of contribution is provided in Section 1.4).

Inferring a new feature space is formally known as feature extraction where many methods exist, including feature engineering and feature learning (i.e., representation learning [43]). Different methods group locations based on different assumptions, and thus yield different feature sets. A general overview of feature extraction methods
Figure 1.4: A toy example of user location data represented in the new feature space.

is provided in Section 1.2 followed by an overview of mixed-membership models, a powerful modeling methodology, in Section 1.3 as they form the backbone of feature learning models proposed in this dissertation. First, however, an analysis of two real user-location data is provided next to give the reader a comprehensive overview of the attributes of such data.

### 1.1 Attributes of User-Location Data

Two real datasets, GoWalla (GW) and Brightkite (BK) collected from Location-Based Social Networks are analyzed, with emphasis on understanding the distributions of both user and location check-ins. Precisely, the distributions of four measures are described and analyzed next to provide a comprehensive view of user-location data. Let \( v \) be a user, \( p \) be a location, \( |V| \) be the number of users and \( |P| \) be the number of locations, and \( X \) be a \( |V| \times |P| \) matrix of user-location check-in counts where the value in \( x_{v,p} \) denotes the number of times user \( v \) visited location \( p \) over the

\[\text{Described in Chapter 6}\]
data collection period.

![User Check-Ins](image1.png)

![User Venues](image2.png)

Figure 1.5: Distributions of users check-ins and venues.

1. User Check-In Distribution quantifies the total number of check-ins per a user, i.e., user check-in degree. User $v$’s check-in degree is given by the summation of his vector of location check-ins ($\sum_p x_{v,p}$). All users check-in degrees are given by: $X1^T$, where $1$ is a $1 \times |P|$ row vector of 1’s. This measure is a rough indication of users activity level, where users in a community have varying activity levels. The user check-in degree distribution is shown in Subfigures 1.5(a) and 1.5(c) for GW and BK datasets, respectively. Each plot is comprised of three elements: 1) the Probability Distribution Function (PDF) of the measure, i.e., user check-ins, in light blue 2) the distribution after logarithmic binning [45] to reduce the noise in the tail of the distribution, in red, and 3) the fitted power law distribution in black. The log-binned
distributions in both plots fit a truncated power law, with exponent $e = 2.11 (1.36)$, and $\lambda = 6.6e-03 (6.2e-03)$, in GW (BK). This reveals that most users have a low number of check-ins while few have large number of check-ins. More specifically, 16% and 30% (22% and 35%) of users in GW (BK) have trajectories shorter than or equal to 3 and 5, respectively.

2. User Venue Distribution quantifies the total number of locations visited by a user, regardless of the frequency of visits. In other words, user venues indicate the number of unique locations visited by the user. The user venue check-in distribution is shown in Subfigures 1.5(b) and 1.5(d). Similarly to the above measure, both distributions here fit a truncated power law distribution, with exponent $e = 2.44$ and $1.34$, and $\lambda = 4e-04$ and $4e-03$, in GW and BK, respectively. This is an indication that most users visit few locations, and the number of users decreases exponentially as the number of visited locations increases.

3. Location Check-In Distribution quantifies the total number of check-ins per a location, i.e., location check-in degree. Location $p$’s check-in degree is given by the summation of its vector of users check-ins ($\sum_v x_{v,p}$). All locations check-in degrees are given by: $\mathbf{1} \mathbf{X}$, where $\mathbf{1}$ is a $1 \times |\mathcal{V}|$ row vector of 1’s. The location check-in degree distribution is shown in Subfigures 1.6(a) and 1.6(c) for GW and BK datasets, respectively. Both distributions fit a truncated power law distribution with exponent $e = 2.79$ and $1.96$ and $\lambda = 6.6e-05$ and $7.5e-05$ in GW and BK, respectively. The distributions show that 39% (55%) of locations have only 1 check-in in GW (BK) while 58% (74%) of locations have maximum of 2 check-ins in GW (BK).

4. Location Visitors Distribution quantifies the total number of users visiting a location, regardless of the frequency of visits by the same users. In other words, location visitors indicate the number of unique users visiting the location. The location unique check-in distribution is shown in Subfigures 1.6(b) and 1.6(d) for GW and BK, respectively. Both distributions fit a truncated power law with exponent $e = 2.58$
and 2.49, and $\lambda = 3e - 4$ and $1e - 04$ in GW and BK, respectively. Similarly to the previous distributions, it is shown that the popularity of locations varies significantly across the collected dataset.

The distributions above provide a general overview of the characteristics of user-location data. They mainly highlight the diversity in users and locations check-in degree, which form challenges to popular feature extraction methods, as will be discussed throughout the dissertation. Next, an overview of general feature extraction methods is provided, with focus on comparing two general categories: feature engineering and feature learning.
1.2 Feature Extraction

Feature engineering is a mainstream approach in feature extraction, where a shared set of features are designed by experts and extracted from data to achieve a given task. Figure 1.7 illustrates the steps often undertaken in feature engineering. For instance, if one is interested in inferring individuals age-group from their user-location data, a specifically designed set of features will be extracted from their trajectories in which it is known/assumed that these features will be useful for this task. For example, the extracted features can include the type of locations visited by the user (work, school, home, ...), the duration of visited locations (i.e., how much time is spent at school, work, home, ...), frequency of visited locations and so on. More often than not, this practice results in a large set of features (where some are redundant, correlated, or irrelevant) which -as a result - requires applying a dimensionality reduction technique or feature selection on top of feature engineering in order to find a reduced feature space. This is followed by learning the required task given the extracted features. Then, if the feature set achieves acceptable results in the required task, the process of feature engineering ends. If not, the entire process of feature extraction, dimensionality reduction (if necessary), then application and evaluation are repeated until satisfactory results is obtained.

Feature engineering is popular due to the superior results it often provides. However, despite its evident advantage, minimizing human involvement is always preferable in knowledge execution [46]. Additionally, feature engineering often ends up ignoring hidden patterns which are important to the task under study but not obvious to the experts. More importantly, it is based on predefined knowledge of the data and the specific task under study, which restricts the use of extracted features to other applications.

Consequently, to minimize the human effort in feature engineering, feature learning using latent variable models is preferred, where features in such models are re-
ferred to as latent patterns. Based on the manifold hypothesis [43], observed high-dimensional data can be better represented by a much lower dimensional space, which captures the most important characteristics of the data, without the disadvantages associated with high dimension (i.e., the curse of dimensionality). This hypothesis is valid for many datasets, including data of human mobility as it is regular and predictable [47], which indicate that there are latent factors, i.e., mobility patterns, governing how and why people move. This new feature space, i.e., latent patterns (as illustrated in Figure 1.4), can be used as an alternative to the observed check-in counts (as illustrated in Figure 1.3), which is better suited for a wide range of tasks.

Feature learning in this context refers to the unsupervised process of extracting latent features from raw data. The extracted features are not task-specific, and they capture the intrinsic dimensionality of the original high dimensional feature space. Latent variable models is a general term used to describe a wide range of models for inferring latent patterns, where instances are described by inferred patterns instead of the original feature space. Indeed, many models fall under this general term, with a shared objective of learning new and representative feature space from the observed data.
raw data. Latent variables models are applied to diverse applications, ranging from recruitment analysis [48] to real estate ranking [49, 50], and from functional zone discovery [51] to travel prediction [26] and collective anomaly detection [52].

It is important to note here that learning latent patterns from user-location is not a trivial task, despite the fact that a large portion of human mobility is regular. This is because trajectories are generated by complex processes that are often hard to quantify, especially when the number of samples (check-ins) in the data are not sufficient. Not to mention the huge variation in user and location check-in degrees, as discussed in the previous section, which adds another layer of challenges.

The main difference between the two approaches (feature engineering vs. learning) is in the generalizability and automation of the approach, where the later minimizes the amount of intervention and preprocessing required by data miners and domain experts. For this reason, the models proposed in this dissertation are instances of feature learning, specifically the proposed models are instances of mixed-membership models. Thus, an overview of mixed-membership models, a powerful modeling framework, is provided next. Note that although the proposed models fall under feature learning, we examined the feature engineering approach to experimentally compare the validity of the two different approaches in Chapter 3.

1.3 Mixed Membership Models

Mixed-membership models [53] are a powerful modeling methodology, which assume that the observed data (i.e., check-ins) is composed of a collection of statistically related groups (e.g., users), where each group comprises a set of observations (e.g., user check-ins). Further, instances of this model assume that 1) observed groups (e.g., users) lie in an unobserved low-dimensional space of latent features (a.k.a., latent patterns), and (2) each group is affiliated with multiple latent features (hence, the mixed-membership name). In other words, it assumes that there exists a set
of mixture components global to the entire collection of data, which denotes the latent patterns (e.g., mobility patterns) shared by all groups. Additionally, each group of observed data has a unique mixture proportions, drawn from the global components, and local to each group. That is, the observed collection of data has global underlying patterns, yet each group in the collection exhibits those patterns with different proportions.

Figure 1.8: A toy example of Mixed-Membership Models

To clarify the assumptions detailed above, consider a simple example of user-location data, with 5 users and 6 locations, illustrated in Figure 1.8. In this example, it is assumed that user-location data is a collection of statistically related users, where each user is described by a collection of his location check-ins (as illustrated in Figure 1.2). Consequently, this indicates that there exists a set of global mixture components\(^3\) shared by all users; 2 in this example and are visualized by the blue and green circles in the middle of the Figure, denoted as patterns. Each global mixture components and latent patterns are used interchangeably in this dissertation.
component has a unique distribution over locations showing the locations that are highly associated with it, as illustrated by the right-most bar charts. In this case, the first component is highly associated with university and gym locations while the second one is more about school and playgrounds. Local mixture proportions, on the other hand, are illustrated by the left-most bar charts. Each bar chart shows the belongingness of a user to each global mixture component. This assumption allows users to ‘belong’ to each component, with varying probability just like in a fuzzy clustering setting, and thus justifies the mixed-membership name. The first user for example is highly associated with the first component, which indicates that he frequently visits university, while the second user is equally described by components one and two which indicates equal probability of visiting either locations associated with each component.

This simple illustration discussed above shows the powerful modeling capability of mixed-membership models as it enables summarizing raw data by latent patterns inferred solely from raw user-location data. Although other latent variable models in general share the same objective, when compared to mixed-membership models the later (1) provides an intuitive and flexible framework for designing latent feature models, (2) enables incorporating different components and assumptions into the model, (3) enables learning latent features efficiently from the observed data, given the vast amount of research on fast inference techniques for mixed-membership models, and (4) provides interpretable representations and this is a fundamental criteria in many real world applications.

For example, Non-Negative Matrix Factorization with constraints [54, 55, 56] can be considered as an intuitive solution to this problem. However, due to the power-law distribution of user and location check-ins, the input is extremely sparse (with about \( \leq 0.003\% \) non-zeros in two real datasets). Decomposition without considering the key attributes of the data, such as the distributions of users and locations, will result
in new representation that lives in a space with latent concepts dominated by few frequent users and locations. Additionally, the computational cost for decomposing a large matrix is high. As shown in Chapter 6, the number of locations is more than one million or half million, while the number of users is hundreds of thousands in two real datasets. More importantly, the new representations inferred by mixed-membership models are easily interpretable, which enables understanding the patterns governing human mobility, unlike those inferred by matrix factorization.

Learning representations of human mobility using mixed-membership models is a relatively new area of research. Yet, mixed membership models are heavily utilized in text mining, and are formally known as topic models. Consequently, most work in human mobility latent feature representation tends to adopt existing off-the-shelf topic models (e.g., Latent Dirichlet Allocation (LDA)) to human mobility data [3, 34]. Due to the fact that data of human mobility and text corpora share many statistical similarities, adopting topic models to human mobility data is justifiable, and has proven to be successful. However, despite this success, existing topic models have assumptions (and limitations) that do not correlate well with observed mobility data. For example, text corpora often contain documents of roughly equal lengths, while user-location data contains users with a power-law check-in distribution as discussed earlier. Additionally, in text mining using LDA, it is assumed that order of words is not important, yet order (actually time) of visited locations provide important context and semantics to user-location data. These differences compel the need to design latent feature models for user-location data.
1.4 Contributions and Thesis Organization

The main objective in this dissertation is to learn latent patterns\footnote{Latent patterns, global mixture components, feature space and representation space are used interchangeably in this dissertation to indicate the same concept.} governing human mobility from raw user-location data and use them as a new representation space to enable utilizing this datasource in diverse applications. The list of contributions in this work is provided below, followed by dissertation outline.

Evaluation framework. Proper evaluation of the inferred latent patterns is very important as it enables assessing the quality of the inferred patterns and thus facilitates comparing different models together. Popular evaluation measures for mixed membership models include computing the log likelihood of held-out data and qualitatively assessing the interpretability of the inferred patterns. The former approach measures the predictive power of a given model, rather than the quality of the inferred patterns, which is of most importance \cite{57}. The second approach, qualitative assessment of the inferred patterns, is often done by eyeballing and is subject to wishful thinking \cite{57}. It is thus necessary to propose a framework that enables quantitative evaluation of the inferred patterns from different latent variable models.

In this dissertation, an evaluation framework is proposed where an external task, social link prediction, is used to quantitively evaluate the inferred user feature space (illustrated in Figure 1.4(a)). That is, the inferred user feature space is used to infer new link formation in the social graph, and performance of the new feature space on this task is used as an indication of the quality of the inferred patterns. In addition to evaluating the inferred user feature space using an external task, several analysis tasks and case studies are conducted when possible to verify the semantics behind the inferred mobility patterns (which provide grouping of locations, as illustrated in Figure 1.4(b)).
Baseline studies. Social link prediction is used to experimentally evaluate baseline methods, where comparison of feature engineering and feature learning is conducted. The objective here is to justify adopting latent variable models over feature engineering. The results of the experimental evaluation on two real datasets highlighted the superiority (and sometime equivalence) of the autamative process of latent variable models when compared to the expert-based feature engineering. The findings here lead to the adoption of the general framework of mixed-membership models as well as highlighted the importance of designing mixed-membership models specifically tailored to handle the attributes (and incorporate the metadata) of user-location data.

Proposed latent feature models. Three latent feature models are proposed in this dissertation, all sharing the same objective of inferring latent mobility patterns from raw user-location data. The problem of designing latent feature models for human mobility data can be viewed as an iterative process, where each model is designed to handle a specific attribute found in the data, then tested and evaluated on real data, and later criticized for potential improvements. All proposed models learn from anonymized low-level location data, without the need for the actual geographical positioning of the location nor its name or category. This minimizes the time-consuming (and error-prone) preprocessing step involved in semantic labeling as well as adds a layer of privacy to the inferred patterns. An overview of each proposed model is described next.

Model 1 As shown in Baseline studies variation in users check-in degree instills challenges for a generic mixed-membership model. The first proposed model, in Chapter 4, targets users with high check-in degree, those represented by the tail of the distribution, as they cause a key challenge to mixed membership models. Chapter 4 focuses on designing a model
that targets this issue simply because location traces for such users are rich with latent patterns which should not be ignored. This is achieved by introducing the concept of sequences which enables learning from very active users as well as facilities incorporating time into the learning process. Details of the proposed model is presented in Chapter 4. Note that an additional challenge is caused by users with very low check-in degrees. However, the first proposed model does not attempt to tackle this issue due to the difficulty of such problem. Short users are simply eliminated from the dataset in this case.

Model 2 The second proposed model, detailed in Chapter 5, builds up on the findings from the evaluation of the first proposed model, with two main contributions. One, the proposed model incorporates time at different granularities (hours of a day, day of a week, and so on) which enables inferring more meaningful patterns that depict the temporal regularities of real-world mobility patterns. This is achieved using a Bayesian approach that avoids overfitting of temporal patterns at minimum additional cost and complexity to a generic mixed-membership model. Two, an efficient and embarrassingly parallel inference algorithm is proposed to facilitate applying this model to large scale user location data. Similarly to the previous model, this model eliminates inactive users from the dataset.

Model 3 The last proposed model, detailed in Chapter 6, is designed to target inactive users, who dominate the users’ check-in distribution as they represent a large portion of users in social networks. The key contributions in this proposed model lie in utilizing ties in the social network, which allows learning from friends of users in order to compensate for their incomplete data and inactivity. In order to incorporate social ties into the learning model without jeopardizing the inference efficiency, the proposed model
utilizes a generalized ploy urn sampling approach and uncover patterns from collections of locations check-ins.

The remaining of this dissertation is organized as follows. Chapter 2 provides essential background including definitions of notations and details on the evaluation framework. Additionally, Chapter 2 includes a comprehensive literature review covering human mobility inference methods and link prediction with mobility patterns. Chapter 3 presents details of the baseline studies, including both methods: feature engineering and learning, and discusses important findings and remarks. Chapter 4 introduces the first proposed model, with experimental evaluation and discussion on findings. Chapter 5 introduces the second proposed model along with the proposed efficient inference methods. Chapter 6 introduces the last proposed model with evaluation and discussion on findings. Finally, Chapter 7 concludes this dissertation with summary of contributions and future research directions.
Chapter 2

Background

2.1 Preliminary

2.1.1 Notations and Problem Definition

Let \( v \) be a user from the set of unique users \( \mathcal{V} \) where \( |\mathcal{V}| \) is the number of users, and \( p \) be a location from the set of unique locations \( \mathcal{P} \) where \( |\mathcal{P}| \) is the number of locations. Users and locations are represented using unique identifiers.

**Check-in.** A check-in, \( c \), is the basic unit of observation in user-location data. In the simplest scenario, a check-in \( c = \langle v, p, t \rangle \) denotes user \( v \) visiting location \( p \) at timestamp \( t \). Depending on the datasource, other attributes can be observed in a check-in record such as the geophysical coordinates of the location (latitude and longitude) or location category (e.g., oriental restaurant). Table 2.1 shows a sample of a few user-location check-in records from the Gowalla (GW) and Brightkite (BK) [44].

<table>
<thead>
<tr>
<th>User ID</th>
<th>Timestamp</th>
<th>Latitude</th>
<th>Longitude</th>
<th>Location ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gowalla</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2010-07-05T04:17:34Z</td>
<td>50.28882455</td>
<td>5.9133318</td>
<td>1389275</td>
</tr>
<tr>
<td>1</td>
<td>2010-07-06T10:46:41Z</td>
<td>50.38100544</td>
<td>3.430709991</td>
<td>1395104</td>
</tr>
<tr>
<td>1</td>
<td>2010-07-07T07:53:20Z</td>
<td>49.61000082</td>
<td>3.803973198</td>
<td>1399686</td>
</tr>
<tr>
<td>1</td>
<td>2010-07-08T05:53:44Z</td>
<td>48.41966903</td>
<td>3.016968297</td>
<td>1404455</td>
</tr>
<tr>
<td>1</td>
<td>2010-07-10T03:33:04Z</td>
<td>46.30046749</td>
<td>5.740870777</td>
<td>1414779</td>
</tr>
<tr>
<td>Brightkite</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2009-06-06T19:44:05Z</td>
<td>37.63049</td>
<td>-122.411084</td>
<td>ec0fee</td>
</tr>
<tr>
<td>1</td>
<td>2009-06-07T06:59:08Z</td>
<td>37.63049</td>
<td>-122.411084</td>
<td>ec0fee</td>
</tr>
<tr>
<td>1</td>
<td>2009-06-07T13:50:12Z</td>
<td>37.616602</td>
<td>-122.404417</td>
<td>89fb8a</td>
</tr>
<tr>
<td>1</td>
<td>2009-06-08T03:10:05Z</td>
<td>37.63049</td>
<td>-122.411084</td>
<td>ec0fee</td>
</tr>
<tr>
<td>1</td>
<td>2009-06-08T17:26:09Z</td>
<td>37.584103</td>
<td>-122.366083</td>
<td>82f76e</td>
</tr>
</tbody>
</table>

Table 2.1: Sample of User-Location Check-in Records from two datasets.
datasets, where user ID, timestamp, latitude, longitude and location ID are recorded.

**Collection of Check-ins.** The collection of all observed check-ins is denoted as $\mathcal{C} = \{c_1, c_2, \ldots, c_N\}$, where $N$ is the number of all check-ins in the dataset. Each subfigure in Figure 2.1 visualizes 30,000 check-ins from GW and BK datasets on a map. In each subfigure 2.1(a) and 2.1(b) points represent individual check-ins and colors denote different users. The 30,000 check-ins represent 0.4% (0.6%) of the total check-ins recorded in each dataset, which covers check-ins for about 5,000 (2,000) users, in GW (BK).

![Figure 2.1: Collection of user-location check-ins from two datasets (a sample).](image)

(a) GW Collection of Check-ins (Sample)

(b) BK Collection of Check-ins (Sample)
**User-Location Matrix.** A user-location matrix, denoted as $X$, is a $|V| \times |P|$ sparse matrix of user-location check-in counts. $x_v$ is the raw vector of user $v$’s check-ins for each location $p \in P$, $x_p$ is the column vector of location $p$’s check-ins by each user $v \in V$, and $x_{v,p}$ is the number of times user $v$ checked-in at location $p$ over the data collection period. Figure 2.2 visualizes a sample of user-location matrix from two datasets, GW and BK. For each dataset, the sample includes a 100 users and a 100 locations, which represents a 0.09% (0.19%) of the users and 0.007% (0.01%) of the locations in GW (BK). Each sample is visualized using a heat map, where the color of check-in counts varies from dark blue (indicating no check-ins) and dark red (indicating maximum check-in count in this sample). Even though the dimensions of the sample (i.e., number of users and locations) is very small, the sparsity of such representation is very clear in both subfigures 2.2(a) and 2.2(b) as most user-location check-in counts are zero (dark blue). This sparsity is logical given the nature of the data, as people can only visit few locations out of the millions available, and are often bounded by space and time constrains [14].

**User Adjacency Matrix, a.k.a, Social Network.** Let $A^\kappa$ be a symmetric user-by-user matrix encoding users social links, where $a_{v,v'}^\kappa = 1$ indicates that users $v$ and $v'$ are connected by a social tie. Superscript $\kappa \in \mathbb{N}$ indicates the maximum degree of connection recorded in the matrix. For example, when $\kappa = 1$, i.e., $A^1$, a link between $v$ and $v'$ indicates that they are connected by a 1-hop degree (direct connection exists). For simplicity, we use $A$ instead of $A^1$ when $\kappa = 1$. When $\kappa = 2$, $a_{v,v'}^2 = 1$ means that they have a direct link or that they are connected via a 2-hop connection (friends of friends). An alternative explanation is that $A^\kappa$ encodes connected users with shortest path up to $\kappa$ hops. We use $a_v^\kappa$ to denote the vector of user $v$’s $\kappa$-hop social links.

Figure 2.3 visualizes a sample of a 1-hop user adjacency matrix, $A$, from GW and BK, respectively. The adjacency matrix is visualized as an unweighted undirected
Figure 2.2: User-location matrix from two datasets (a sample).
graph representing the topological structure of the social network, where users are represented as nodes and positive links, i.e., when \( a_{v,v'} = 1 \), as edges in the graph. Nodes are colored based on their degree (i.e., number of social links/edges) where blue indicates low number of links and red indicates the opposite. The sample in subfigure 2.3(a) includes 1000 links and 951 users from GW, which is about 0.2% and 0.8% of the actual links and users respectively in the dataset. Similarly, the sample in subfigure 2.3(a) includes 1000 links and 701 users, which is about 0.5% and 1.3% of the actual links and users respectively in BK. Both subfigures show that most users have low number of links, while only a few have many social connections.

![GW User Adjacency Matrix (Graph)](image1)

![BK User Adjacency Matrix (Graph)](image2)

(a) GW User Adjacency Matrix (Graph)   (b) BK User Adjacency Matrix (Graph)

Figure 2.3: User adjacency matrix from two datasets (a sample).

**Problem Definition.** Given a collection of user-location check-ins, \( C \), the objective is to represent users and locations in a new unified compact feature space, where the new space captures the latent mobility patterns (i.e., global mixture components) in the raw data. This problem is formally defined as latent mobility modeling; given the input \( C \), the objective (i.e., output) is to uncover the following new feature spaces:
1. User feature space, $\mathbf{h}_v$, a $1 \times K$ vector denoting the new representation space for user $v$ over the inferred global mixture components, where $K$ is the number of inferred mixture components. Each element $h_{v,k}$ indicates the relevance of component $k$ to user $v$. When packed row-by-row, it provides $\mathbf{H} = \{\mathbf{h}_1; \mathbf{h}_2; \ldots; \mathbf{h}_|V|\}$, a $|V| \times K$ matrix denoting the new users’ feature space. An illustration of this matrix was provided in Subfigure 1.4(a), showing a toy example of the user feature space, when $K = 2$. Quantitative evaluation of the inferred user feature space is achieved using social link prediction, where the quality of prediction is used as an indication of the goodness of the inferred patterns. An overview of social link prediction is provided in the next section.

2. Location feature space, $\mathbf{w}_p$, a $1 \times K$ vector denoting the new representation space for location $p$ over the inferred global mixture components. Each element $w_{p,k}$ indicates the relevance of component $k$ to location $p$. When packed row-by-row, it provides $\mathbf{W} = \{\mathbf{w}_1; \mathbf{w}_2; \ldots; \mathbf{w}_|P|\}$, a $|P| \times K$ matrix denoting the new locations’ feature space. An illustration of the transpose of this matrix, i.e., Feature-by-location space, was provided in Subfigure 1.4(b), showing a toy example of the features’ distributions over locations, when $K = 2$. Each column in this matrix denotes the relevance of locations to a pattern $k$, and thus it enables understanding the semantic meaning behind the pattern, by observing what locations are grouped together. Qualitative evaluation of this matrix is conducted in Chapters 5 and 6 by analyzing top locations per pattern.

The number of the global components, a.k.a., the dimension of the new representation space, $K \ll \min(|P|, |V|)$ is a predefined parameter. The global mixture components are represented as distributions over the basic observed unit, which can either be users or locations depending on the proposed model’s assumption, for specific details of each proposed model, please refer to Chapters 4, 5, and 6. The new representation space is restricted in the non-negative domain for interpretability.
2.1.2 Social Link Prediction

Social link prediction is utilized in this dissertation as a mean to evaluate the quality of the inferred user feature space (for an illustration of user feature space, please refer to Subfigure 1.4(a)). Evaluating this output, i.e., the user feature space, is important in this work as it provides an indirect way to quantitively compare different latent mobility models. This section includes general background, where an overview of the link prediction problem is provided, followed by a high-level analysis of the well-known characteristics of social networks. The background section is followed by a detailed description of link prediction methods, which includes details of unsupervised measures and supervised methods. Details of how the user feature space is included in link prediction methods is also provided.

2.1.2.1 Background

Link Prediction. Link prediction is an instance of dyadic prediction \[58\] which was formalized by Hofmann et. al. in 1999. The objective of this set of problems is to predict a label for the interactions between pairs of entities (dyads), rather than for a single entity. This problem is important because it provides a way of understanding one aspect of network evolution\[1\], i.e., new edge formation. Consequently, link prediction has applications in many fields including identifying interactions between protein pairs \[59\], inferring the structure of criminal/terrorist networks \[60\], and predicting friendships (links) between pairs of users in an online social network. The later is formally known as social link prediction \[61\] and is utilized in this dissertation to evaluate the inferred user feature space, where new links are predicted based on pairwise measures computed using the inferred users feature space.

As mentioned earlier, social link prediction is a well studied problem where the

\[1\]Network evolution includes the addition and deletion of links and edges in a network. However, link prediction focuses on discovering new (or missing) links only.
objective is to determine whether a pair of users are connected (or will form a connection in the near future). Consider for example the social network illustrated in Subfigure 2.4(a). The network consists of five nodes (from A to E) and five edges at a given time step. The task of link prediction is to infer missing (or new) links between pairs of existing nodes. This is illustrated in Subfigure 2.4(b) where new links (represented as dashed red lines) are predicted to form between nodes A − C and B − E.

![Diagram of a social network](image)

(a) Snapshot of a Social Network  (b) Illustration of link prediction for two pairs

Figure 2.4: Illustration of social link prediction.

**Problem Definition.** Given a user adjacency matrix, $A$, a snapshot of the matrix at a given time step, $A_{known}$, is defined to contain all users in $A$ but only a subset of the edges in $A$. Given $A_{known}$, the task of link prediction is to infer $A_{predict}$ which contains predictions of the missing (new) links that were originally in $A$ but not in $A_{known}$. This requires computing a score for each pair of users, where there is a total of $\frac{|V| \times (|V|-1)}{2}$ pairs in a network. Various methods exist for computing the score, and the ones utilized in this dissertation are discussed in Section 2.1.2.2. For now, note that the computed score represents a similarity measure between two users, and links are predicted to form between users with high similarity.

---

2 This instance of link prediction, where a snapshot of a network is given, is formally known as *structural* link prediction. This is different from *temporal* link prediction where multiple fully observed networks at different time steps are provided, and the objective is to predict the graph structure at the next time step. Link prediction in this dissertation is used to indicate *structural* link prediction in online social networks.
Recall that the objective of link prediction is to simply infer positive or negative links in the network (i.e., as in a binary classifier). This is achieved by ranking all pairs of users by their score, and setting a threshold by which pairs with higher scores are classified as positive links, and as negative otherwise. The decision rule for pair \( v \) and \( v' \) is thus given by,

\[
D(v, v') = \begin{cases} 
1 & \text{if } \text{score}(v, v') \geq \eta \\
0 & \text{otherwise}
\end{cases}
\] (2.1)

To evaluate the measured scores, one can simply compute the True Positive Rate (TPR) and False Positive Rate (FPR) for a selected threshold,

\[
TPR = \frac{TP}{TP + FN}, \quad FPR = \frac{FP}{FP + TN},
\] (2.2)

where \( TP, FN, FP, TN \) are the number of True Positives, False Negatives, False Positives, and True Negatives respectively.

Instead of setting a single threshold, one can vary the threshold and compute \( TPR \) and \( FPR \) for each threshold value. The resulted \( TPR \) and \( FPR \) for all threshold settings, when plotted, provide a curve known as the Received Operating Characteristics (ROC) curve (as illustrated in Figure 2.5(a)). The Area Under the ROC Curve (AUC), highlighted in Figure 2.5(b) can then be computed using any numerical integration methods, e.g., using Riemann sum or the trapezoidal method. AUC is a standard evaluation metric for link prediction, mainly because - unlike the standard accuracy metric - it is invariant to the extreme class imbalance problem in social links (i.e., disproportion of positive to negative links in social networks, as discussed in the next section). The maximum value of AUC is 1, and the value can be interpreted as how strong the prediction is. Thus, values close to 1 indicate good predictions while values close to 0.5 indicate poor or random predictors. Three examples of ROC
curves are shown in Figure 2.5(a) showing poor, good and great model results, with reference to random prediction.

Figure 2.5: Illustration of the ROC and AUC evaluation metric for link prediction.

Attributes of Social Networks. The structure behind social networks has several well known attributes that make such networks distinct. This includes the small world phenomenon [62], the power law degree distribution [63, 64], and the clustering effect [65]. Description along with high-level analysis of these attributes from GW and BK datasets is provided next.

- The small world phenomenon indicates that the average distance between pairs of users in the network is very small compared to the size of the network. The average shortest path link, $\hat{s}$, between any random pair of users in both datasets is small, $\hat{s} = 5.65$ in GW and $\hat{s} = 4.92$ in BK, due to the existence of a few hub nodes with relatively large degrees. Subfigures 2.6(a) and 2.6(b) show the shortest path link distribution of both datasets. Both distributions follow a log-normal/power-law, where the power law holds for large values, while a log-normal distribution describes small values [45].

- The scale-free phenomenon indicates that most nodes have few links in the network, and only a few have many links (known as hubs). That is, the node degree distribution is uneven. Subfigures 2.6(c) and 2.6(d) show the Complementary
Figure 2.6: Distributions of basic network measures: node degree, shortest path link, and clustering coefficient.
Cumulative Distribution Function (CCDF) of node degree (defined as the No. of links per node) in both datasets, where both fit a power-law (heavy-tailed) distribution with exponent $e = 1.6$ and $1.7$ in GW and BK, respectively.

- The clustering phenomenon indicates that there are small groups well connected in social network. The average local clustering coefficient, which measures the connectivity of a node’s neighborhood, is 0.23 and 0.17 in GW and BK, respectively. Subfigures 2.6(e) and 2.6(f) show the distribution of average clustering coefficient of nodes given their respective degrees.

These attributes provide a general overview of the online social network, where link prediction is performed. Details on link prediction methods utilized in this dissertation are provided next.

### 2.1.2.2 Link Prediction Methods

Variety of methods exist for link prediction, with a common objective of computing a similarity score for pairs of users in the network. In general, link prediction methods can be categorized as either unsupervised measures [61] or supervised classification methods [66]. Measures based on node proximity in the network topology, a.k.a., topology measures, are often regarded as the go-to approach for unsupervised link prediction. Topology measures are computed from the current structure of the user adjacency matrix $A$ alone, and often provide good prediction results [67]. The idea behind this concept is simple, users (i.e., nodes) are more likely to communicate (i.e., form edges) in the future if they are close or similar to each other in the present. Popularly used topology measures include Common Neighbors, Agamic-Adar, and Jacquard Coefficient, detailed next:
1. Common Neighbors (CN) is the number of common neighbors between pairs of users \( v \) and \( v' \), defined as:

\[
CN(v, v') = |N(v) \cap N(v')|,
\]

where \(|\cdot|\) gives the cardinality of a set and \( N(\cdot) \) is the set of neighbors for a given user.

2. Adamic-Adar (AA), a.k.a., Frequency-weighted common neighbors, is introduced by Adamic and Adar [68]. AA is a modified common neighbors measure, which uses the logarithm of user degree, rather than the even weight of 1, in the summation of common neighbors, defined as:

\[
AA(v, v') = \sum_{n \in CN(v, v')} \frac{1}{\log(|N(n)|)}.
\]

3. Jaccard coefficient (Jacc) is a weighted version of common neighbors, such that

\[
Jacc(v, v') = \frac{|N(v) \cap N(v')|}{|N(v) \cup N(v')|}. \quad (2.3)
\]

When additional data is accessible, node proximity can be measured using other implicit means in addition to the network topology, such as similarity in user features. Topology measures, along with user features, are often incorporated together to achieve better prediction results in a supervised framework. Supervised link prediction is a challenging problem due to the extremely unbalanced proportion of positive (existing) to negative (non-existing) links, where in two real datasets, only 0.008% (0.015%) of elements in \( A \) are positive with value 1 in GW (BK). Two supervised link prediction methods are utilized in this dissertation, supervised matrix factorization and feature-based classification using mainstream classifiers, detailed next.
Supervised Matrix Factorization with side features. Menon et. al. [69] proposed a supervised matrix factorization approach for link prediction where the proposed framework factorizes the user adjacency matrix and facilitates incorporating optional side features, as either node-specific features or edge-specific features. Additionally, the proposed method overcomes the imbalance problem by directly optimizing for a ranking loss, and is efficient when compared to other link prediction matrix factorization methods.

The objective function to minimize in the link prediction problem, in [69], is given by Equation (2.4), where \( E \) is the set of observed dyads such that \( E = \{(i, j): A_{ij} \neq ?\} \), where (?) represents the unknown dyads, \( E^+_i \) and \( E^-_i \) are the present and absent dyads of node \( i \) respectively. The link function \( L(\cdot) \) is defined as the sigmoid function, while the loss function \( l(\cdot) \) is the logarithmic function. The \( \Lambda \) is set to be an identity matrix \( I \), since the user adjacency matrix is symmetric. The inferred user features space are added to the model as a side feature \( h_v \). When no representation \( h_v \) is used, i.e., using only topology measures, the term \( h^T_i V h_j \) and its corresponding regularization term \( \frac{\lambda_V}{2} \| V \|^2_F \) are removed from Equation (2.4). This case, i.e., no added representations, is considered as a baseline in evaluation, for showing the effectiveness of adding the new representations.

\[
\min_{U, \Lambda, V, w, b} \frac{1}{|E|} \sum_{i=1}^{|V|} \sum_{j \in E^+_i, k \in E^-_i} l(L(u^T_i \Lambda(u_j - u_k) + h^T_i V h_j), 1) + \frac{\lambda_U}{2} \| U \|^2_F + \frac{\lambda_A}{2} \| \Lambda \|^2_F + \frac{\lambda_V}{2} \| V \|^2_F \\
(2.4)
\]

The optimization problem is solved using stochastic gradient descent, where the proposed approach learns from the known present and known absent dyads only, \( E^+_i \) and \( E^-_i \), without considering the unknown dyads (?) as the absent ones, in such a way that a regular SVD would do. The regularization parameter is used to avoid
over-fitting. Once the objective function is minimized using the known and absent dyads, the original adjacency matrix is reconstructed from the learned parameters and link prediction on unknown dyads is achieved.

**Feature-based Classification.** Feature-based methods \[70\] use general classifiers\[7\] to predict the existence of links between pairs of users, given a set of user-user features. Classifiers take as an input pairs of users, described by a combination of topology measures and similarity measures computed from their inferred user feature space. For example, for one pair of users \(v\) and \(v'\), the score of a link existing between them is predicted with input \{\(h_v - h_{v'}\), \(\text{Topology}(v_i, v_j)\}\}, where \(h_v\) and \(h_{v'}\) are users’ \(v\) and \(v'\) feature space learned from trajectories and \(\text{Topology}\) can be any of CN, AA or Jacc. That is, one topology measure is selected from the list of the three measures introduced previously, along with the distance between users feature space. This combination is then fed to a general classifier, where the objective is to infer a score for each pair. To overcome the issue of unbalanced labels, link prediction is limited to the 2-hop neighbors of each user. Thus, eliminating many true negative links from the learning space. Two mainstream classifiers are utilized here, which are decision trees and K-nearest neighbors.

Social link prediction is used as an evaluation measure for the inferred user feature spaces in Chapters 3, 4 and 6. In Chapters 3 and 4, the inferred user feature spaces are used along with topology measures for link prediction, using feature-based classification and matrix factorization with side features. Models are compared based on their performance on these methods, along with unsupervised topology measures as baselines. In Chapter 6, however, the inferred user feature spaces are used directly for link prediction, where a score between pairs of users is computed from their new feature spaces, i.e., \(-||\{h_v - h_{v'}\}\|_2^2\). Then, the ROC curve and AUC metric are computed

\[3\text{Note that classifiers are sensitive to the quality of the feature space, based on which results will vary significantly. Good features improve the quality of the results, while bad features degrade the performance.}\]
from the scores, and results are compared to unsupervised topology measures. The reason this simple approach was adopted in Chapter 6 because the proposed model incorporates social links into the inferred patterns and thus eliminates the need to add topology-based measures to the inferred representations, and allows using them directly to compute unsupervised scores.

2.2 Related Work

The related work discussed in this section is categorized into two general areas, which discusses work on inferring human mobility patterns and on link prediction with mobility patterns.

2.2.1 on Inferring Human Mobility Patterns

In literature, human mobility pattern inference is often done as an initial step towards achieving other tasks such as activity prediction and next place prediction. Models of human mobility can be categorized based on the method used for latent pattern inference. Earlier work attempted to exploit existing latent generic models such as Principal Component Analysis (PCA) (e.g., [71]) and Latent Dirichlet Allocation (LDA) (e.g., [72]). Temporal and spatial data are considered by carefully designing an input space that combines both. However, as there is a pressing need to properly leverage raw location data, research in this area has shifted towards designing special models, including for example [73] [74] [75]. Recently, Kapicioglu et al. [74] introduced Collaborative Place Models (CPM) to infer patterns from trajectories of geolocations. Phung et al. [73] proposed Latent Social theme Dirichlet Allocation (LSDA), based on N-gram LDA, for semantic location traces. Probabilistic Latent Semantic Trajectories (PLST), proposed in [76], models the semantic locations visited by a user as a mixture density of latent variables. Baratchi et al. [77] proposed a hierarchical semi-hidden Markov model which models single individuals (one at a time).
Table 2.2: Summary of related work by the used method, the type of location description, the treatment of trajectory incompleteness, and the incorporation of metadata as side features (time and social ties).

<table>
<thead>
<tr>
<th>Reference</th>
<th>Method</th>
<th>Location Description</th>
<th>Treatment of Incompleteness</th>
<th>Side Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eagle et al. [71]</td>
<td>PCA</td>
<td>Semantic</td>
<td>Filled by estimation</td>
<td>×</td>
</tr>
<tr>
<td>Farrahi et al. [72]</td>
<td>LDA</td>
<td>Semantic</td>
<td>Labeled as missing</td>
<td>×</td>
</tr>
<tr>
<td>Phung et al. [75]</td>
<td>LSDA</td>
<td>Semantic</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Joseph et al. [78]</td>
<td>LDA</td>
<td>ID</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Deb et al. [76]</td>
<td>PLST</td>
<td>ID</td>
<td>Data is complete</td>
<td>×</td>
</tr>
<tr>
<td>Baratchi et al. [77]</td>
<td>S-HMM</td>
<td>ID</td>
<td>Data is complete</td>
<td>×</td>
</tr>
<tr>
<td>Proposed Model I</td>
<td>HuMoR</td>
<td>ID</td>
<td>No</td>
<td>Time</td>
</tr>
<tr>
<td>Proposed Model II</td>
<td>M-Times</td>
<td>ID</td>
<td>No</td>
<td>Multi-scale Time</td>
</tr>
<tr>
<td>Proposed Model III</td>
<td>STAR</td>
<td>ID</td>
<td>Yes</td>
<td>Social Ties</td>
</tr>
</tbody>
</table>

Unlike existing work, the proposed models in this dissertation are tailored specifically to handle key attributes of human mobility data. The main difference resides on how (and why) model components were designed in each model. Comparison of existing work with the proposed models in terms of the type of location description handled by the model, the treatment of incomplete data, and the fusion of additional data (side features and social ties) is summarized in Table 2.2.

Existing work can be categorized based on the type of locations they learn from. A few methods, such as [74], learn patterns from raw GPS traces, while a larger set of methods require trajectories with semantically labeled locations as inputs for the model, e.g., [71] [72]. Semantically labeled trajectories denote user trajectories with locations described by their corresponding semantics, as opposed to their actual physical locations or names. Semantic labels of locations often include general categories such as home, work, leisure, and other, although specific labels can also be used, such as Chinese restaurant. Methods can also be designed to infer latent representations from IDs of locations, instead of from their actual geographical locations or semantical labels. For example, the method proposed in [77] first discretizes GPS locations into grids, and then infers patterns from IDs assigned to those grids.

From the three categories of location description discussed above, learning from
semantic labels of locations requires the most effort in preprocessing raw trajectory
data of either GPS or CDR locations. The only exception is data collected from
Location Based Social Networks (LBSNs) if location categories were collected together
with users check-in data. As for the geographical locations, not all human mobility
data is recorded in terms of their actual physical locations. Thus, models designed to
infer from actual physical locations cannot be applied directly to other forms of data.
As a result, in this dissertation, models are designed to learn from (anonymized) IDs
of locations. This minimizes preprocessing and the induced error in methods using
semantic locations, as well as insures that the proposed models work with data of
human mobility with different location types as well as various granularities (as they
all can be represented by their IDs).

Models of human mobility representation can also be categorized based on how
they handle incomplete user-location check-ins, i.e., missing data. Models proposed in
[74, 77] learn from (almost) complete GPS data collected using participatory sensing,
so no supplementary treatment of missing data is necessary here. Models proposed
in [71, 72] learn from complete with estimates trajectories, where authors first esti-
mated missing locations using prior knowledge of users locations, then learn from the
recovered trajectories. Models in [78, 75], learn mobility patterns from incomplete
semantically labeled trajectories with no attempt to compensate for the missing data.
The first two proposed models, HuMoR and M-TIMES, learn from long enough tra-
jectories of anonymized locations, without direct treatment of incompleteness but just
eliminating short trajectories. Note that incomplete trajectories are not necessarily
short, while short trajectories are usually incomplete. The third proposed model,
STAR, incorporates side features into the learning process to compensate for data
incompleteness. To the best of our knowledge, STAR is the first to introduce a model
that learns from incomplete trajectories with anonymized locations by leveraging on-
line social links.
Additionally, existing models can be categorized based on how time is incorporated into the inference models. Existing models have utilized a single time scale into the learning process, e.g., [73, 71, 72, 74, 75]. Farrahi et. al. [72] divided a day into blocks to discover daily location-based routines. A similar idea was used in [75] where a day is divided by 24 hours instead of blocks. Kapicioglu et. al. [74] combined global temporal patterns with user-specific spatiotemporal patterns and inferred each user’s location distribution for weekhours. The first proposed model, HuMoR [73], utilizes discrete time per group of user locations. The second proposed model, M-TIMES, is the first to consider multiple time scales during the inference process, using an efficient and scalable Bayesian approach. That is, the model considers different time scales in inference, such as the hours of a day, days of a week, and months of a year. Users of the model can control the level of details required in the temporal patterns by setting time scales differently.

Note that mixed-membership models are of wide popularity in text mining (generally known as topic models). There are approaches incorporating the temporal patterns while uncovering ‘static’ topics from text corpora. The approach proposed by Masada et. al. [80] handles time similarly to the second model, i.e., using a Bayesian approach. However, time is not considered as a multi-scale value, but rather as a single timestamp. Wang et. al. [81] proposed a method named TOT, in which time was modeled using a Beta distribution. npTOT [82] and TONPT [83] are two models attempting to overcome the limitations of TOT imposed by the unimodal distribution. However, both models introduce biases with the new introduced parameters, as well as complexity, often at the cost of the model’s speed. Additionally, both models treat time as a single continuous value. Thus, this approach cannot be adopted in mobility pattern inference as it ignores the patterns associated with multiple time scale. Moreover, in all models that assume a continuous time, time is initially normalized between 0 and 1, which results in loss of details.
Finally, the scalability and efficiency of mobility learning model are important, due to the need to learn from large scale user-location data. Existing work either 1) suffers from the vocabulary explosion problem, especially when time is combined with locations to generate words [72], or 2) is limited by the inefficiency of Gibbs sampling often used in LDA-like models. This being said, the inference algorithm proposed in Chapter 5 enables modeling spatial mobility patterns over multiple time-scales from large scale user-location data in an efficient way.

2.2.2 on Link Prediction with Mobility Patterns

Feature engineering, especially where features are extracted from mobility patterns, have been utilized in link prediction before. In fact, link prediction using location history has gained rapid attention recently [84][85]. One stream of research has focused on introducing new similarity measures based on actual physical locations [86][85][87][84]. The basic intuition behind these studies is: new links are more likely to form between people who have visited the same places. Scellato et. al. [84] examined the correlation between users visiting the same places and link formation in an online social network. The presented approach was able to reduce the prediction space using the findings of correlation. Wang et. al. [85] explored the relationship between network-based user similarity measures and actual spatial and temporal physical proximity. The results achieved by their method validate the logical intuition behind the adopted measure. Similarly, in this dissertation, the concept of utilizing mobility patterns to infer links in a network is adopted, using general link prediction methods. Yet, the main objective here is to evaluate the inferred user mobility patterns, through link prediction, rather than to propose state-of-the-art link prediction methods.
Chapter 3

Baseline Studies

The main goal of this chapter is to experimentally compare two categories of feature extraction, feature engineering and feature learning using latent feature models. For a general overview of each category, please refer to Section 1.2. In this chapter, new user feature spaces are extracted from user-location check-ins using feature engineering and four latent feature models. The new spaces are then evaluated based on their performance on social link prediction. The objective of this chapter has two folds, 1) to demonstrate the process of feature engineering, where features are manually extracted from the raw data as opposed to automatically inferred using existing latent feature models, and 2) to showcase the suitability of latent feature models in uncovering mobility patterns, where the new representations learned from generic latent feature models achieve equivalent results to those obtained using feature engineering, yet without the additional cost of human effort involved. More importantly, the representation learned using a generalized latent feature model achieved superior results to those obtained using either feature engineering or generic latent feature models.

The rest of this chapter is organized as follows. Section 3.1 describes features extracted using feature engineering from user-location check-ins, while Section 3.2 provides an overview of four latent feature models utilized in this study, which are: Principal Component Analysis (PCA), Latent Dirichlet Allocation (LDA), BiTerm Model (BTM), and a generalization of BTM denoted as Compact Representation Model (CRM). Section 3.3 then provides details of the evaluation setting and results, followed by analysis and discussion of the findings in Section 3.4.
3.1 Using Feature Engineering

The process of feature engineering described here follows the diagram illustrated in Figure [E.7], where features are extracted by carefully considering the data and task under study, and then evaluated in an iterative process until satisfactory results are achieved. As discussed earlier in Section 2.1.2, link prediction denotes the task of inferring new link formation between pairs of users, i.e., it answers the question; which pair of users are likely to meet (form a link) in the near future?. Thus, the objective here is to extract features from the given data that can help in predicting link formation. The datasets used in this chapter contain user-location check-ins, communication records (i.e., phone calls and text messages), and social links (to facilitate evaluation of results).

Knowing that individuals tend to associate with those who are similar\(^1\), extracted features for link prediction should capture similarity of users over as many dimensions as possible to obtain better results. Thus, given the data under study, the extracted features using feature engineering considered both location and communication data to achieve satisfactory results in link prediction. Three sets of features are extracted and described next, which are: location-based features, communication-based features, and hybrid-based features. These features are used to describe individuals, i.e., form the new users features space. Once the feature space is formed, it is used as input to supervised link prediction methods, as detailed in Section 2.1.2.2.

**Location-based Features.** This set of features describe users in terms of their frequency of visiting work and leisure locations. The set of work and leisure locations are inferred from the data using a simple rule-based approach, as illustrated in Algorithm [1] and described next. The intuition behind this set of features is to capture similarity in users mobility to work and leisure locations, because individuals with

\(^1\)This tendency is formally known as individuals homophily \([88]\) and it has been observed over many networks.
Algorithm 1 Locations Semantic Labeling

**Input:** Locations for each user $v \in V$

**Output:** $Work_G$, $Leisure_G$, and $Home_G$

for each user $v \in V$ do

work$_v = null$; leisure$_v = null$; home$_v = null$

for each location, $p$ do

if time of $p$ between 6 am and 5 pm on weekdays then

work$_v = work_v \cup p$ \{working hours\}

else if time of $p$ between 6 pm to midnight on weekdays and all day weekends then

leisure$_v = leisure_v \cup p$ \{leisure hours\}

else if time of $p$ between 1 am and 5 am then

home$_v = home_v \cup p$ \{home hours\}

end if

end for

end for

$Work_G = JOIN(work_v), \forall v \in V$

$Leisure_G = JOIN(leisure_v), \forall v \in V$

$Home_G = JOIN(home_v), \forall v \in V$

Three labels, home, work, and leisure, associated with users’ main places of interests are extracted from the data, using a simple rule-based approach (see Algorithm 1). The sets are extracted in two steps. First, home, work, and leisure locations are extracted per user $v$ based on the timestamp of each visit. For example, the label work is assigned to locations frequently visited during working hours in weekdays, while label leisure is assigned to locations visited at the evening during weekdays or weekends. Home label is assigned to locations frequently visited after midnight to before sunrise. After extracting the sets for every user, sets with the same label are grouped together for all the users to form a global set, resulting in a total of three global sets: $Work_G$, $Leisure_G$, and $Home_G$. Two sets of location-based features are then extracted, mobility in working and mobility in evening/leisure, detailed next.

- Mobility in Working. Locations labeled as work are used to characterize the users’ mobility in work locations. Each user is described by a vector of his visit frequency to the $Work_G$ locations. Users working/studying at same places have close distribution of their mobility in work vector. Thus, the difference in mobility patterns of work locations can be computed by the Euclidean distance of their corresponding
vectors. Users with small distance values are expected to be colleagues/classmates and are potential to have social links.

- Mobility in Evening/Leisure. Mobility patterns of evening/leisure are defined in an analogous manner. Locations labeled as evening/leisure are used to characterize users behaviors after working. The difference between mobility in evening/leisure vectors of two users quantifies how similar these users spend their leisure time at the recognized leisure places.

Mobility in home is not included here for link prediction, because 1) mobility in home is a sparse and high-dimensionality vector, since users have private homes, unlike work and leisure where such places are public and are visited by many, and 2) unlike homes, work and leisure places are more common for people to build their social ties.

**Communication-based Features.** This set of features quantify users’ communication patterns. This includes how often a user texts/calls per day, what is the average duration of a call and length of a text message, the average and variance of the difference between records, and so on. The intuition behind utilizing this set of features for link prediction is that users who have similar communication patterns (e.g., text frequently) are more likely to belong to a cluster of similar individuals (e.g., teenagers) and thus have higher chances of knowing each others given the homophily phenomenon.

**Hybrid-based Features.** This set of features quantify users’ communication patterns at specific locations. That is, this set of features are extracted from both communication logs and inferred locations. This includes features measuring the frequency of communication records (i.e., text, calls) when at work, home, and leisure.

Users are then described by the three sets of extracted features, location-based features, communication-based features, and hybrid-based features. This new user feature space is then evaluated using the two supervised link prediction methods,
detailed in Section 2.1.2.2. The results obtained using the features extracted by feature engineering are then compared to those inferred using latent feature models, detailed next, in an effort to experimentally compare these two approaches of feature extraction.

3.2 Using Latent Feature Models

Four latent feature models are utilized here to infer user feature space from user-location check-ins. The first three models, Principal Component Analysis (PCA), Latent Dirichlet Allocation (LDA), and Biterm Topic Model (BTM) are generic models, while the fourth model is a generalization of BTM proposed here for user-location check-ins. All latent feature models in this section infer the new user feature space, without taking into consideration timestamps of visited locations nor communication data, unlike in the feature engineering approach detailed previously.

3.2.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is an unsupervised, non-parametric, linear dimensionality reduction technique, that aims at finding a set of orthogonal (uncorrelated) Principal Components (PCs) from the original feature space. PCs are linear combinations of the original features. The importance of a PC is measured by how much variance it captures from the original input matrix, in which variance is assumed to indicate importance. PCs thus can be ordered by their variances in a decreasing order. This means that the 1st PC captures the largest variance in the data, and the 2nd PC captures the 2nd largest variance, and so on. PCA is used as a dimensionality reduction method by ignoring the PCs with the lowest variance (thus, lowest variability). This assures that less important information (i.e., noise) is removed from the data. The number of selected PCs is usually less than the number of features in the original space. Removing PCs with lowest variance denote eliminating unnec-
essential concepts, which may be noise or outliers, and thus only capturing the main conceptual components, which can be used to represent users in a lower dimensional space with little information loss.

Given the above definitions, the problem of PCA is to find a transformation matrix $M$ such that $H = XM$, where $X$ is the user-location check-in matrix of $|V|$ users (samples) and $|P|$ locations (features) and $H$ is a new de-correlated and ordered feature vector, denoting the inferred user feature space. This means that the off-diagonal values in $\text{cov}(H)$ are zeros (no correlation), and that the diagonal variances in $\text{cov}(H)$ are ordered from high to low. A wide range of memory-efficient and scalable algorithms for computing PCA are available, see [89] for an example. However, providing details of them is out of the scope of this work. In this work, PCA is used to reduce the original high-dimensional user-location check-in counts $X$ into a lower dimensional space $H$, where users are represented by a linear combination of the original feature space (PCs).

3.2.2 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA), proposed by Blei et al. [90] is a mixed-membership model, with both latent and observed random variables, designed to model discrete (count) data. In this baseline study, the input to LDA is a collection of users (i.e., statistically related groups), where each user has a collection of location check-ins. This input format indicates that the global mixture components, i.e., patterns, have distributions over locations, while the local mixture proportions represent the users distributions over patterns, i.e., the new users feature space.

To illustrate the conditional dependencies between the random variables in the model, Figure 3.1 visualizes the graphical model of LDA\footnote{Figure 3.1 shows the graphical model for LDA-smoothed as introduced in [90], but for short this model is referred to as LDA.} LDA is specified over a directed acyclic graph, where nodes represent random variables, directed edges de-
Figure 3.1: Graphical representation of LDA. Each node in the graph represents a random variable, shaded nodes are observed random variables, and plates indicate repetition; $|\mathcal{V}|$ is the number of users, $|\mathcal{C}_v|$ is the number of check-ins for every user, $\alpha$ and $\beta$ are the (hyper)parameters of the Dirichlet Distribution, $\mathbf{h}$ is the per-user distribution over patterns, $z$ is a selected pattern, $p$ is a location check-in, and $\mathbf{w}$ is the per-pattern distribution over locations.

Note conditional dependencies between parent and child nodes, and plates indicate repetition. Shaded grey nodes are observed variables, unshaded nodes are latent random variables, and small black nodes are hyperparameters of the prior distributions. Location check-ins $p$ is the only observed random variable in the model. $\mathbf{w}$, $\mathbf{h}$, and $z$ are latent random variables denoting the per-pattern distribution over locations, the per-user distribution over patterns, and the per-check-in latent pattern respectively. $K$, $|\mathcal{V}|$, and $|\mathcal{C}_v|$ are the number of latent patterns (pre-set by the user), the number of users and the number of check-ins per user respectively. $\beta$ and $\alpha$ are the hyperparameters of the prior distributions for $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K\}$ and $\mathbf{H} = \{\mathbf{h}_1; \mathbf{h}_2; \ldots; \mathbf{h}_{|\mathcal{V}|}\}$ respectively.

The graphical probabilistic model represents a factorization of the joint probability distribution for all random variables. That is, the joint distribution is a product of the conditional distributions in the figure, in which each random variable is conditioned
on its parent node. The full joint distribution of LDA is thus given by,

$$P(p, z, H, W | \alpha, \beta) = \prod_{k=1}^{K} P(w_k | \beta) \prod_{v} |V| P(h_v | \alpha) \prod_{c} |C_v| P(z | h_v) P(p | w_k = z). \quad (3.1)$$

The graphical representation of the model clearly defined the structure and dependencies between the random variables. Yet, the assumptions behind LDA are fully understood by considering the generative process behind the model, which explains how observed data can be generated from the model given the latent variables. The generative model provides details on the prior distributions along with their hyperparameters. The generative process of LDA is described next.

1. For each pattern, $k = 1 : K$
   
   (a) Draw $w_k \sim \text{Dirichlet}_{|P|}(\beta)$

2. For each user $v \in V$
   
   (a) Draw $h_v \sim \text{Dirichlet}_K(\alpha)$
   
   (b) For each location check-in $c \in C_v$
       
       i. Draw $z \sim \text{Multinomial}(h_v)$
       
       ii. Draw $p \sim \text{Multinomial}(w_k = z)$

The generative process of LDA illustrates that each pattern, $k$, has a multinomial distribution over $|P|$ locations ($w_k$), where $w_k$ has a Dirichlet prior parametrized

---

3The Dirichlet distribution is a continuous multivariate probability distribution. It is conjugate to the Categorical and Multinomial distributions. Thus, it is often used in Bayesian settings as a prior distribution to the parameters of the Categorical and Multinomial distributions, as it simplifies computation. This justifies the choice of the distributions used in LDA, and in consequent models proposed in this dissertation. Note that the prior distribution expresses one’s belief about a random variable before data is observed. A Dirichlet distribution with order $K \geq 2$ and concentration parameters of positive real values $\alpha = \{\alpha_1, \ldots, \alpha_K\}$ is denoted as Dirichlet$_K(\alpha)$. A vector $\theta$ drawn from Dirichlet$_K(\alpha)$ is a $1 \times K$ vector of probabilities, where $\theta_i \in (0, 1)$ and $\sum_{i=1}^{K} \theta_i = 1$, iff $0 \leq \alpha_i < 1, i = 1 \ldots K$. A simplified case is when $\alpha$ is symmetric, i.e., $\alpha_1 = \cdots = \alpha_K$, where Dirichlet$_K(\alpha)$ reduces to Dirichlet$_K(\alpha)$. Detailed discussion of the effect of symmetric vs. asymmetric parameters on the Dirichlet distribution is provided in Chapter 4.
by a symmetric hyper parameter $\beta$. Additionally, each user, $v$, is represented by a vector of multinomial distributions over the latent patterns ($h_v$), where $h_v$ has a Dirichlet prior parametrized by a symmetric hyper parameter $\alpha$. For each user $v$, $h_v$ represents the probability of latent patterns, where a latent pattern $z$ is selected based on its probability of occurrence, $P(z_v = k) = h_v^{(k)}$. Location check-ins are then drawn for each user based on $h_v$ and $w_k$, where first a latent pattern $z$ is sampled from a Multinomial distribution with parameter ($h_v$). Then, a location is sampled from a Multinomial distribution with parameter $w_{k=z}$, which denotes the distribution of the global mixture component associated with the selected pattern $z$.

At this point, the generative process detailed how observations can be generated from the model, according to the assumptions and distributions specified. However, the objective of LDA is to model observed data, not generate it. Thus, it is necessary to infer the latent random variables in the model, given the observed data, i.e., infer the posterior distribution. The posterior distribution of the model is given by,

$$P(z, W, H | p, \alpha, \beta) = \frac{P(z, W, H, p | \alpha, \beta)}{P(p | \alpha, \beta)}.$$  (3.2)

The integral of the marginal likelihood, $P(p | \alpha, \beta)$, is intractable to compute exactly mainly due to the dependencies in the model. Thus, approximate inference algorithms are used to infer the posterior instead. Precisely, the objective is to infer the latent variables $W$ and $H$, denoting the latent representations for locations and users respectively. A variety of approximate inference methods exist, with two main categories: variational inference (e.g., the Expectation-Maximization algorithm) and Markov Chain Monte Carlo (e.g.,Collapsed Gibbs Sampling). Details of existing inference methods are out of the scope of this chapter. However, inference methods for the proposed models in this dissertation are detailed in their corresponding chapters.

The intuition and assumptions behind LDA can be easily adopted to user-location data - even without tailoring the model to the specific data attributes and considering
informative metadata (such as order of check-ins). However, aside from the importance of order in trajectory data, two main problems are anticipated when adopting LDA to user-location data. First, LDA is sensitive to frequent locations, where frequent locations dominate the patterns’ distributions over locations [91]. One possible solution is to adopt an asymmetric hyper parameter ($\alpha$) over the Dirichlet distribution prior of the local mixture proportions ($h_v$), as proposed in [92]. This variation of LDA, known as LDA-Asymmetric (LDA-A for short), is adopted in the baseline experiments for evaluation. Figure 3.2 illustrates the effect that different parameters have on a Dirichlet distribution. The second anticipated issue is the activity level of users, precisely users with low number of check-ins, given that LDA does not perform well in short documents [93]. To overcome this issue, another baseline method, the Biterm Topic Model is considered.

### 3.2.3 Biterm Topic Model (BTM)

Biterm Topic Model (BTM) is a probabilistic graphical model designed for short documents [93]. When documents in a corpus are represented by a vector of vocabulary check-ins, i.e., bag-of-words, the feature space becomes increasingly sparse as documents length decreases. This issue is problematic for LDA, since the patterns inferred by LDA depend on word (location) co-occurrences in the document (user) level. BTM is proposed to overcome this issue. The graphical representation of the model is presented in Figure 3.3, where the conditional dependencies between the random variables in the model are visualized. Given the conditional dependencies in the graphical model, the full joint distribution of BTM is given by,

$$
P(p, z, \theta, W|\alpha, \beta) = P(\theta|\alpha) \prod_{k=1}^{K} P(w_k|\beta) \prod_{d} P(z|\theta) P(p_i|w_{k=z}) P(p_j|w_{k=z}). \quad (3.3)$$

Similarly to LDA, BTM has $K$ global mixture components, $W = \{w_1, w_2, \ldots, w_K\}$,
Figure 3.2: Simplex of a 3-parameter Dirichlet distribution, Dirichlet(\(\alpha\)), with symmetric priors (a) when \(\alpha = [1,1,1]\) which is a special case with equal probabilities, (b) when \(\alpha = [2,2,2]\), and (c) when \(\alpha = [3,3,3]\) where larger values in \(\alpha\) imply more concentration towards the center of the simplex, and asymmetric priors (d) when \(\alpha = [2,3,1]\), (e) when \(\alpha = [3.6,5.5,1.8]\) and when (f) when \(\alpha = [5.3,8,2.6]\). Higher probabilities are indicated by dark red, and it decreases gradually until 0 (indicated by dark blue).
Figure 3.3: Graphical representation of BTM. Each node in the graph represents a random variable, shaded nodes are observed random variables, and plates indicate repetition; $\alpha$ and $\beta$ are the parameters of the Dirichlet priors, $\theta$ is a community-level distribution over patterns, $z$ is a selected pattern, $p$ is a location check-in, $w_k$ is the multinomial distribution of pattern $k$ over all locations, and $|D|$ is the number of unordered pairs of location check-ins.

where each mixture component $w_k$ is a multinomial distribution over all locations $|P|$. Additionally, each $w_k$ has a Dirichlet prior parametrized by a symmetric hyper parameter $\beta$. However, unlike LDA where the model has per-user local proportions, BTM has a community-level mixture proportion that specifies the probability of latent patterns for all biterms, $D$, in the dataset. A biterm, $d$, is a pair of locations, $p_i$ and $p_j$, in a user’s check-in records, where $p_i$ and $p_j$ are not necessary in order (unordered). The collection of all extracted biterms is denoted as $D$. The intuition behind BTM is to model unordered pairs of locations visited by the same user. The model is specified by its generative process, detailed next.

1. Draw $\theta \sim \text{Dirichlet}_K(\alpha)$

2. For each pattern, $k = 1:K$

   (a) Draw $w_k \sim \text{Dirichlet}_{|P|}(\beta)$

3. For each bitem $d \in |D|$
(a) Draw $z \sim \text{Multinomial}(\theta)$

(b) Draw $p_i \sim \text{Multinomial}(w_{k=z})$

(c) Draw $p_j \sim \text{Multinomial}(w_{k=z})$

The latent variables in the model are inferred using collapsed Gibbs sampling, detailed in [93]. Once the model's parameters are inferred, the per-user local proportions can be computed given the inferred parameters $\theta$ and $w_k$. First, it assumes that the user-local proportions, $h_v$, equals to the expectation of the patterns associated with the user-biterms (i.e., biterms generated from the user). That is, the user-local proportions over latent patterns for user $v$, his generated biterms $|D_v|$, and a selected pattern $z$ is given by,

$$P(z|v) = \sum_d P(z|d)P(d|v).$$

(3.4)

where $P(z|d)$ is computed from the latent parameters inferred by BTM, i.e., $\theta$ and $w_k$, and $P(d|v)$ denotes the empirical distribution of a biterm in the user check-ins.

The main objective of BTM is to overcome the sever sparsity issue in the bag-of-words matrix caused by short documents. Though it succeeded at inferring better patterns from short documents as illustrated by Cheng et. al. [93], BTM has two main drawbacks. First, BTM is very sensitive to frequently visited locations, where such locations end up dominating the mixture components resulting in uninteresting patterns. Second, the use of pairs of unordered locations makes BTM an unscalable framework for modeling large scale user-location data. This is because in a dataset of user-location check-ins, with $|C_v|$ number of check-ins per user, the number of generated biterms $|D| = \frac{|C_v|(|C_v|-1)}{2}$.

To overcome the first drawback, frequently visited locations are eliminated from the dataset in baseline experiments because the alternative solution which suggests utilizing an asymmetric hyper parameter over the local proportions cannot be adopted
Figure 3.4: Graphical representation of CRM. Each node in the graph represents a random variable, shaded nodes are observed random variables, and plates indicate repetition; $\alpha$ and $\beta$ are the parameters of the Dirichlet priors, $\theta$ is the corpus-level distribution over patterns, $z$ is a selected pattern, $p$ is a location check-in, $w_k$ is the multinomial distribution of pattern $k$ over all locations, $I$ is the length of a sequence, and $|S|$ is the number of ordered sequences of location check-ins.

in this framework. To overcome the second drawback, on the other hand, a generalization of BTM is proposed next as the fourth baseline model, where ordered sequences of location check-ins are considered instead of unordered pairs. This generalization significantly reduces the learning set from $(|C_v| \cdot \frac{(|C_v|-1)}{2})$ to $(|C_v| - I + 1)$, where $I$ is the length of the sequence. As a result, the associated running time and complexity is significantly reduced.

3.2.4 Compact Representation Model (CRM)

Compact Representation Model (CRM) is a generalized version of BTM, where patterns are inferred from collections of ordered sequences of length $I$, as opposed to collections of unordered biterms of length 2. For example, in BTM a biterm contains two locations $c_i$ and $c_j$ where $i$ and $j$ are not necessary in order, where as a sequence in CRM with length $I = 2$ contains $c_i$ and $c_{i+1}$. Although restricting length to 2 and
considering unordered pairs is a reasonable assumption for text corpora, it does not translate well to user-location data. Consequently, this generalization is done for two main reasons, 1) to reduce the number of biterms as the sequence length becomes larger and only ordered check-ins are considered, which makes the proposed generalized model suitable to very large datasets, and 2) in user location data, especially if locations are sampled frequently with small elapsed time between consecutive points, extracting sequences with only consecutive locations maximizes the probability that such locations would share the same selected pattern.

The graphical representation of the model is illustrated in Figure 3.4. The graphical model illustrates the difference between BTM and CRM, where \( \mathcal{I} \) locations are drawn for each selected pattern instead of a fixed length of 2. The full joint distribution of CRM given the graphical model is thus given by,

\[
P(p, z, \theta, W|\alpha, \beta) = P(\theta|\alpha) \prod_{k=1}^{K} P(w_k|\beta) \prod_{s} P(z|\theta) \prod_{i} P(p_i|w_{k=z}). \tag{3.5}
\]

The generative process of CRM further illustrates the difference between CRM and BTM, detailed next.

1. Draw \( \theta \sim \text{Dirichlet}_K(\alpha) \)

2. For each pattern, \( k = 1 : K \)
   (a) Draw \( w_k \sim \text{Dirichlet}_{|P|}(\beta) \)

3. For each sequence \( s \in |S| \)
   (a) Draw \( z \sim \text{Multinomial}(\theta) \)
   (b) For each element in the sequence, \( i \in |\mathcal{I}| \)
      i. Draw \( p_i \sim \text{Multinomial}(w_{k=z}) \)
For the collection of sequences generated from user-location data, the generative process can be explained as follows. For every extracted sequence, a pattern is drawn from the community-level distribution. Then, for every element in the sequence, a location is drawn from the latent pattern’s distribution over locations.

Illustration of how ordered sequences are extracted from CRM is provided in Subfigure 3.5(a). Subfigure 3.5(a) visualizes location check-ins for a single user, where check-ins are denoted by points in the path. An extracted sequence is an ordered list of check-ins visited by a user, with $I$ number of check-ins per a sequence. Subfigure 3.5(a) visualizes 5 extracted sequences from the user location data, each with length $I = 5$. Note that each sequence is assigned to 1 selected pattern, $z$, unlike in LDA where each check-in is drawn a different $z$. This model choice fits user-location data with frequent check-ins per user, such as in the datasets used in the experiments of this chapter, yet it is unrealistic for sparse user location data where the elapsed time between two consecutive check-ins is usually large.

An illustration of the intuition behind CRM is provided in Figure 3.5. Subfigure 3.5(b) visualizes the collection of sequences, $S$, extracted from all users in the dataset, represented as lines inside the black circle. In this toy example, the number of latent patterns $K = 3$, and thus 3 distinct colors are used to represent the 3 patterns. The user icons below the circle indicates that the sequences are extracted from all users in the dataset, and not only one. Additionally, the community-level mixture proportion over latent patterns, $\theta$, is illustrated by the colored bar charts in Subfigure 3.5(b). A pattern $z$ is drawn from a multinomial distribution with $\theta$ probabilities, for each sequence. This is visualized using colors in the graph, where each sequence is assigned a single color. It is apparent from the toy example that $\theta_2$ has higher probability of occurrence in the community, and thus more sequences are colored red. Finally, the multinomial distribution of patterns over locations is illustrated in Subfigure 3.5(c) showing locations with higher probabilities of occurring in each patterns.
Figure 3.5: A high-level illustration of the Compact Representation Model (CRM).
The posterior distribution of the model is given by,

\[ P(z, W, \theta | p, \alpha, \beta) = \frac{P(z, W, \theta, p | \alpha, \beta)}{P(p | \alpha, \beta)}. \]  

(3.6)

Similarly to LDA and BTM, the posterior in this model is intractable and thus approximate inference methods are adopted. In this case, a collapsed Gibbs sampler is proposed to infer the latent patterns, which is simply a generalization of the BTM collapsed Gibbs sampler proposed in [93]. Gibbs sampling is a Markov Chain Monte Carlo (MCMC) method, which iteratively draws and updates one latent variable from the posterior given all other latent variables and observations. Sampling is then repeated for each latent variable and over hundreds of iterations. Based on the MCMC theory, after an initial period known as the burn-in period, the resulting sample will be from the true posterior.

Recall that CRM has three latent random variables to infer, \( z, W, \theta \). The mixture assignment, \( z \), is a sufficient statistic of both \( W, \theta \). \( W, \theta \) are thus collapsed into \( z \), and the inference problem reduces to sampling \( z \) given all other latent variables \( z_{-i} \) and observations \( S \). Using collapsed Gibbs sampler, sampling from the posterior is given by,

\[ P(z_i | z_{-i}, S) \propto (\eta_k^{(-i)} + \alpha_k) \prod_{j=1}^{\tau} \frac{(\eta_{p_j | k}^{(-i)} + \beta)}{(\eta_k^{(-i)} + j - 1 + \beta)} \]  

(3.7)

where \( \eta_k = \sum_{s=1}^{\left| S \right|} \delta(z_s - k) \) is the number of times pattern \( k \) was drawn, superscript \( (-i) \) is used when sequence \( i \) is excluded, \( \eta_{p_j | k} \) is the number of times location \( p_j \) was assigned to pattern \( k \), and \( \eta_{j | k} = \left\{ \eta_{p_j | k} \right\}_{j=1}^{\left| P \right|} \) is the total number of locations assigned to pattern \( k \).

Equation 3.7 is derived by employing two basic concepts (1) the chain rule, and (2) the Dirichlet-multinomial conjugacy property; which states that the probability
density function of a Dirichlet-multinomial is a Dirichlet distribution with an updated parameter. After repeatedly sampling from Equation 3.7 over hundreds of iterations, samples of $z$ will be drawn from the true posterior. At this point, both $W, \theta$ can be computed using the sampled $z$, given the expectation of a Dirichlet distribution $Dir(x|\alpha)$, which is

$$E(x_i) = \frac{\alpha_i}{\sum_{k=1}^{K} \alpha_k}.$$ (3.8)

More precisely, given $z, S, \alpha, \beta$, the location distribution of each pattern can be estimated as: $P(W|S, \beta) = \frac{1}{Z_w} Dir(W|\beta + \eta_{i,k})$, where $Z_w$ is a normalization factor and $\eta_{i,k} = \{\eta_{p|k}\}^{[P]}$. According to Equation (3.8), each element in the matrix $W$ is estimated using,

$$w_{p|k} = \frac{\beta + \eta_{p|k}}{\sum_{p=1}^{[P]} (\beta + \eta_{p|k})}.$$ (3.9)

The community-level distribution over patterns is given by $P(\theta|z, \alpha) = \frac{1}{Z_\theta} Dir(\theta|\alpha + \eta)$, where $\eta = \{\eta_{k}\}_{k=1}^{K}$. Again, according to Equation (3.8), The $k$-th element in the motif distribution vector $\theta$ is given by,

$$\theta_k = \frac{\alpha_k + \eta_k}{\sum_{k=1}^{K} (\alpha_k + \eta_k)}.$$ (3.10)

Finally, the model does not infer the new user representation over latent patterns directly. Instead, the new user feature space is approximated using the inferred latent patterns from CRM. Suppose a user $v \in V$ has $|S_v|$ number of sequences, after learning patterns from sequences of all users, the per-user pattern distribution $P(z|v)$ is approximated using the chain rule and the obtained $\Phi$ and $\Theta$. 
\[ P(z|v) = \sum_{i} P(z, s_{i,v}|v) = \sum_{i} P(z|s_{i,v}, v)P(s_{i,v}|v) \quad (3.11) \]

\( P(s_{i,v}|v) \) can be estimated empirically from the data,

\[ P(s_{i,v}|v) = \frac{\eta_{s_{i}|v}}{\sum_{j=1}^{|S_v|} \eta_{s_{j}|v}} = \frac{\eta_{s_{i}|v}}{\eta_{.|v}} \quad (3.12) \]

where \( \eta_{s_{i}|v} \) is the frequency of sequence \( i \) in user \( v \). As for \( P(z|s_{i,v}, v) \), assume that pattern \( z \) and user \( v \) are conditionally independent given \( s_{i,v} \), and patterns are independent from each other. Then,

\[ P(z|s_{i,v}, v) = P(z|s_{i,v}) = \prod_{k} P(z_k = k|s_{i,v}) \quad (3.13) \]

where

\[ P(z = k|s) = \frac{P(s|z)P(z)}{P(s)} = \frac{\prod_{i} w_{k,p_i} \theta_k}{\sum_{k'} \theta_{k'} (\prod_{i} w_{k',p_i})} \quad (3.14) \]

Therefore, taking Equation (3.12) and (3.14) into (3.11),

\[ P(z|v) = \sum_{i} \left( \prod_{k} \frac{\prod_{i} w_{k,p_i} \theta_k}{\sum_{k'} \theta_{k'} (\prod_{i} w_{k',p_i})} \eta_{s_{i}|v} \right) \quad (3.15) \]

The complete algorithm for inferring the latent random variables in CRM is illustrated in Algorithm 2.

3.3 Evaluation

Recall that the objective of this chapter is to experimentally evaluate the new user representations learned using feature engineering and latent feature models. The user representation space learned by each baseline method, FE, PCA, LDA, LDA-A, BTM,
Algorithm 2 Baseline Model (CRM): Estimating Model Parameters

**Input:** $K, S, \alpha, \beta$

**Output:** $W, \theta$

Randomly initialize $z, \forall s \in S$ and update counters

for each iteration do

for each sequence $s \in S$ do

for each pattern $k \in K$ do

$$P(z_i = k | z_{-i}, S) = \frac{(\eta_k^{(-i)} + \alpha_k) \prod_{j=1}^T \frac{(\eta_{y_{ij}|k}^{(-i)} + \beta)}{\eta_{y_{ij}|k}^{(-i)} + \beta + j + \beta}}$$

sample $k \sim P(z|S)$

Compute $W$ and $\theta$ where $w_{p|k} = \frac{\beta + \eta_{p|k}}{\beta + \eta_{p|k} + |S|}$ and $\theta_k = \frac{\alpha_k + \eta_k}{\alpha_k + |S|}$

Compute $P(z|v)$.

---

Table 3.1: Datasets Statistics. MDC and RM Basic Statistics.

<table>
<thead>
<tr>
<th></th>
<th>MDC</th>
<th>RM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data collection period</td>
<td>19 months</td>
<td>9 months</td>
</tr>
<tr>
<td>Number of Users</td>
<td>158</td>
<td>107</td>
</tr>
<tr>
<td>Number of Active users</td>
<td>107</td>
<td>84</td>
</tr>
<tr>
<td>Number of Social links</td>
<td>197</td>
<td>151</td>
</tr>
<tr>
<td>Number of Unique Locations</td>
<td>103,655</td>
<td>29,617</td>
</tr>
</tbody>
</table>

and CRM, is utilized in a link prediction task, using supervised matrix factorization and feature-based classification, as detailed in Section 2.1.2.2. The results achieved by each representation in the link prediction task are used as an indirect measure to evaluate the quality of the inferred user representations.

Evaluation is conducted on two datasets: Mobile Data Challenge (MDC) and RealityMining (RM), both with relatively dense users trajectories collected using participatory sensing. Note that only a subset of the original data is utilized here, users with few records (i.e., inactive users) are eliminated, and successive location points are combined to reduce the sampling rate. Basic statistics of the both datasets used in this evaluation are presented in Table 3.1 and details of the datasets are provided next.

**Dataset I.** The Mobile Data Challenge (MDC) [94] dataset was collected under
the initiative of the Lausanne Data Collection Campaign, which started in 2009 and lasted for about 19 months. The data was collected from 158 participants (107 are active) and it consists of (a) the individuals’ phone usage including application and media usage, as well as calendar and contacts information, (b) communication logs which include call logs, text messages, and data connections, (c) sequences of Global Positioning System (GPS) trajectories and cell-IDs, and (d) user specific data, which includes demographics such as age, gender, and work-status. MDC is a CDR data, where cell-IDs associated with communication records are regarded as visited locations. Communications between users are represented as undirected links in the social graph.

**Dataset II.** The RealityMining [95] dataset was collected by MIT Media Laboratory from 2004 to 2005. The group of participants contained a mixture of undergraduate and graduate students from different research labs and programs, as well as staff and research scientists. The data set contains (a) users’ communication activities including phone calls, text messages, and data connections, (b) a list of nearby bluetooth devices and cell-IDs, and (c) users’ affiliation, social ties, and behavioral patterns. A total of 107 users participated in the data collection campaign. 23 of the users were not active during the collection period, and thus were removed from our experimental evaluation. Active users, in this case, are defined as the set of users who have at least one communication record during the data collection period.

**Experimental Settings.** All latent feature models inferred patterns from the same user-by-location matrix of check-in counts, where frequently visited locations are removed for better inference. This is analogous to stop word removal in topic discovery. All graphical models, LDA, LDA-A, BTM and CRM, are trained for 1000 iterations, and the asymmetric $\alpha$ parameter in LDA-A is optimized every 50 iterations, after an initial burn-in period of 200. The common parameter $\beta$ for all
graphical models is set to be $\beta = 0.01^4$ and $K = 5$ in all models unless otherwise specified.

**Visualization of the Inferred Patterns.** To understand the output of the latent feature models, the user feature space inferred using CRM is visualized using a heatmap in Figure 3.6 for both datasets. The heatmap not only illustrates the user feature output, but also shows that clusters of similar users can be identified from the new representation space. In the MDC dataset for example, Figure 3.6(a) visualizes the user feature space for all users, where columns represent patterns ($K = 5$), rows represent users, and values are user $v$’s belongingness to pattern $i$. Clusters of users with similar patterns can be easily recognized in this representation space.

After further investigation of users in each cluster, it was found that members of each cluster belong to the same age group (mostly), where age groups of users are provided as labels in the MDC dataset. Two sets of age groups are identified; Group 1 and Group 2 referring to users with ages between [22, 49] and ($<21$), represented by blue and purple bars respectively. From the heatmap, clusters of users with the same patterns belong to mostly the same age group. Yet the same age group can span multiple clusters, each with unique pattern (see Age group 1 for example). Additionally, there also exist a cluster of users with similar pattern distribution, but belonging to different age group, i.e., mixture of groups (marked white).

Clusters of similar users can also be identified in the RM dataset. In Figure 3.6(b), two sets of users with the same affiliation are identified Graduate and Business Students (colored blue and purple respectively), where the affiliation label is provided in the RM dataset. There exist multiple clusters with the same affiliation (e.g., 2 clusters of Business Students) which indicate that not all members of the same affiliation must have the same pattern distribution, and the opposite is also true in which users with similar distributions belong to the same cluster but have dissimilar

---

4Setting $\beta \ll 1$ generates sparse pattern representations, and thus increases patterns separability.
affiliations (marked white). This simple analysis shows that clustering users by their mobility patterns can reveal interesting subgroups in the population.

**Link Prediction.** In the MDC dataset, the social network $A$ is split into $A_{known}$ and $A_{predict}$ by the date of June 2010, leaving 7 months of data for learning and 4 months for testing. There is a total of 107 active users with 197 links in $A$, from which 168 links are in $A_{known}$, and 29 in $A_{predict}$. Topology measures for link prediction are extracted from $A_{known}$, and the new representations are extracted from mobility logs between November 2009 and June 2010. The new representations are then used to predict new links in $A_{predict}$, using feature-based classification and matrix factorization. The same procedure is applied to RM dataset.

Evaluation results (AUC values) of link prediction, on the MDC and RM datasets, are shown in Table. 3.2. Values under *Representations* indicate the AUC (actually mean AUC averaged over 10 runs) values obtained when the new representations used in conjunction with pairs of topology measures and prediction methods (in rows). MF indicates Matrix Factorization without added representations and MF-
Table 3.2: Mean AUC for link prediction for MDC and RM.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Baseline Measures</th>
<th>Methods</th>
<th>Representations</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CN 0.645</td>
<td>Trees 0.782</td>
<td>FE</td>
<td>0.805</td>
</tr>
<tr>
<td></td>
<td>KNN 0.859</td>
<td>PCA</td>
<td>0.883</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LDA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LDA-A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>BTM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CRM-2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CRM*</td>
<td></td>
</tr>
<tr>
<td>AA 0.645</td>
<td>Trees 0.779</td>
<td>FE</td>
<td>0.802</td>
</tr>
<tr>
<td></td>
<td>KNN 0.861</td>
<td>PCA</td>
<td>0.884</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LDA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LDA-A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>BTM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CRM-2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CRM*</td>
<td></td>
</tr>
<tr>
<td>Jacc 0.645</td>
<td>Trees 0.779</td>
<td>FE</td>
<td>0.810</td>
</tr>
<tr>
<td></td>
<td>KNN 0.845</td>
<td>PCA</td>
<td>0.887</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LDA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LDA-A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>BTM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CRM-2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CRM*</td>
<td></td>
</tr>
<tr>
<td>MF 0.820</td>
<td>MF-w 0.873</td>
<td>FE</td>
<td>0.901</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PCA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LDA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LDA-A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>BTM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CRM-2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CRM*</td>
<td></td>
</tr>
<tr>
<td>RM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CN 0.708</td>
<td>Trees 0.792</td>
<td>FE</td>
<td>0.811</td>
</tr>
<tr>
<td></td>
<td>KNN 0.797</td>
<td>PCA</td>
<td>0.850</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LDA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LDA-A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>BTM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CRM-2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CRM*</td>
<td></td>
</tr>
<tr>
<td>AA 0.712</td>
<td>Trees 0.768</td>
<td>FE</td>
<td>0.822</td>
</tr>
<tr>
<td></td>
<td>KNN 0.791</td>
<td>PCA</td>
<td>0.860</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LDA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LDA-A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>BTM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CRM-2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CRM*</td>
<td></td>
</tr>
<tr>
<td>Jacc 0.704</td>
<td>Trees 0.776</td>
<td>FE</td>
<td>0.849</td>
</tr>
<tr>
<td></td>
<td>KNN 0.791</td>
<td>PCA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LDA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LDA-A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>BTM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CRM-2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CRM*</td>
<td></td>
</tr>
<tr>
<td>MF 0.755</td>
<td>MF-w 0.752</td>
<td>FE</td>
<td>0.839</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PCA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LDA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LDA-A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>BTM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CRM-2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CRM*</td>
<td></td>
</tr>
</tbody>
</table>

w indicate MF with the inferred representations. Two feature-based classifiers are used: unpruned decision trees and K-Nearest Neighbors, and three topology measures (Common Neighbors (CN), Adamic Adar (AA), and Jaccard Coefficient (Jacc)), in conjunction with seven variants of user feature space, inferred using FE, PCA, LDA, LDA-A, BTM\(^5\), CRM when \(I = 2\), and CRM* denoting the best setting of \(I\).

The AUC results provide interesting insights on the new user feature spaces. First, the user feature spaces, regardless of the methods used to infer them, provide significant improvement in link prediction when compared to the unsupervised topology measures. New user space extracted using feature engineering indeed helps improve the link prediction results. However, similar and sometimes superior results are also achieved using user feature spaces inferred by generic latent feature models, as those obtained by PCA, LDA, LDA-A, and BTM. Note that there is no significant difference between the results of LDA and LDA-A. This is because frequent locations where eliminated from the input. The same applies to BTM; users with small number of check-ins were eliminated from the dataset because they affect the performance of all models. Consequently, the results obtained using BTM did not outperform other

\(^5\)Note that BTM is only applied to RM data, as applying BTM on MDC data is extremely inefficient given the size of user-location check-ins in MDC data.
Figure 3.7: ROC curves of link prediction on MDC dataset, when using KNN classifier (a)-(c) and using matrix factorization in (d).

Equivalent results to BTM were obtained when using CRM-2, which is the same as BTM but only infers patterns from ordered pairs of visited locations. This means that the same results can be obtained with the additional benefit of reduced memory usage. Lastly, the results achieved from CRM* outperforms all including Feature Engineering (FE). This indicates that it is possible to achieve good results equivalent to using task-specific and human-based feature extraction methods, without the need of the actual human effort and (probably) the introduced error.

For better visual comparison of the results, Figure 3.7 shows the ROC curve for the results of feature-based classification using KNN (CN, AA, and Jacc), and matrix factorization on the MDC dataset. It is very clear that using only topology measures is not performing as good as FE and CRM.

For better visual comparison of the results, Figure 3.8 shows the social network
Figure 3.8: Visualization of Link Prediction results using Matrix Factorization on the social network $G_{predict}$. Nodes represent users with new links, where the numbers in the nodes indicate user IDs. Grey edges indicate new but unpredicted links, blue edges indicate properly predicted links by the current representation learning space, red edges highlight new links which are identified by the current feature space but not by original. (a) no added features (original); and others with representation learned from (b) FE; (c) PCA; (d) LDA; (e) LDA-A; and (f) CRM.
for $G_{predict}$ in the MDC dataset. Nodes represent users with new links, and edges indicate the newly formed links. Grey edges indicate links that were missed, blue edges are the properly predicted links using the baseline method (no added features), and red edges are the links successfully predicted by the newly added features. Note that the figure shows the results when the False Positive Rate (FPR) is 20%, and when using matrix factorization as the prediction methodology. From Figure 3.8(a), it is clear that using no additional features is inferior to the other results, where the number of grey edges exceeds the blue ones. The result of using FE in Figure 3.8(b) shows a significant improvement where the red edges in the graph show the properly predicted links, which were missed in the baseline approach. Figures 3.8(c), 3.8(d), and 3.8(e) show results for PCA, LDA and LDA-A. Each shows improvement over the initial approach. More importantly, Figure 3.8(f) shows the result of CRM, where the difference between the predicted links in CRM and FE becomes clear. CRM clearly outperforms FE and all other methods, which indicates that the learned representation space preserves the latent patterns in the original data. Thus, this indicates that the proposed generalized model is capable of learning compact representations for user-location data, where the new space can capture the important latent patterns in the original high dimensional feature space.

3.4 Discussion

The work presented in this chapter illustrates an example of a feature engineering-based approach where features are extracted with the objective of employing them in social link prediction. Comparing to representation learning using generic latent feature models, feature engineering is a task-dependent and time consuming process where deep insight and understanding about the data and task(s) under study is required in order to achieve satisfactory results. The conducted experiments in this chapter compared the results of link prediction obtained using feature engineering
with those obtained using four generic latent feature models: PCA, LDA, LDA-A, and BTM. The results clearly demonstrate the impact of various feature learning methods on link prediction results. Without doubts, feature engineering enables experts to embed their knowledge while designing feature sets for a specific task. However, the experiment results showed that generic latent feature models can achieve equivalent results to those obtained using feature engineering, yet without the additional cost of human effort involved in the process. More importantly, comparable results were achieved even though latent feature models did not utilize metadata (e.g., time of check-ins and features extracted from communication records) into the inference process, which were used in feature engineering.

A generalized version of BTM is proposed in this chapter, to better account for the characteristics of the data under study. Comparing to results obtained using feature engineering and generic latent feature models, the proposed model achieved superior results. This highlights the importance of designing special latent feature models, tailored to handle the attributes of user-location data and incorporate metadata for better inference results. The models proposed in the next three chapters are examples of such models. An earlier version of the work presented in this chapter was published in the Proceedings of the 29th ACM Symposium On Applied Computing (SAC 2014), and was presented in Gyeongju, Korea, on March 24 - 28, 2014 [96].
Chapter 4

Model I: Incorporating Timestamps

Time is important to understand the context behind users movements. Because of regularity in human movement, time is often used to add meaning to raw location data and infer the context behind users movements. For example, places often visited regularly during working hours on weekdays often denote work or school areas, while locations visited occasionally on weekends often denote leisure areas. In this dissertation, the use of anonymized user location data is advocated, where no semantic categories or geographical location are added to the learning process, to ensure the privacy of users. For this reason, it is important to add timestamps into the learning process to add some contexts to the raw anonymized user location data.

In this chapter, HuMoR is proposed, a model that infers patterns from traces of human mobility data with timestamps as side features. HuMoR is designed to infer latent patterns from location co-occurrences in users trajectories, and supplement it with side features to improve inference especially when learning from sparse data. More specifically, HuMoR extracts latent patterns from collections of sequences, where sequences represent subsets of users trajectories with common side features. HuMoR provides a general framework for adding metadata, as it conditions local mixture proportions on per-sequence side features. This approach gives an additional layer of flexibility to the proposed model while enriching the learning process.

This chapter is organized as follows. Sections 4.1 and 4.2 present details of the proposed model and the model parameter inference, respectively. Section 4.3 showcases evaluation results on two real datasets, and Section 4.4 draws concluding remarks.
4.1 The Proposed Model

Model Description. Human Mobility Representation (HuMoR) model, a mixed-membership model for inferring users and locations representations is proposed. HuMoR treats human mobility data as collections of statistically related users sequences, each containing a subset of users’ checked-in locations. As mentioned previously, the basic assumption behind any mixed-membership model denotes that there exists a set of latent patterns, i.e., global mixture components, underlying the data. In HuMoR, those global mixture components, i.e., patterns, uncover shared recurring patterns from sequences of locations co-visited by users with similar side features. Additionally, the model infers mixture proportions local to each sequence, at which global components occur. In HuMoR, global mixture components are distributions of patterns over locations, and thus provide means for computing the new location representations ($W$). The local mixture proportions, on the other hand, are distributions of sequences over patterns, which provide means for computing the new user representations ($H$).

The key elements behind the proposed model are described next,

1) A latent pattern, $k$, has a distribution over locations, represented by $w_k$, $k = 1...K$.

2) A sequence, $s$, is defined as a subset of users trajectories with shared side features. A sequence has a distribution over patterns, represented by $\theta_s$.

Given the variation in users check-in degree, it is important for models of human mobility to capture such imbalance. As a result, the concept of sequences is introduced to harness the knowledge embodied in trajectories of active users, i.e., users with very high check-in degree. Extracted sequences have comparable lengths, and thus are much easier to handle than trajectories with varied activity levels. By introducing the concept of sequences, formally defined next, the model learns from sequence-level location co-occurrence of all users.

Additionally, to handle the big variation in location check-in degree, the proposed
model incorporates an optimized asymmetric Dirichlet prior over the local mixture proportions, which allows drawing patterns with popular locations more often than others. Moreover, the proposed model utilizes side features by conditioning the asymmetric Dirichlet prior over them. This conditioning is intended to capture additional similarity between sequences, given features imposed by users (such as timestamps for example). That is, by imposing such conditioning, patterns not only capture locations appearing together, but also locations with the same side features, even if they did not co-occur in the same sequences. Since our model imposes no restriction on the data source of side features, incorporating cross-domain data classifies our model as an instance of data fusion approaches [97].

Model Specification. The proposed model is specified using three approaches: its generative probabilistic process, its graphical representation (see Figure 4.1), and its joint distribution. Notations specific to this proposed model are defined first.

Notations. Let \( s = \{c_1, \ldots, c_I\} \) be a sequence defined as a subset of user \( v \)'s location check-ins \( C_v \), where \( I \) denotes the length of the sequence. All elements in a sequence must share common side features, denoted as \( q \). For example, if time is considered as a side feature, \( s \) can be a sequence of locations occurring in the morning. Given user \( v \), \( s_v^{(j)} \) is user \( v \)'s \( j \)th sequence such that \( s_v^{(j)} \subset C_v \). \( S_v = \langle s_v^{(j)} \rangle_{j=1}^{I} \) is user \( v \)'s collection of sequences, and \( S = \langle S_v \rangle_{v=1}^{[V]} \) is the collection of sequences for all users.

Let \( q_s \) be a boolean vector encoding side features for sequence \( s \), where \( n \) is the length of the side feature vector. The length, \( n \), corresponds to \( 1+ \) the cardinality of the side feature, i.e., the number of categories describing the side feature. For example, if the side feature encodes gender with two possible categories, male and female, then \( n = 1 + 2 = 3 \) in this case. Additionally, if multiple side feature are included, the cardinality of the side features equals the total number of categories in all features.
The generative probabilistic process for HuMoR is:

1. For each pattern, \(k = 1 : K\)
   
   (a) Draw \(w_k \sim \text{Dirichlet}_{|P|}(\beta)\)
   
   (b) Draw \(\lambda_k \sim [\text{Normal}(0, \sigma^2_0 I), \text{Normal}(0, \sigma^2)]\)

2. For each sequence \(s \in S\)
   
   (a) Assign \(\alpha_s = \exp(q_s^T \Lambda)\)
   
   (b) Draw \(\theta_s \sim \text{Dirichlet}_K(\alpha_s)\)
   
   (c) For each location in \(s\), \(i = 1 : I\)
      
      i. Draw \(z \sim \text{Multinomial}(\theta_s)\)
      
      ii. Draw \(p_{s,i} \sim \text{Multinomial}(w_{k=z})\)

where \(\lambda_k\) is an \(n \times 1\) per-pattern vector. In which, \(n - 1\) elements are drawn from a per-pattern Gaussian distribution with variance \(\sigma^2_0\) and mean 0. The \(n\)th value, a.k.a., the default value of pattern \(k\), is sampled from a Gaussian distribution with \(\sigma^2\) and mean 0. To simplify notations, both variances are denoted as \(\sigma^2\). \(\lambda_k, k = 1...K\) can be packed column by column, resulting in an \(n \times K\) matrix denoted as \(\Lambda\). The parameter of the Dirichlet prior (i.e., the hyperparameter) on the per-pattern location distribution is denoted by \(\beta\). The hyperparameter on the per-sequence pattern distribution is denoted as \(\alpha_s\), which is a log-linear function of the sequence feature \(q_s\) and \(\Lambda\). The per-sequence pattern distribution drawn from \(\text{Dirichlet}(\alpha_s), \theta_s, s = 1...|S|\) can be packed row-by-row, resulting in a \(|S| \times K\) matrix denoted as \(\Theta\). The mixture assignment, \(z\), is a pattern drawn for location \(p_{s,i}\), i.e., the \(i\)th location in sequence \(s\). \(K\) is the number of patterns in the model. In this model, a check-in \(c\) including \(p\) and \(q\) are the only observed variables, all other variables are latent.

The generative process of HuMoR is explained as follows. For each sequence, a probability distribution over patterns is drawn, \(\theta_s\), from \(\text{Dirichlet}(\alpha_s)\), where \(\alpha_s\)
Figure 4.1: Graphical representation of HuMoR. Nodes represent random variables. Shaded nodes are observed and plates indicate repetition; $I$ is the length of a sequence, $|S|$ is the number of sequences extracted from the data, $p$ is a location drawn randomly from the Multinomial location distribution, parameterized by $w_k = z$ which is randomly drawn from Dirichlet($\beta$). $\theta$ is the pattern distribution drawn from Dirichlet($\alpha_s$), $\alpha_s$ is conditioned on the sequence features $q_s$ and $\Lambda$, where $\Lambda$ is randomly drawn from a normal distribution with variance $\sigma^2$.

is a function of the sequence feature $q_s$, and $\Lambda$. Then, for every element in the sequence, a mixture assignment is drawn, i.e., a pattern $z$, from the multinomial pattern distribution given the probabilities in $\theta_s$. Finally, a location is drawn from the multinomial location distribution given the probabilities in $w_{k=z}$.

The full joint distribution over the parameters in Figure 4.1 has a probability distribution given by the chain rule in the following Equation,

\[
P(\sigma^2, \Lambda, q, \alpha, \beta, W, \Theta, z, p) = P(\sigma^2)P(\Lambda|\sigma^2) \prod_{k=1}^{K} P(w_k|\beta) \\
\prod_{s=1}^{|S|} P(q)P(\alpha_s|q_s, \Lambda)P(\theta_s|\alpha_s) \prod_{i=1}^{I} P(z|\theta_s)P(p_{s,i}|w_{k=z}).
\]

(4.1)

**Illustrative Example.** Consider visiting time as a side feature and assume there
are three time ranges, each representing a unique portion of the day (e.g., morning, evening, and night). The cardinality of this feature space is 3, and thus the side features for sequence $s$, $q_s$, will be a vector of length $n = 1 + 3 = 4$. The side feature vector, $q_s$, encodes the time feature of sequence $s$, by assigning one in the proper time range and zero otherwise in addition to a pattern default value 1. As $\alpha_k = \exp(q_s^T \lambda_k), k = 1...K$ is the parameter of per-sequence pattern prior, the first $n-1$ elements in $\lambda_k$ associate sequences having the same value of time range together, even when sequences have different locations. The $n$th element in $\lambda_k$ is added to result in $\alpha_n$, which ultimately associates sequences having the same side features, with the same patterns. When the parameter $\sigma^2$ for drawing $\lambda_k$ is large, the default values for all patterns will fluctuate more freely, and thus distinguish patterns from each other.

**Model Comparison.** As mixed-membership models is an active research area in topic modeling, components of our proposed model are compared with models proposed to handle unconventional text corpora (i.e., collections of documents with special attributes). To handle long documents for example, [98] proposed dividing them into shorter pages, in which topics are more concentrated. The concept is similar to our introduced notion of sequences. However, the proposed model conditions segmenting users trajectories based on location side features, to ensure more them-specific subsets of the original long trajectories as well as to enrich the learning representation. Conditioning the hyperparameter of the local mixture proportions on side features was proposed in [99], which was utilized by the proposed model.

It is important to emphasize here that the proposed model imposes no restriction on the data source. Thus, incorporating cross-domain data classifies the proposed model as an instance of data fusion approaches [97]. This conditioning and optimizing of the hyperparameter serves the proposed model in two ways: 1) It incorporates side features into the learning process, and 2) It handles the power law distribution of locations check-ins. In text mining, the power law distribution of word usage known
as Zipf’s law in linguistics \[100\], is commonly treated by elimination of the most frequent (and often meaningless) stop words. This approach cannot be utilized for human mobility since frequently visited locations represent interesting (rather than meaningless) locations.

4.2 Estimating Model Parameters

In this section, description of the algorithm proposed for inferring the model parameters is provided. The general objective is to reverse the generative process of the proposed model and learn the posterior distribution, i.e, the distribution of the latent variables \((z, W, \Theta, \alpha, \Lambda)\) given the observed data is,

\[
P(\Lambda, \alpha, W, \Theta, z|p, q, \beta) = \frac{P(p, q, \Lambda, \alpha, W, \Theta, z|\beta)}{P(p, q|\beta)}. \tag{4.2}
\]

This distribution is intractable and cannot be computed exactly. Thus, the proposed inference model alternates between estimating the mixture assignments, \(z\), using collapsed Gibbs sampling from the posterior distribution (i.e., sampling pattern assignments) and optimizing \(\Lambda\) given \(z\) using the standard L-BGFS optimizer \[101\] in MALLET \[102\]. Values of \(\Lambda\) are then used in addition to the sequence side features to estimate \(\alpha\). Then, \(W\) and \(\Theta\) are estimated using Equation (4.5) and (4.7), respectively. Lastly, \(W\) is normalized and compute \(H\) using Equation (4.6) and (4.11), respectively. The complete inference algorithm is given in Algorithm 3.

4.2.1 Estimating the Mixture Assignments

Collapsed Gibbs sampling is an MCMC algorithm, which iteratively draws and updates one sample from the population given all other samples. In HuMoR, this implies sampling from the conditional distribution, \(P(z_i|z_{-i}, \mathcal{P})\), since \(z\) is a sufficient statistic.
Algorithm 3 Model I: Estimating Model Parameters

Input: $K, S, Q = \{q_1, \ldots, q_s\}, \sigma^2, \beta$

Output: $W, \Theta$

1: Initialize $\Lambda, z$, and update counters
2: for each iteration do
3: \hspace{1em} if mod(iteration, optimization interval) = 0 then
4: \hspace{2em} Optimize $\Lambda$
5: \hspace{1em} for each sequence $s \in S$ do
6: \hspace{2em} $\alpha_s = \exp(q^T_s \Lambda)$
7: \hspace{2em} for each location $i = 1 : I$ do
8: \hspace{3em} for each pattern $k = 1 : K$ do
9: \hspace{4em} $P(z_i = k|z^{(-i)}, P) \propto (\eta_{k,s}^{(-i)} + \alpha_k) \cdot \frac{\eta^0_{p,k} + \beta}{\eta^{(-i)}_{k} + \beta_0}$
10: \hspace{2em} Sample $k \sim P(z|P)$
11: \hspace{1em} Compute $W$ and $\Theta$ where $w'_{p,k} = \frac{\eta_{p,k} + \beta}{\eta_{k,s} + \beta_0}$ and $\theta_{k,s} = \frac{\eta_{k,s} + \alpha_k}{I + \alpha_s}$
12: \hspace{1em} Renormalize $W$, where $w_{p,k} : \frac{w'_{p,k}}{\sum_{k'} w'_{p,k'}}$
13: \hspace{1em} Compute $H$, where $h_{v,k} = \sum_{s} |S| \left( \frac{\prod_v w_{p,k} \theta_{k,s}}{\sum_{k'} (\prod_v w_{p,k'}) \theta_{k',s}} \prod_p |P| \eta_{p,v} \eta_{.,v} \right)$

for $W$ and $\Theta$. Sampling is given by,

\[ P(z_i = k|z^{(-i)}, P) \propto (\eta_{k,s}^{(-i)} + \alpha_k) \cdot \frac{\eta^0_{p,k} + \beta}{\eta^{(-i)}_{k} + \beta_0}. \tag{4.3} \]

where $\eta_{k,s}$ is the number of times pattern $k$ was drawn in sequence $s$, superscript $(-i)$ is used when element $i$ is excluded, $\eta_{p,k}$ is the number of times location $p$ was assigned to pattern $k$, $\eta_{.,k} = \{\eta_{p,j,k}\}_{j=1}^{P}$ is the total number of locations assigned to pattern $k$, and $\beta_0 = \sum_{p=1}^{P} \beta$.

HuMoR’s complexity, using collapsed Gibbs sampling, is $O(K \cdot |S| \cdot l_s)$, which is comparable to LDA’s complexity, when using the same inference approach, $(O(K \cdot |V| \cdot l_v))$, since $|S| \approx (|V| \cdot l_v)/l_s$, where $l_s$ and $l_v$ are the average lengths of sequence and user trajectories, respectively.
4.2.2 Computing the New Representations

The location distribution of each pattern can be estimated as:
\[ P(w|\mathcal{S}, \beta) = \frac{1}{Z_w} \text{Dirichlet}(w|\eta_k + \beta), \]
where \( Z_w \) is a normalization factor and \( \eta_k = (\eta_{1,k}, \cdots, \eta_{|\mathcal{P}|,k}) \). Since the expectation of Dirichlet \((x|\vartheta)\) is,
\[ E(x_i) = \frac{\vartheta_i}{\sum_{k=1}^{K} \vartheta_k}, \tag{4.4} \]
each element in the matrix \( W \) is estimated using,
\[ w_{p,k} = \frac{\eta_{p,k} + \beta}{\eta_{p,k} + \beta_0}. \tag{4.5} \]

Additionally, \( W \) is renormalized to get the location over patterns distribution, using,
\[ w_{p,k} : \frac{w_{p,k}}{\sum_{k'} w_{p,k'}} \tag{4.6} \]

The pattern distribution is given by \( P(\theta|z, \alpha) = \frac{1}{Z_\theta} \text{Dirichlet}(\theta|\eta_s + \alpha) \), where \( Z_\theta \) is a normalization factor and \( \eta_s = (\eta_{1,s}, \cdots, \eta_{k,s}) \). Again, given Equation (4.4), the \( k \)th element in the pattern distribution vector \( \theta \) is given by,
\[ \theta_{k,s} = \frac{\eta_{k,s} + \alpha_{k,s}}{I + \alpha_0,s}. \tag{4.7} \]

As HuMoR does not model the full user trajectory generation process. Details of how to estimate the per-user pattern distribution, \( P(z|v) \) is provided next. Let user \( v \in \mathcal{V} \) has \( |\mathcal{S}_v| \) number of sequences, after learning per-sequence patterns for all users (\( \theta \) in Eq. (4.7)), \( P(z|v) \) is approximated using the chain rule,
\[ P(k = z|v) = \sum_{s} P(z, s|v) = \sum_{s} P(z|s, v)P(s|v). \tag{4.8} \]
\( P(s|v) \) can be estimated empirically from the data,
\[
P(s|v) = \prod_{i=1}^{n} P(p_i|v) = \prod_{p} \frac{\eta_{p,v}}{\sum_{p'} \eta_{p',v}} = \prod_{p} \frac{\eta_{p,v}}{\eta_{,v}}
\]

where \( \eta_{p,v} \) is the frequency of location \( p \) in user \( v \), and \( |\mathcal{P}_s| \) and \( |\mathcal{P}_v| \) are the number of unique locations in sequence \( s \) and user \( v \), respectively.

Let pattern \( z \) and user \( v \) be conditionally independent given sequence \( s \), \( P(z|s,v) = P(z|s) \). Using Bayes’ formula and the parameters inferred by HuMoR, \( P(z|s) \) is estimated by,
\[
P(k = z|s) = \frac{P(s|z)P(z)}{P(s)} = \frac{\prod_{i=1}^{n} w_{p_i,k} \theta_{k,s}}{\sum_{k'}(\prod_{i} w_{p_i,k'}) \theta_{k',s}}.
\]

Therefore, taking Equations (4.9) and (4.10) into (4.8),
\[
P(k = z|v) = \sum_{s} \left( \frac{\prod_{i=1}^{n} w_{p_i,k} \theta_{k,s}}{\sum_{k'}(\prod_{i} w_{p_i,k'}) \theta_{k',s}} \frac{|\mathcal{P}_s|}{\eta_{,v}} \right),
\]

which represents the per-user pattern distribution, \( h_{v,k} \).

For new users, the per-user pattern distribution can be estimated using
\[
P(k = z|v) = \sum_{c} \frac{w_{p,v}}{\sum_{k'} w_{p,k'}} \frac{\eta_{p,v}}{\eta_{,v}},
\]

which is simpler and more efficient than the folding in approach and produces comparable results.

### 4.3 Evaluation

**Datasets.** Evaluation is conducted on two publicly available datasets: MDC and GW. Table 4.1 shows summary statistics of subsets of the data used. Details of the
Table 4.1: Datasets Statistics. No. of Active Users (|V|), No. of Social Links (|A_{nnz}|), No. of Checkins (|C|), Average Check-ins Per User (CpU) and Average Check-ins Per Location (CpL).

|    | |V| | A_{nnz} | |C| | CpU | CpL |
|----|---|---|---------|---|---|----|----|
| MDC | 107 | 197 | 1,270,045 | 11870 | 25 |
| GW  | 43,956 | 222,319 | 2,887,700 | 65 | 29 |

MDC dataset was provided in Section 3.3.

The GoWalla (GW) dataset was collected by the Stanford Network Analysis Project (SNAP) from a LBSN with the same name, using public Application Program Interface (API). The dataset contains individuals check-in data as well as their undirected online social links. Check-in data include timestamp, latitude, longitude and location ID for each checked-in location per individual. GoWalla (GW) data were collected between 2009 to 2010, with a total of 196,591 users, 6,442,890 check-ins, and 950,327 online social links. GW is a LBSN, where visited locations are reported by users, and friendships in the network are represented as undirected links in the social graph. Comparing to GW, MDC contains more visited locations per user. This is simply because visits are collected automatically by mobile carriers as opposed to LBSN where users manually checkin to locations.

**Experimental Settings.** The objective here is to evaluate the proposed model and verify that the new representation preserves the latent patterns in the original raw feature space. Comparison is conducted between the representations learned by HuMoR, with those inferred using FE, PCA, LDA, and LDA-A. Evaluation of the performance is done on a subset of the original data, summarized in Table 4.1. This is because in preprocessing, users and locations with low number of check-ins (less than 5) are eliminated from either or both datasets in an iterative manner, as neither the proposed model nor the baselines can learn proper latent patterns from very short user location data.

PCA, LDA and LDA-A are applied on the same user × location frequency matrix.
Figure 4.2: Visualization of per-sequence and per-user pattern distributions.

All graphical models are trained for 1000 iterations, and parameters are optimized every 50 iterations, after an initial burn-in period of 200. The common parameter $\beta$ for all graphical models is set to be $\beta = 0.01$. Setting $\beta \ll 1$ generates sparse pattern representations, and thus increases patterns separability. For HuMoR, time range is used as a sequence feature, where day is divided into four equal intervals. Different intervals for time range have been tested, and the results obtained were robust against small variations in the settings of this feature.

The first $n - 1$ elements in $\lambda$, associated with sequence features, are drawn from a Gaussian distribution with $\sigma^2 = 0.5$. The $n$th element, associated with the pattern default value, is drawn from a Gaussian distribution with $\sigma^2 = 100$. Collectively, the value of $\lambda$ along with the sequence features determine $\alpha_s$, which in turn controls the per-sequence pattern distribution. Sequences with similar $\alpha_s$’s, will have similar pattern distributions. Thus, setting a large variance for the pattern default value enables patterns to be easily distinguished from each other. The sensitivity of the model to $\sigma^2$ is investigated, and results are reported later in this section. Regarding the number of patterns $K$, the model’s sensitivity to it is investigated after comparing the performance of different models, where $K$ is set to 5.

**Visualization of the Inferred Patterns.** Figure 4.2 shows a heatmap of the new representation learned by HuMoR per sequence ($\theta_s$, Figure 4.2(a)) for 2 selected
users and per user ($h_v$, Figure 4.2(b)) for 950 users in GW. The new representation is mostly sparse, i.e., sequences (and users) are associated with only few patterns, given the setting of $\beta$. For example, in 4.2(a) user A has sequences with 2 distinct patterns, while user B has only 1 pattern. This sparse representation of users can instantly be used to group them into clusters of users with similar mobility patterns.

**Link Prediction.** The new user representation spaces are utilized in link prediction using feature-based classification and supervised matrix factorization. For details on each approach please refer to Section 2.1.2.2, and for a general overview of link prediction please refer to Section 2.1.2.

Table 4.2 shows the AUC averaged over 10 runs when adding five different representation produced by FE, PCA, LDA, LDA-A and HuMoR to three different topology measures (CN, AA, and Jacc). Values under *Representations* indicate the mean AUC values obtained when the new representations used in conjunction with pairs of baseline measures and prediction approaches (in rows). MF indicates supervised Matrix Factorization without added representations and MF-w indicate Matrix Factorization with the inferred representations, where MF is used as a baseline measure for comparison.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Baseline Measures</th>
<th>Methods</th>
<th>Representations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>FE</td>
<td>PCA</td>
</tr>
<tr>
<td>MDC</td>
<td>CN 0.645 Trees</td>
<td>0.782</td>
<td>0.768</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>0.859</td>
<td>0.831</td>
</tr>
<tr>
<td></td>
<td>AA 0.645 Trees</td>
<td>0.779</td>
<td>0.759</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>0.861</td>
<td>0.831</td>
</tr>
<tr>
<td></td>
<td>Jacc 0.645 Trees</td>
<td>0.845</td>
<td>0.831</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>0.779</td>
<td>0.769</td>
</tr>
<tr>
<td></td>
<td>MF 0.820 MF-w</td>
<td>0.873</td>
<td>0.893</td>
</tr>
<tr>
<td>GW</td>
<td>CN 0.892 Trees</td>
<td>0.778</td>
<td>0.793</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>0.727</td>
<td>0.689</td>
</tr>
<tr>
<td></td>
<td>AA 0.901 Trees</td>
<td>0.876</td>
<td>0.875</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>0.934</td>
<td>0.938</td>
</tr>
<tr>
<td></td>
<td>Jacc 0.832 Trees</td>
<td>0.772</td>
<td>0.789</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>0.721</td>
<td>0.774</td>
</tr>
<tr>
<td></td>
<td>MF 0.906 MF-w</td>
<td>0.922</td>
<td>0.942</td>
</tr>
</tbody>
</table>
Table 4.2 shows that the overall best result in each dataset is achieved by HuMoR. The overall best result in MDC is achieved by Matrix Factorization (MF) with representations from HuMoR. In GW, the best result is achieved by KNN with representations from HuMoR added to AA. In both datasets when comparing to the column of Baseline Measures, it can be easily observed that the new representation learned by HuMoR always improves the results significantly and outperforms all other representation learning methods. This is not the case for other baseline methods, especially in the GW dataset, as they even worsen the results because of the inappropriately added representations.

When comparing the results achieved using HuMoR on both datasets, it is observed that profound improvement is obtained in the GW dataset. Recall that GW is a LBSN dataset with a large number of inactive users, who mostly have sparse location traces. As HuMoR captures patterns from collective sequences of all individuals in the dataset, the sparsity of location traces does not impact the model’s performance as it does with mainstream topic models (LDA and LDA-A). For the same reason, the CRM model, proposed in the previous chapter, could not be applied to the GW dataset.

Parameter Sensitivity. The sensitivity of HuMoR to the setting of the number of patterns \((K)\) and to the pattern variance \((\sigma^2)\) in Figure 4.3 is investigated. Figure 4.3(a) shows the AUC results for both datasets when varying the value of \(K\). For the MDC data, prediction results degrade when the number of patterns increases over 5. This is not the case with the GW data as no significant change is observed. This can be attributed to the characteristical difference between the two datasets: MDC, a dense user-location dataset, and GW, a sparse user-location dataset. Two interpretations hold in this case: 1) MDC has a small number of users for which a small \(K\) is sufficient for capturing the latent patterns, and 2) MDC has dense (thus more informative) trajectories per users, where larger \(K\) leads to learning more refined and
user-specific topics, which do not contribute significantly to a task like link prediction where emphasis is placed on homophily of pairs and not individual differences.

Figure 4.3(b) shows the AUC results for both datasets when varying the value of $\sigma^2$. It shows that the model is robust to the setting of $\sigma^2$ in both datasets.

![Graph](image)

(a) to Number of Patterns

(b) to Pattern Variance

Figure 4.3: Sensitivity analysis of HuMoR parameters.

### 4.4 Discussion

HuMoR, a mixed-membership model, is proposed to infer latent patterns from traces of location IDs where location check-ins follow a power-law distribution. Unlike existing mobility models, HuMoR is capable of learning patterns from users with different activity levels while incorporating side features into the learning process. In theory, the proposed model allows utilizing multiple side features at the same time, which enables learning patterns over multiple time-scales. However, in practice, conditioning sequencing on multiple side features will result in many very-short trajectories where the model cannot learn from. Additionally, the efficiency of the model is limited by Gibbs sampling which prevent utilizing it in real large scale user-location data. In the next chapter, the objective is to propose a model that incorporates multiple side features into the learning process as well as propose an efficient inference algorithm.
The work presented in this chapter was published in the *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI 2016)*, and was presented in Phoenix, Arizona, on February 12-17, 2016 [73].
Chapter 5

Model II: Incorporating Multiple Time-Scales

Individuals movements have patterns at different time scales, ranging from daily, weekly and monthly patterns. For example, a daily pattern of a student in a weekday during school year is significantly different from its counterpart during holidays. Consequently, it is important to model time at different scales to capture the distribution of a latent pattern at different granularities, i.e., over hours of a day, days of a week, months of a year.

Existing studies [73, 71, 72, 74, 75] fail to efficiently incorporate multiple time scales into the learning model. They either learn simplified temporal patterns without considering the regularities of human movements over multiple time-scales, or adopt an expensive approach that limits the use of such models in real large-scale data. In this chapter, a general and efficient latent feature model for uncovering human mobility patterns from anonymized locations is proposed, while simultaneously considering temporal regularities over multiple time scales. Evaluation is done on two real datasets, which shows the superiority of the proposed approach when compared to baseline models.

The rest of this chapter is organized as follows. Sections 5.1 and 5.2 present details of the proposed model and the model parameter inference, respectively. Section 5.3 showcases evaluation results, and Section 5.4 draws concluding remarks.
5.1 The Proposed Model

Model Description. Multi-Time Scales Mobility Representation (M-Times) model is proposed in this chapter. M-Times is a mixed-membership model, where location data are grouped by users (collections of user check-ins), and each user is modeled by latent patterns. Latent patterns, a.k.a., global mixture components, are shared across users, yet each user has different mixture proportions. The proposed model infers those latent patterns, referred here as mobility patterns, and users’ mixture proportions from collections of spatial and temporal check-ins, where time at different scales is considered.

Model Specification. Figure 5.1 illustrates the graphical representation of M-Times. The generative process is described as: For each user, a probability distribution for mobility patterns, $h_v$, is drawn from $\text{Dirichlet} (\alpha)$. Then, for every check-in, a mixture assignment, i.e., a mobility pattern $z$, is drawn from the multinomial distribution parameterized by $h_v$. Finally, a location $p$ is drawn from the multinomial location distribution, parameterized by $w_k$, and a time $t^{(g)}$ is drawn (for every time-scale $g$) from the multinomial time distribution, parameterized by $\omega_k^{(g)}$. In this model, $p$ and $t$ are the only observed variables, $K, G, u,$ and $n$ are user-set parameters, and all other variables are latent. Formally, the generative process of the model is:

1. For each latent pattern, $k = 1 : K$
   
   (a) Draw $w_k \sim \text{Dirichlet}_{|P|}(\beta)$
   
   (b) For each time scale, $g = 1 : G$
       
       i. Draw $\omega_k^{(g)} \sim \text{Dirichlet}_{|T^{(g)}|}(\delta)$

2. Draw $\alpha \sim \text{Gamma}(u, n)$

3. For each user $v \in V$
   
   (a) Draw $h_v \sim \text{Dirichlet}_K (\alpha)$
(b) For each check-in $c \in C_v$

i. Draw $z \sim \text{Multinomial}(h_v)$

ii. Draw $p \sim \text{Multinomial}(w_k=z)$

iii. For each time scale, $g = 1 : G$

A. Draw $t^{(g)} \sim \text{Multinomial}(\omega^{(g)}_{k=z})$

**Illustrative Example.** To better understand the proposed model, the following illustrative example is provided. First, each of the uncovered global mixture components (latent patterns) governs where and when users check-in and captures a semantic/purpose. For example, one component is about ‘entertainment’ and another is about ‘work’. The ‘entertainment’ component is illustrated by the location distribution with peaks at entertainment venues such as theaters and concert venues, and the time distribution over hours with peaks after working hours and during weekends. For the ‘work’ component, the distribution will peak at office venues during the starting of working hours.

Second, a user’s check-ins are generated by having a mixture of these components (conducting activities with different semantics/purposes). For example, a user went out for an ‘entertainment’ purpose will highly likely visit a location peaking (e.g., theaters) in the location distribution at a time (e.g., 9pm) peaking in time distribution
of the ‘entertainment’ component. Users have different mixtures of latent patterns that are interesting to him/her. User profiling is to characterize them by the latent semantics/purposes.

**Modeling Time.** M-Times provides a flexible approach for incorporating multi-scale time into the learning process via a Bayesian approach, where a Dirichlet prior is introduced over the time Multinomial\(^1\). Thus it avoids overfitting the model to temporal data, and adds minimum additional cost. Additionally, using this approach, latent temporal patterns can be captured at different granularities/time-scales, i.e., daily patterns, weekly patterns, monthly patterns, and so on. The granularity of the temporal pattern is defined by the user of the model, and it generally depends on what patterns the user wants to capture from the data. For example, it is possible to only use the ‘year’ of checked-in location (in a data set collected over many years) if one is interested in capturing high-level changes of patterns over prolonged periods of time. Additionally, the model learns from any combination of temporal granularities, e.g., daily/weekly patterns, daily/monthly patterns, daily/yearly patterns, etc.

**Symmetric and Asymmetric Dirichlet hyperparameters.** The choice of symmetric and asymmetric hyperparameters for the Dirichlet priors is critical to learning meaningful latent mobility patterns. In order to handle the big variation in location check-in degree, the Dirichlet prior over the local user mixture proportions should have an asymmetric hyperparameter ($\alpha$). This setting will tell the model that some latent patterns are more frequent than others, and thus those patterns will eventually capture all the frequent locations, leaving other patterns free to capture co-occurrences of more-interesting and less-frequent location check-ins. Note that adopting an asymmetric $\alpha$ is not only beneficial for handling the large variation in location popularity, as it also plays an important role in minimizing the sensitivity of

---

\(^1\)Note that because a multi-scale approach over time is adopted, time at any scale is a discrete value. For example, time at an hour-of-day scale is a discrete number denoting the hour of the day (0, 1, 2, ..., 23), and time at a day-of-week scale is a discrete number ranging from (1, ..., 7).
the model to the number of mixture components preset by the user, by maximizing the stability of the inferred patterns. This is because by using an asymmetric $\alpha$, pattern usage will not be uniform, and thus increasing the value of $K$ will have insignificant impact on the main patterns. Although for small values of $K$, which do not capture all the hidden patterns in the raw data, the model will still be sensitive to this user set parameter.

As for the Dirichlet priors over the global location and time mixture components, utilizing a symmetric hyperparameter is a much suitable option. Consider the case of location mixture components for example. Using a symmetric hyperparameter ($\beta$) will tell the model that all locations have equal chance of occurring, and thus there will be no preference for popular locations. However, if an asymmetric hyperparameter is used instead, the model will favor popular locations and mixture components will eventually capture the empirical location check-in distribution. This is undesirable as one is interested in capturing latent patterns in the data and not the global patterns dominating the data. The same discussion applies to the time mixture components and thus a symmetric hyperparameter ($\delta$) is also utilized.

**Model Comparison.** As mentioned earlier, mixed-membership models are of wide popularity in text mining (generally known as topic models). There are approaches incorporating the temporal patterns while uncovering ‘static’ topics from text corpora. The approach proposed by Masada et. al. [80] incorporates time using a Bayesian approach. However, time is not considered as a multi-scale value, but rather as a single timestamp. Wang et. al. [81] proposed a method named TOT, in which time was modeled using a Beta distribution. npTOT [82] and TONPT [83] are two models attempting to overcome the limitations of TOT imposed by the unimodal distribution. However, both models introduce biases with the new introduced parameters, as well as complexity, often at the cost of the model’s speed. Additionally, both models treat time as a single continuous value and thus ignore the mobility patterns
over multiple time scale. Moreover, in all models that assume a continuous time, time is initially normalized between 0 and 1. However, details occurring over a short period of time are lost when normalizing.

In this proposed model, time is modeled as a discrete multi-scale value, with a Dirichlet prior over each time scale. Users of the model can control the level of details required in the temporal patterns by setting time scales differently. The model avoids overfitting to the temporal patterns due to the imposed Dirichlet prior.

5.2 Estimating Model Parameters

In this section, an efficient (and easy-to-parallelize) inference algorithm is proposed for the latent variables in M-Times. Note that $z$ is a sufficient statistic for $H, W, \Omega$. Thus, the objective reduces to inferring only four latent variables, $z, \alpha, \beta, \delta$, i.e., the mixture assignments and the hyperparameters. The remaining variables, i.e., new representations, $H, W, \Omega$ can be later computed using the inferred parameters. To achieve this goal, a two-step inference algorithm is proposed, which alternates between the following:

- Estimating the mixture assignments, $z$, using the current values of the hyperparameters (detailed in Section 5.2.1).

- Optimizing the hyperparameters $\alpha, \beta, \delta$ using the current values of the mixture assignments (detailed in Section 5.2.2).

Finally, the new representation spaces for users, $H$, locations, $W$, and time, $\Omega$, are computed given the inferred parameters (detailed in Section 5.2.3).
5.2.1 Estimating the Mixture Assignments

The posterior distribution of M-Times, given by,

$$P(z, H, W, \Omega, \alpha, \beta, \delta | C) = \frac{P(C, z, W, H, \Omega, \alpha, \beta, \delta)}{P(C, \beta, \delta)}, \quad (5.1)$$

is intractable. Hence, there is no closed-form (i.e., exact) solution to the inference of $z$. A variety of approximate methods are available for posterior maximization, including collapsed Gibbs sampling. In the basic form of collapsed Gibbs sampling, a mixture assignment ($z$) is iteratively drawn for sample $i$ from the conditional distribution, given all current assignments except $i$, $z^{(-i)}$, until convergence is reached.

The conditional distribution\(^2\) of M-Times is given by,

$$P(z = k | z^{(-i)}, C) \propto \frac{\eta_k^{(-i)} + \alpha_k}{\eta_k^{(-i)} + \alpha_0} \cdot \frac{\eta_p^{(-i)} + \beta}{\eta_p^{(-i)} + \beta_0} \cdot \prod_g \frac{\eta_t^{(g)(-i)} + \delta_t^{(g)}}{\eta_t^{(g)(-i)} + \delta_0^{(g)}} \quad (5.2)$$

where $\eta_{k,v}$ is the number of times the latent pattern $k$ was drawn for user $v$, superscript $(-i)$ is used when element $i$ is excluded, $\eta_{p,k}$ is the number of times location $p$ was assigned to latent pattern $k$, $\eta_{t,g,k}$ is the number of times time $t$ at scale $g$ was assigned to latent pattern $k$, $\eta_{t,k}$ is the total number of check-ins assigned to latent pattern $k$, $\alpha_0 = \sum_{k=1}^{K} \alpha_k$, $\beta_0 = \sum_{p}^{P} \beta$, and $\delta_0^{(g)} = \sum_{t}^{T} \delta^{(g)}$. Note that the superscript $(-i)$ is eliminated hereafter for simplicity.

Sampling from this conditional distribution requires $O(K)$ time, forming the main bottleneck in this inference algorithm. Proposing an efficient inference algorithm is critical due to the large volume of user-location data, which makes using simple inference algorithms (such as collapsed Gibbs sampling) impractical. In order to improve the efficiency of the inference algorithm, a key task is to reduce the sampling com-

\(^2\)Note that for the purpose of sampling, it is not necessary to normalize the first term by the user check-in length as this value remains constant across patterns.
Algorithm 4 Estimating Mixture Assignments

Input: \( C, \alpha, \beta, \delta \)

Output: \( z \)

Initialize randomly: mixture assignments and proposals

for each iteration do

for each location \( p \in P \) do

for each location check-in do

Accept user proposals,

\[
\pi_{k \rightarrow k'}^{\text{user}} = \min \left\{ 1, B \cdot \frac{(\eta_{p,k'} + \beta)}{(\eta_{p,k} + \beta)} \right\}
\]

for each location check-in do

Draw from the spatial proposal,

\[
q_{\text{spatial}}(z = k) = (\eta_{p,k} + \beta)
\]

Build alias tables

for each user \( v \in V \) do

for each user check-in do

Accept spatial proposals,

\[
\pi_{k \rightarrow k'}^{\text{spatial}} = \min \left\{ 1, B \cdot \frac{(\eta_{k',v} + \alpha_{k'})}{(\eta_{k,v} + \alpha_{k})} \cdot \prod_{g=1}^{G} \frac{(\eta_{v,k,g} + \delta_{g})}{(\eta_{v,k,g} + \delta_{g})} \right\}
\]

for each user check-in do

Draw from the user proposal,

\[
q_{\text{user}}(z = k) = (\eta_{k,v} + \alpha_{k}) \cdot \prod_{g=1}^{G} \frac{(\eta_{v,k,g} + \delta_{g})}{(\eta_{v,k,g} + \delta_{g})}
\]

where \( B = \frac{(\eta_{v} + \delta_{0})}{(\eta_{v} + \delta_{0})} \cdot \prod_{g=1}^{G} \frac{(\eta_{v} + \delta_{g})}{(\eta_{v} + \delta_{g})} \).

plexity. Inspired by [103], the proposed inference algorithm reduces the complexity by utilizing four key elements: Metropolis-Hasting, AliasSampling, delayed updates, and cache locality. The proposed algorithm is presented in Algorithm 4 and details are provided next.

Metropolis-Hasting is an MCMC algorithm that allows sampling from a proposal distribution, \( q(\hat{x}|x) \), when the true distribution, \( p(x) \), is difficult to sample from. For the unnormalized conditional distribution of M-Times, two proposal distributions are proposed. The spatial and user/temporal proposals, which are given by the following equations respectively,

\[
q_{\text{spatial}}(z = k) = (\eta_{p,k} + \beta), \tag{5.3}
\]

\[
q_{\text{user}}(z = k) = (\eta_{k,v} + \alpha_{k}) \cdot \prod_{g=1}^{G} (\eta_{v,k,g} + \delta_{g}). \tag{5.4}
\]

These proposal distributions were chosen carefully to reduce the complexity and
overhead of sampling, which will be detailed next. For now, note that samples are drawn from the two proposals iteratively to maximize the mixing coefficient and obtain faster convergence. Drawn samples are accepted with an acceptance probability, \( \pi \), proportional to the true distribution, such that

\[
\pi = \min\left\{ 1, \frac{p(\tilde{x})q(x|\tilde{x})}{p(x)q(\tilde{x}|x)} \right\}.
\]

The acceptance probability for the spatial and user temporal proposals are given by the following equations respectively,

\[
\pi_{spatial}^{k \rightarrow k'} = \min\left\{ 1, B \cdot \frac{(\eta_{k',v} + \alpha_{k'})}{(\eta_k + \alpha_k)} \cdot \prod_{g=1}^{G} \frac{(\eta_{t,k'}^{(g)} + \delta^{(g)})}{(\eta_{t,k}^{(g)} + \delta^{(g)})} \right\},
\]

\[
\pi_{user}^{k \rightarrow k'} = \min\left\{ 1, B \cdot \frac{(\eta_{p,k'} + \beta)}{(\eta_{p,k} + \beta)} \right\},
\]

where \( B = \frac{(\eta_k + \beta_0)}{(\eta_k + \beta_0)} \cdot \prod_{g=1}^{G} \frac{(\eta_{t,k}^{(g)} + \delta^{(g)})}{(\eta_{t,k}^{(g)} + \delta^{(g)})}. \)

At this point, the original conditional distribution is reduced to two proposal distributions. Next, description of how to sample from each proposal efficiently is introduced.

**The spatial proposal.** The spatial distribution is a mixture of multinomials. To sample efficiently from this distribution, one can utilize a two-step ancestral sampler as follows,

\[
\text{Step 1: Draw } u \sim \text{Bernoulli}(\frac{\sum_k \eta_{p,k}}{\sum_k \eta_{p,k} + \sum_k \beta})
\]

\[
\text{Step 2: } z \sim \begin{cases} 
\frac{\eta_{p,k}}{\sum_k \eta_{p,k}} & \text{if } u = 1 \\
\frac{\beta}{\sum_k \beta} & \text{if } u = 0
\end{cases}
\]

Note that sampling from the first case in (5.8) can be achieved in \( O(1) \) time using random positioning, and sampling from the second case in (5.8) can also be achieved in \( O(1) \) time since it is equivalent to sampling from a uniform distribution, Uniform(\( K \)), as \( \beta \) is symmetric.

**The user proposal.** In order to sample from the user proposal efficiently, the
formula is reorganized to be,

\[ q^{\text{user}}(z = k) = \eta_{k,v} \cdot \prod_{g=1}^{G} (\eta_{t,k}^{(g)} + \delta^{(g)}) + \alpha_k \cdot \prod_{g=1}^{G} (\eta_{t,k}^{(g)} + \delta^{(g)}) \cdot \] 

(5.9)

Based on this formula, one can utilize an ancestral sampling as in the previous proposal, where one can sample from the first term with success probability of,

\[
\frac{\sum_k \eta_{k,v} \cdot \prod_{g=1}^{G} (\eta_{t,k}^{(g)} + \delta^{(g)})}{\sum_k \eta_{k,v} \cdot \prod_{g=1}^{G} (\eta_{t,k}^{(g)} + \delta^{(g)}) + \sum_k \alpha_k \cdot \prod_{g=1}^{G} (\eta_{t,k}^{(g)} + \delta^{(g)})},
\] 

(5.10)

and from the second otherwise.

The first term is sparse in \( \eta_{k,v} \), which means that one can sample from it in \( O(K_v) \ll O(K) \) where \( K_v \) is the number of non-zero patterns in user \( v \). From initial experiments, it was found that \( K_v \) tends to be very small, which is logical since users tend to be associated with few patterns only. The second term, on the other hand, is dense in \( \alpha_k \), which means that direct sampling takes \( O(K) \). However, note that the second term is independent on user check-ins, and is a function of the hyperparameters and the time metadata. Additionally, note that time values, at each scale, has a small cardinality, e.g., 24 hours in a day and 7 days in a week. Given these two observations, the alias sampling method is utilized to sample from this dense distribution in \( O(1) \) time, after an initial period of \( O(K) \) for constructing an alias table for every feature combination. That is, an alias table is built for each tuple of feature values. Given that the updates of mixture assignments are delayed, alias sampling can be successfully adopted since the distribution is stale.

Finally, the cache locality of the proposed inference algorithm is improved by making use of the common terms between the spatial and user proposals and acceptance

\[ \text{For example, hour 2, weekday 1, and month 4 has one alias table. Every other combination has a distinct alias table. Alias tables are built/updated once per iteration and used to sample for all users check-ins.} \]
rate. Observe that the user proposal and the spatial acceptance rate both access the high dimensional (and sparse when $K$ is large) count matrix, $\eta_{k,v}$, while the spatial proposal and the user acceptance rate both access $\eta_{p,k}$, a high dimensional and sparse count matrix as well. In order to reduce cache misses, user proposal and spatial acceptance rate are grouped together in one phase. Additionally, spatial proposal and user acceptance rate are grouped together in another phase. Again, given the delayed update mechanism employed earlier, this grouping is possible and has no additional degrading impact on convergence.

5.2.2 Optimizing the Hyperparameters

The hyperparameters of any mixed membership model has a critical role in defining the sparsity and mutual exclusiveness of the mixture proportions [104]. One can set the hyperparameters based on prior beliefs (or assumptions) about the latent space. However, this approach is not feasible for this proposed model, given that the proposed model has $(G+1)$ symmetric hyperparameters and one asymmetric hyperparameter to tune. A better alternative is to optimize the hyperparameters given the observed data, in order to automate the process and reduce the human effort involved.

An efficient and easy-to-parallelize variant of Minka’s fixed point iteration [105,104] is utilized for optimizing the hyperparameters of the Dirichlet-multinomial distributions (a.k.a., Polya Distribution), $\alpha$, $\beta$, and $\delta$. First, description of the process of optimizing the asymmetric hyperparameter is provided. Then, the update equations for the symmetric case is briefly presented as they follow the same method.

The asymmetric hyperparameter, $\alpha$. The hyperparameter of the local mixture proportions, $\alpha_k$, is assumed to be independently drawn from the Gamma distri-
bution with two (hyperhyper)parameters \((n, u)\),

\[
P(\alpha_k|n, u_k) = \frac{1}{\Gamma(u_k)n} (\frac{\alpha_k}{n})^{u_k-1} e^{\frac{-\alpha_k}{n}}.
\] (5.11)

\(\alpha\) can be estimated by maximizing the posterior probability, \(P(\alpha|C) \propto P(C|\alpha)P(\alpha)\), where the likelihood \(P(C|\alpha)\) and the hyperprior \(P(\alpha)\) can be transformed into log forms. The summation of log-likelihood and log-prior has a lower bound, which can be used to find \(\alpha_k^*\) maximizing \(P(\alpha|C)\) [104]. The fixed-point iteration runs repeatedly two equations below until convergence is reached.

\[
[\alpha_k]^{new} = [\alpha_k]^{old} \sum_v \left[ \Psi(\eta_{k,v} + [\alpha_k]^{old}) - \Psi([\alpha_k]^{old}) \right] 
\sum_v \left[ \Psi(\eta_{0,v} + [\alpha_0]^{old}) - \Psi([\alpha_0]^{old}) \right]
\] (5.12)

\[
[\alpha_0]^{new} = \sum_{k=1}^{K} [\alpha_k]^{old}.
\] (5.13)

where \(\Psi\) is the Digamma function (i.e., the first derivative of the log Gamma function). Computing the difference of the Digamma functions \(\Psi(x + y) - \Psi(y)\) can be achieved using the Digamma recurrence relation,

\[
\Psi(x + y) - \Psi(y) = \sum_{i=1}^{x} \frac{1}{i - 1 + y}.
\] (5.14)

This approach can be done iteratively in order to maximize the computation efficiency and reduce the calculations involved [104]. However, parallelizing this computation is challenging due to the introduced linear dependency. To overcome this limitation, the following approximate function for the Digamma difference is used instead, proposed by Mackay and Peto [106],

\[
\Psi(x + y) - \Psi(y) = \frac{1}{y} + \log \frac{x + y - 0.5}{y + 0.5}.
\] (5.15)

With this approximate method, the cost of optimizing the hyperparameters of the
Dirichlet-multinomials is significantly reduced.

The symmetric hyperparameters, $\beta, \delta^{(g)}$. The approximate fixed point iteration method, detailed above, is utilized for optimizing the symmetric hyperparameters, $\beta$ and $\delta$, as well. To optimize $\beta$, the following equations run iteratively until convergence is reached,

$$[\beta]^{new} = [\beta]^{old} \frac{\sum_p |P| \left[ \frac{1}{\beta} + \log \frac{\eta_{k,k} + \beta - 0.5}{\beta + 0.5} \right]}{\sum_p |P| \left[ \frac{1}{\beta_0} + \log \frac{\eta_{k,k} + \beta_0 - 0.5}{\beta_0 + 0.5} \right]}, \quad (5.16)$$

$$[\beta_0]^{new} = \sum_p [\beta]^{old}. \quad (5.17)$$

The same method is applied to $\delta^{(g)}$, where the following update functions,

$$[\delta^{(g)}]^{new} = [\delta^{(g)}]^{old} \frac{\sum_t T^{(g)} \left[ \frac{1}{\delta^{(g)}} + \log \frac{\eta_{k,k}^{(g)} + \delta^{(g)} - 0.5}{\delta^{(g)} + 0.5} \right]}{\sum_t T^{(g)} \left[ \frac{1}{\delta_0^{(g)}} + \log \frac{\eta_{k,k}^{(g)} + \delta_0^{(g)} - 0.5}{\delta_0^{(g)} + 0.5} \right]}, \quad (5.18)$$

$$[\delta_0^{(g)}]^{new} = \sum_t [\delta^{(g)}]^{old}, \quad (5.19)$$

run iteratively until convergence.

### 5.2.3 Computing the New Representations

The user over latent patterns distribution can be estimated as: $P(h_v | z, \alpha) = \frac{1}{Z_{h_v}}$ Dirichlet($h_v | \eta_v + \alpha$), where $Z_{h_v}$ is a normalization factor and $\eta_v = (\eta_{1,v}, \cdots, \eta_{K,v})$.

Since the expectation of Dirichlet($x | \theta$) is $E(x_i) = \frac{\theta_i}{\sum_{k=1}^{K} \theta_k}$, the $k$th element in distribution vector $h_v$ is given by,

$$h_{k,v} = \frac{\eta_{k,v} + \alpha_k}{|C_v| + \alpha_0}. \quad (5.20)$$
Similarly, the location distribution of each latent pattern can be estimated as:

\[ P(w_k | C, \beta) = \frac{1}{Z_w} \text{Dirichlet}(w_k | \eta_k + \beta), \]

where \( Z_w \) is a normalization factor and \( \eta_k = (\eta_{1,k}, \cdots, \eta_{|P|,k}) \). Again, given the expectation of the Dirichlet, each element in the matrix \( W \) is estimated using,

\[ w_{p,k} = \frac{\eta_{p,k} + \beta}{\eta_{.,k} + \beta_0}. \]  (5.21)

Note that to get the location over latent patterns distributions, \( W \) is renormalized using, \( w_{p,k} : \sum_{k'} w_{p,k'} \).

Finally, the time distribution of each latent pattern can be estimated as:

\[ P(\omega_{k}^{(g)} | C, \delta) = \frac{1}{Z_{\omega}^{(g)}} \text{Dirichlet}(\omega_{k}^{(g)} | \eta_k + \delta), \]

where \( Z_{\omega}^{(g)} \) is a normalization factor and \( \eta_k = (\eta_{1,k}^{(g)}, \cdots, \eta_{T^{(g)},k}^{(g)}) \). Again, given the expectation of the Dirichlet, each element in the matrix \( \Omega^{(g)} \) is estimated using,

\[ \omega_{t,k}^{(g)} = \frac{\eta_{t,k}^{(g)} + \delta^{(g)}}{\eta_{.,k}^{(g)} + \delta_0^{(g)}}. \]  (5.22)

## 5.3 Evaluation

### Datasets

The objective of this evaluation is to shed light on the semantic meaning behind the inferred mobility patterns. For this reason, M-Times is evaluated using two publicly available datasets collected from LBSNs: NYC and TKY\[107\]. Table 5.1 provides summary statistics of each dataset. Both datasets are used in evaluation because they contain location category metadata and geographical address, which are used to assess the coherence of the inferred mobility patterns. Note that neither location categories nor their geophysical addresses are used during the learning phase, as M-Times learns from anonymized location IDs only.

### Experimental Settings

NYC and TKY are collected over a period of 11
Table 5.1: Datasets Statistics.

<table>
<thead>
<tr>
<th></th>
<th>No. of Users</th>
<th>No. of Locations</th>
<th>No. of users Check-ins</th>
<th>No. of Location Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>NYC</td>
<td>1083</td>
<td>38333</td>
<td>91024</td>
<td>83</td>
</tr>
<tr>
<td>TKY</td>
<td>2293</td>
<td>61858</td>
<td>211955</td>
<td>20</td>
</tr>
</tbody>
</table>

months. Consequently, the number of time scales \((G)\) is set to 3, capturing three time granularities: hours of a day, days of a week, and months of a year. It is important to reemphasize here that the proposed model provides a flexible approach to model any combination of time scales, yet this combination is selected because it simply provides a comprehensive overview of the pattern that one is usually interested in studying. \(K\) is set to 500\(^5\) LDA, LDA with asymmetric \(\alpha\) (LDA-A), and M-Times are all applied to the same datasets for comparison. The model is trained for 1000 iterations, and optimize the hyperparameters every 50 iterations, after an initial burn-in period of 200\(^6\).

**Evaluation Results.** Recall that the main objective in this chapter is to infer insightful mobility patterns from raw user-location data. Thus, the coherence of the inferred mobility patterns are evaluated by visualizing their top locations as well as their multi-scale time distributions. The purpose of this visualization is to shed light into the basis on what mobility patterns represent. Mobility patterns are considered to be coherent if the patterns of checked-in locations and time are consistent. Figure 5.2 visualizes two selected mobility patterns. Each pattern is described by four subfigures, a map showing the geophysical positions of the 20 most probable locations, the temporal distribution of the pattern over hours, weekdays, and months. Additionally, a list of the location categories associated with the 20 top locations is also provided. Collectively, all four distributions per a pattern provide insight into the semantics

---

\(^5\)Using large \(K\) is preferable over small \(K\) in order to capture interesting spatiotemporal patterns from the raw data. Additionally, the inferred patterns are robust to changes in \(K\) when it is large.

\(^6\)This is because the initial samples do not accurately represent the joint distribution, so it is recommended to optimize after a given period often noted as burn-in.

Table 5.2: Comparison of top locations, NYC dataset.

<table>
<thead>
<tr>
<th>Models</th>
<th>Top locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDA</td>
<td>Park, TrainStation, Park, TrainStation, Airport, Ferry, Office</td>
</tr>
<tr>
<td>LDA-A</td>
<td>Park, Gym, Park, Ferry CollegeAcademicBuilding, Office, ElectronicsStore</td>
</tr>
<tr>
<td>M-TIMES</td>
<td>Gym, SportingGoodsShop, Gym, Gym, Gym, Gym</td>
</tr>
</tbody>
</table>

behind the inferred pattern. For example, the first row in Figure 5.2 show patterns of check-ins to various fitness centers during the early morning, peaking on Tuesdays, in the month of April. The second pattern, on the other hand, has locations with different entertainment categories, with hour distribution peaking after working hours and on Fridays.

Note that baseline methods, LDA and LDA-A, do not provide time distributions per mobility pattern. As a result, only the spatial distributions are considered when comparing M-Times with both LDA and LDA-A. Table 5.2 shows the categories of
the top locations associated with selected patterns from the NYC dataset. The list of top locations in LDA, as expected, are dominated by the popular locations in the entire dataset, which are visualized using a word cloud in Figure 5.3. LDA-A provides better inferred patterns in the NYC data, where popular locations did not dominate all patterns. The proposed model provides the most coherent list of top locations as they all indicate fitness related centers and shops. It is important to highlight here that this improved coherency (and additional temporal distributions) is actually achieved with minimum cost added to the running time, as illustrated in Figure 5.4.

Figure Sensitivity of Hyperparameters. In this section, an investigation of
the impact of pre-setting the hyperparameters (as opposed to optimizing them as we advocate) is presented here. The impact of varying the setting of the $\beta$ hyperparameter (of the Dirichlet prior on the spatial patterns) is quantified, as an example. This is achieved by measuring the impact on the inferred patterns by measuring local and global KL-divergence\footnote{Note that KL is not a proper metric, and so the symmetric version is utilized here where the maximum of the two values is considered.} on the spatial patterns, $W$. Local KL-divergence denotes measuring the KL-divergence between each pair of the inferred spatial patterns. Global KL-divergence, on the other hand, is computed between each spatial pattern and the global empirical location distribution in the raw data. Small values indicate that the pair of patterns are similar, while larger values indicate dissimilar distributions.

The results are visualized when varying $\beta$ (see Figure 5.5) on both datasets using a boxplot. It is observed that as $\beta$ increases, the inferred patterns become more similar to the global distribution (Figure 5.5(a)) and as a result to each other (Figure 5.5(b)). This finding is consistent with the rule-of-thumbs on setting $\beta$, where a small value is preferred to avoid mimicking the global distribution. This being said, optimization of the hyperparameters, if done efficiently, reduces the overhead of finding the best parameter(s) value for the model, which varies in different datasets.
5.4 Discussion

This chapter presented an efficient mixed-membership model for inferring latent mobility patterns from raw user-location data. Comparing to previous work, the proposed model has the flexibility to capture regularities of human movement over multiple time-scales. Additionally, the proposed inference algorithm is efficient, which enables using the model as a tool to efficiently and effectively explore raw user-location data, available at an unprecedented scale thanks to the popularity of online social networks.

The main limitation of the proposed model is its inability to handle users with few recorded locations (inactive users). Note that the first proposed model, in Chapter 4, introduced sequences to handle the big variation in users check-in degree, which simply decomposes users trajectories into (almost) equal length sequences. Although this solution proved to be successful to handle the tail of the power law distribution, much of the user trajectories are very short and are eliminated in a preprocessing step as the model cannot handle them, very much like a generic LDA does not handle short tweets.

In the next chapter, however, the objective is to handle this limitation using two main components: 1) treat data as collections of locations, and thus avoid eliminating users with short trajectories, and 2) boost check-ins of locations by friends of visitors (using social ties), and thus avoid eliminating locations with few check-ins. Justification and details of the proposed model is presented in the next chapter. The work presented in this chapter is submitted to the 26th International Joint Conference of Artificial Intelligence (IJCAI 2017).
Chapter 6

Model III: Incorporating Social Ties

User location data collected from LBSNs provide incomplete user movements, as locations are manually entered and not automatically logged. Given that 30% (35%) of social network users have check-ins $\leq 5$ in GW (BK), learning patterns from this large number of incomplete users is a challenging task. Existing work differs on how they treated incomplete trajectories, where [71, 72] completed trajectories with estimates, and models in [78, 76] learned from incomplete semantically labeled trajectories with no attempt to compensate for missing data. Models I and II, proposed in Chapters 4 and 5 respectively, learn from long enough trajectories of anonymized locations, without any treatment of incompleteness but just eliminating short trajectories.

In this chapter, the proposed model learns from incomplete trajectories with anonymized locations by leveraging online social links. Note that incomplete trajectories are not necessarily short, while short trajectories are usually incomplete. This is achieved by viewing user location data as collections of statistically related location, where each location is described by a collection of users visits. This is different from the assumptions of Models I and II, as they learn from collections of users instead. This simple change in assumptions facilitates adding social ties in the model, using a generalized ploy urn sampling method, and achieves superior results.

Comparing to existing work, the proposed model, Social Trajectory Amplification and Representation (STAR), considers extrinsic factors (social ties) while learning mobility patterns, as opposed to existing work which only focuses on individuals’ intrinsically motivated mobility patterns (i.e., utilizing metadata directly associated
with check-ins, rather than with users). In this regard, this work introduces an important dimension to the human mobility framework.

The rest of this chapter is organized as follows. Section 6.1 motivates using social ties to compensate for incomplete trajectories. Sections 6.2 and 6.3 present details of the proposed model and the model parameter inference, respectively. Section 6.4 showcases evaluation results, and Section 6.5 draws concluding remarks.

6.1 Motivation

This section discusses the motivation behind including social ties while learning latent patterns. The attributes of two real datasets, Gowalla (GW) and Brightkite (BK), were analyzed previously in Section 1.1. The analysis highlighted the incompleteness of user location data collected from LBSNs, which emphasizes the need to learn from more than the single individual’s data. In this chapter, learning from social ties is advocated. To motivate the reason behind the proposed data fusion approach, analysis of social links and mobility is provided next.

Analysis of Links and Mobility. The correlation between human mobility and online social links is studied, to motivate amplifying individuals’ trajectories by those of their social ties. Precisely, the aim here is to answer the question: Are individuals’ mobility correlated with their 1, 2, and 3-degree links?

To answer the question, an approach to quantify the impact of the user’s social ties \(a^\kappa\)'s on his mobility \(v\) is proposed. First, user \(v\)'s mobility vector is reconstructed given only the mobility of users he is connected with, \(a^\kappa\). Then, the reconstruction sampling \(^1\) error is computed between a sample \(^1\) from the original mobility vector of \(v\), \(x_v\), and the reconstructed trajectory sample, \(\hat{x}_v\). \(\kappa = 1, 2, 3\) in \(a^\kappa\) indicates including the 1-, 2-, and 3-hop connected neighbors.

\(^1\) Sampling is used to overcome the curse of dimensionality, where sampling is achieved by considering all locations visited by \(v\) in addition to a random subset of non-visited locations.
The distribution of reconstruction error for all settings is reported in Figure 6.1, where errors closer to zero indicate better reconstruction. Clearly, the results show that 1) individuals’ mobility are correlated with their social links and 2) higher correlation is found between mobility of individuals and those of their 2-hop social links. As a result, the proposed representation learning model learns patterns from individuals along with their 2-hop social connections, though the model is general and can learn from any setting of $\kappa$.

In addition to the analysis presented above, many of the existing work investigated the correlation between social links and human mobility [109]. Given the high correlation found between the two, existing work have utilized patterns extracted from human mobility to infer social links [95, 96]. In fact, the effect of social links goes beyond human mobility. Many existing work study the impact of social networks on the spreading of behavior [110, 62], including as an example the spread of obesity in social networks [111].

### 6.2 The Proposed Model

**Model Description.** Unlike the two models proposed in Chapters 4 and 5 respectively, the model proposed in this chapter learns from collections of locations, where
each location is visited by a set of users. Users visit the same location due to some activities, and activities are the latent patterns to be inferred by the propose model. Each location is then represented as a distribution over activities, $w_p, p = 1...|P|$, and each activity has an attractiveness distribution over users (participants), $h_k, k = 1...K$. This change in settings is done so that the contribution of social ties can be added to the latent patterns using a generalized Pólya Urn sampling, with minimum cost (only a constant factor is added to the cost of regular LDA), and high scalability w.r.t. to $|V|$, as the degree of user ties follows a heavy-tailed power law distribution, as demonstrated previously in Figure 2.6. The graphical representation of the proposed model is presented in Figure 6.2.

Illustrative Example. To understand the representation learning process, an explanation of how a user $v$ visits a location $p$ is detailed below. For one location $p$, it has a distribution over activities (e.g., a plaza can host a variety of events). Denoting this per-location distribution over activities $w_p$, it can be sampled from a Dirichlet distribution with an asymmetric prior, $\alpha$, which itself is drawn from a Gamma distribution with two parameters $n, u$.

For one user $v$, whether or not he/she visits $p$ depends on two main factors:
Factor 1: which type of activity is occurring at $p$; and

Factor 2: his/her preference of attending such an activity.

Factor 1 is discussed firstly. As mentioned earlier, $w_p$ is the probability of different activities happening at $p$. An activity $z$ is selected according to the categorical distribution parameterized by $w_p$. For example, let a sunset concert happen at the plaza. Which user will come for the concert? Factor 2 is now discussed. Each activity has its attractiveness to different users, sampled from a Dirichlet distribution with a prior $\beta$. Activity $z$ (e.g., a concert) has attractiveness given in $h_{k=z}$ showing the probability of attracting each user. A user is more likely to be attracted if he/she is interested in it. Therefore, a user $v$ is chosen to visit $p$ according to the categorical distribution parameterized with $h_{k=z}$. Once a user $v$ visits a location $p$, his/her probability of visiting the same location again increases, analogously to sampling from a Pólya Urn distribution.

In fact, another factor attracting a user to visit $p$ is,

Factor 3: how many of his/her friends visit $p$.

For example, user $v$ may join a concert if his/her friends are going for it. The attractiveness distribution $h_{k=z}$ of one activity over users is reconsidered. Suppose user $v$ is socially connected with $v'$ ($a_{v,v'} = 1$). Due to the social influence analyzed previously, the probability of user $v$ going for $z$ is increased if $v'$ also goes for $z$. We thus should update $h_{k=z}$ by including $v'$ if $a_{v,v'} = 1$, $\forall v' \in \mathcal{V}$, as given in Eq. (6.4), such that the element in $h_{k=z}$ corresponding to $v$ is amplified by that of socially connected $v'$. This sampling approach is known as generalized Pólya Urn, where the social ties, $A$, are referred to as the addition matrix.

Model Specification. The formal generative process goes as follows:

where $n, u$ are the parameters of the Gamma distribution, $\mathcal{C}_p$ is the set of users who have check-ins at $p$ and $z$ is an activity drawn from the location-activity distribution.

In STAR, user check-ins, including $v$ and $p$ and social links $A$ are observed, and
1. Draw $\alpha \sim \text{Gamma}(n, u)$

2. For each activity, $k = 1:K$
   
   (a) Draw $h_k \sim \text{Dirichlet}_{|V|}(\beta)$

3. For each location $p \in \mathcal{P}$
   
   (a) Draw $w_p \sim \text{Dirichlet}_K(\alpha)$
   
   (b) For each user $v \in \mathcal{C}_p$
      
      i. Draw $z \sim \text{Categorical}(w_p)$
      
      ii. Draw $v \sim \text{Categorical}(\mathcal{C}_p)$

all other variables are latent. $w_p, p = 1...|\mathcal{P}|$ compose a $|\mathcal{P}| \times K$ matrix of location-activity distributions, $\mathbf{W}$, which is the new representation of locations. The $h_k, k = 1...K$ can be packed row by row, resulting in a $K \times |V|$ matrix $\mathbf{H}$, which is an activity-user distribution matrix.

Model Comparison. The proposed model is closely related to two topic models, LDA with optimized parameters [92] and coherent LDA [112]. START 1) utilizes the concept of optimized asymmetric location-activity hyperparameter [92] to handle the power law distribution of user check-ins, as well as 2) incorporates the generalized Pólya Urn model [112] to enhance the activity attractiveness distributions over users based on their social ties. Together, this combination of model elements serves to address the difficulties imposed by the special data attributes and the requirement of learning with social ties.

6.3 Estimating Model Parameters

Given the graphical model shown in Figure 6.2, the posterior distribution of the latent variables $(z, \Phi, \Theta, \alpha)$ given the observed data is,

$$P(\alpha, \Phi, \Theta, z|\mathcal{C}, n, u, \beta) = \frac{P(\mathcal{C}, \alpha, \Phi, \Theta, z|n, u, \beta)}{P(\mathcal{C}|n, u, \beta)}.$$  (6.1)
This distribution is intractable and cannot be computed exactly. The proposed approximate inference model alternates between 1) estimating $z$ using Generalized Pólya Urn Gibbs sampling and 2) optimizing $\alpha$ using fixed-point iteration.

**Estimating the Mixture Assignments.** Collapsed Gibbs sampling is an MCMC algorithm, which iteratively draws and updates one sample from the population given all other samples. In STAR, this implies sampling from the conditional distribution, $P(z_i|z^{-i}, C)$, since $z$ is a sufficient statistic for $H$ and $W$. Sampling is given by,

$$P(z_i|z^{(-i)}, C) \propto \frac{\eta_{z_i}^{(-i)} + \alpha_z}{\eta_{z_i}^{(-i)} + \beta_0} \cdot \sum_{v' \in V} \eta_{v'|z} \alpha_{v,v'}^2 + \beta,$$

where $\eta_{z_i}$ is the number of times activity $z$ was drawn for location $p$, superscript $(-i)$ excluded $i$-th element in $z$, $\eta_{v|z}$ is the number of times user $v$ attending activity $z$, $\eta_{z} = \sum_{j=1}^{|V|} \eta_{v|z}$ is the total number of users attending activity $z$, and $\beta_0 = \sum_{v=1}^{|V|} \beta$. Note how sampling using the generalized Pólya Urn framework incorporates social links into the formula. The conditional posterior includes not only users $v$, but also his friends. This approach of boosting the posterior by social ties, enforces coherence among users in the inferred activities, which serves the observation that mobility of connected users is correlated.

**Computing the New Representations.** The activity-user distribution for each activity is $P(h|C, \beta) = \frac{1}{Z_h}$Dirichlet$(h|\eta + \beta)$, where $Z_h$ is a normalization factor and $\eta = (\eta_{1,k}, \cdots, \eta_{|V|,k})$. Since the expectation of Dirichlet$(x|\theta)$ is,

$$E(x_i) = \frac{\theta_i}{\sum_{k=1}^{K} \theta_k},$$

each element in $h$ (the attractiveness of activity $z$ to user $v$) is estimated using,

$$h_{v|z} = \frac{\sum_{v'}^{|V|} \eta_{v'|z} \alpha_{v,v'}^2 + \beta}{\eta_{z} + \beta_0},$$

(6.4)
which also considers the social ties of $v$ (where $a_{v,v'}^2 = 1$).

The per-location activity distribution is given by $P(W|z, \alpha) = \frac{1}{Z_W} \text{Dirichlet}(W|\eta_p + \alpha)$, where $Z_W$ is a normalizing factor and $\eta_p = (\eta_{1p}, \cdots, \eta_{Kp})$. Again, given Equation (6.3), the $z$-th element in the distribution vector $w$ is,

$$w_{z|p} = \frac{\eta_{z|p} + \alpha_z}{|C_p| + \alpha_0}.$$  

(6.5)

The model’s complexity is equivalent to basic LDA, when $A$ is a sparse matrix, it is $O(K\cdot|P|\cdot|C_p|)$.

**Optimizing the Location-Activity Hyperparameter.** The hyperparameter $\alpha_k$ is assumed to be independently drawn from the Gamma distribution,

$$P(\alpha_k|n,u) = \frac{1}{\Gamma(u)n} (\frac{\alpha_k}{n})^{u-1} \exp\left(-\frac{\alpha_k}{n}\right).$$  

(6.6)

$\alpha$ can be estimated by maximizing the posterior probability. $P(\alpha|C) \propto P(C|\alpha)P(\alpha)$, where the likelihood $P(C|\alpha)$ and the hyperprior $P(\alpha)$ can be transformed into log forms. The summation of log-likelihood and log-prior has a lower bound, which can be used to find $\alpha_k^*$, thus maximizing $P(\alpha|C)$. The fixed-point iteration repeatedly runs Equation (6.7) and (6.8) until convergence is reached,

$$[\alpha_k]^{new} = [\alpha_k]^{old} + \frac{\sum_{l=1}^{L} \left[\Psi(\eta_{k|l} + [\alpha_k]^{old}) - \Psi([\alpha_k]^{old})\right] + u}{\sum_{l=1}^{L} \left[\Psi(\eta_{l|l} + [\alpha_0]^{old}) - \Psi([\alpha_0]^{old}) - \frac{1}{n}\right]},$$  

(6.7)

$$[\alpha_0]^{new} = \sum_{k=1}^{K} [\alpha_k]^{old},$$  

(6.8)

where $\Psi$ is the Digamma function (the first derivative of the log Gamma function). In this chapter, the digamma difference recurrence relation is used to compute the value.
Not the approximate, but actually the exact function as proposed by Wallach in [104]. The approximate function was used in the previous chapter to enable parallelization, as one of the goals was to improve efficiency. Here, the proposed inference method is simple, yet a more complex and efficient method can be considered in future work.

6.4 Evaluation

Datasets. The model is evaluated on two publicly available LBSNs\(^2\) Gowalla (GW) and Brightkite (BK) [44]. Table 6.1 contains basic statistics of the two datasets. User trajectories are represented as a bag-of-users, \(X\), and undirected social links are represented as the user adjacency matrix, \(A\).

<table>
<thead>
<tr>
<th></th>
<th>Gowalla</th>
<th>BrightKite</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Users</td>
<td>107,092</td>
<td>50,687</td>
</tr>
<tr>
<td>No. of Social Links</td>
<td>456,830</td>
<td>194,091</td>
</tr>
<tr>
<td>No. of Locations</td>
<td>1,280,969</td>
<td>772,965</td>
</tr>
<tr>
<td>No. of Check-ins</td>
<td>(\sum_{c,v} X)</td>
<td>3,981,334</td>
</tr>
</tbody>
</table>

Experimental Settings. The new user representation space is evaluated by measuring its performance on a link prediction task. The results are compared to those of the representations learned using two baseline models: LDA [90], and LDA with asymmetric location-activity hyperprior (LDA-A) [92].

The evaluation framework has the following steps:

1) Given the bag-of-users location representation, the new representations are learned. This includes the location-by-activity and activity-by-user matrices, using either the proposed model or one of the baseline models stated above.

2) The inferred activity-by-user matrix are renormalized to get a user-by-activity representation, where each user is described as a distribution over inferred activities.

\(^2\) This, however, should not limit the proposed model to data from LBSNs only as large-scale human trajectories from different sources exhibit similar attributes (such as CDRs). Thus, social ties can be leveraged to amplify incomplete trajectories in other datasources as well.
3) Unsupervised link prediction is applied, detailed next, and the average AUC is computed. Higher AUC indicates more accurate link prediction and thus better representations.

For all graphical models, $K$ is set to 5 unless otherwise noted, with thousand iterations, and parameters were optimized every 50 iterations, after an initial burn-in period of 200. The common parameter $\beta$ is set to be $\beta = 0.01$. The model’s sensitivity to $K$ is analyzed after comparing the performance of different models.

**Evaluation Results.** Figure 6.3 shows heatmaps of the new representations learned by STAR. Figure 6.3(a) and 6.3(b) show the per location activity representation, $w_p$, for a thousand randomly selected locations in GW and BK, respectively. Figure 6.3(c) and 6.3(d) show the per user activity representation, $h_u$, for a thousand randomly selected users in GW and BK, respectively. Note the difference between $w_p$ and $h_u$. The representations learned for users over activities, $h_u$, are mostly sparse, indicating that users are mostly characterized by a single activity. However, there exist locations with uniform distribution over activities, e.g., the light lines from location 190 to location 200 in Figure 6.3(a), though many locations are characterized by a single activity.

Similar observations are obtained when varying $K$. Figure 6.3(e) and 6.3(f) show the location and user representation of BK when $K = 50$. Larger $K$ reveals latent activities in a finer level. Both user and location representation heatmap show smaller and more clearly distinct block-diagonal pattern, without horizontal light lines. The distinct block-diagonal pattern of the heatmaps in Figure 6.3 evidences the high quality of the obtained representation. It properly captures latent patterns of locations and users, and can instantly be used to group them into clusters.

Three locations were manually inspected in GoogleMap and compared with their learned representations. Figure 6.4(a) shows the actual locations in a map and the representations learned by STAR when $K = 5$. Location 85 and 22 have representa-
Figure 6.3: Heatmaps of per-location and per-user activity distributions achieved by STAR when $K = 5$ and 50. The scale ranges from 0 (blue) to 1 (yellow).

tions $[0, 0, 0, 0, 1]$ and $[1, 0, 0, 0, 0]$, respectively. This indicates that location 85 is characterized by Activity 5 only, and location 22 is characterized by Activity 1 only. By zooming in the map, it is observed that location 85 is a building of “The Plastics Industry Trade Association” and location 22 is Dunkin Donuts, in Figure 6.4(b). It makes sense that these two locations are characterized by only one Activity, as users visit these places for simple and pure purpose, working or eating. Location 39 has a non-sparse representation $[0.3, 0.3, 0.1, 0.1, 0]$, which in other words is a mixture of different activities. By looking up, in Figure 6.4(b) the observed location represents a street with a bus station nearby. Checked-ins in this location can be attributed to several reasons. e.g., people waiting for a bus, people stuck in their car due to traffic jams, or some other reasons. The learned representation characterizes well the possibility of different activities.

**Link Prediction.** The inferred user representations are utilized in unsupervised link prediction, to assess the quality of the representation. For one pair of users $v_i$ and $v_j$, their proximity is calculated as $-||h_{v_i} - h_{v_j}||^2_2$, where $h_v$ is user $v$’s new representation. The existence of link between them is predicted when their proximity
Figure 6.4: Case Study: Inspection of 3 locations

is above a threshold, varying which results in an ROC curve. Table 6.2 shows the AUC values when using new representations learned by LDA, LDA-A and STAR. In both datasets (GW and BK), the new representation learned by STAR outperforms all other representation learning methods and the topology measures. Recall that both GW and BK have extremely sparse location-user frequency count matrix (< 0.003% non-zeros elements). STAR, unlike baselines, is shown to be able to capture patterns from the incomplete trajectories.

Note that unsupervised scores are measured here because they provide superior performance. In Chapters 3 and 4 computing scores using the user feature space only did not provide results that outperformed topology measures. This is not surprising given that topology measures comprise important information about the structure of the network, and is a good indicator of future link formation [67]. This is why in Chapters 3 and 4 topology measures were added to user mobility measures in a supervised framework to compute the final measures.

In this chapter, however, the model already includes topology measures while inferring users representations. This was initially done to boost learning for users with few check-ins records. However, it provided another benefit which allowed utilizing the new feature space directly in computing unsupervised scores for link prediction. It
Table 6.2: AUC values of link prediction

<table>
<thead>
<tr>
<th>Data</th>
<th>Topology Measures</th>
<th>Learned Representations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CN</td>
<td>AA</td>
</tr>
<tr>
<td>GW</td>
<td>0.592</td>
<td>0.557</td>
</tr>
<tr>
<td>BK</td>
<td>0.661</td>
<td>0.664</td>
</tr>
</tbody>
</table>

is important to note that unsupervised measures are usually preferred because there is no need to specially handle the imbalance of link labels (positive to negative ratio) as it is the case with supervised methods. In supervised methods, this is handled by eliminating many of the negative links (e.g., only considering 2-hop pairs of users) which basically means changing the data distribution, in such a case the performance measures will not reflect the real capabilities and limitations of the model [113].

Parameter Sensitivity. The sensitivity of STAR to the setting of the number of activities ($K$) is analyzed here. Figure 6.5 shows the AUC results of link prediction for BK dataset when varying $K$. STAR significantly outperforms both LDA and LDA-A with much higher AUC values when varying $K$. Prediction results degrade when the number of activities increases over 50, in all models. As optimizing the location-activity hyper-prior minimizes the model’s sensitivity to $K$. This behavior is only reached when $K$ is large enough, for small values of $K$ such as in our experiments, a model with optimized $\alpha$ is still sensitive to the user set parameter $K$.

![Figure 6.5: Sensitivity analysis of $K$](image)
6.5 Discussion

This chapter presented STAR, a mixed membership model designed to infer latent patterns from anonymized and incomplete location-user traces, where user check-ins obey a power-law distribution. The model incorporates social links and asymmetric priors in the learning process, as a way of compensating for the limitations imposed by the incomplete data. The work presented in this chapter was published in the Proceedings of the 16th IEEE International Conference on Data Mining (ICDM 2016), and was presented in Barcelona, Spain, on December 12 - 15, 2016 [79].
Chapter 7

Concluding Remarks

This chapter concludes the dissertation with a summary of contributions and discussion on future research directions.

7.1 Summary of Contributions

The demand on utilizing human mobility data in different applications is growing with the increasing availability of user-location data at an unprecedented scale. It is infeasible to directly utilize the high-dimensional user-location matrix of check-in counts due to the curse of dimensionality. Thus, learning a new representation space is critical for such data, which is the main objective of this dissertation. A summary of the main contributions is provided next:

• Proper evaluation of the inferred representation is an important task in this work as it allows for assessing the quality of the new feature space, and thus enables comparing different models. The first contribution in this dissertation is in proposing a quantitative evaluation measure to assess the quality of the inferred representation. This is done using an external task, where performance on a social link prediction is measured using the Area Under the ROC Curve (AUC). The AUC value is used to assess the inferred representation as well as to compare it to those inferred by other latent feature models. In theory, the performance on any relevant task can be used to assess the quality of the inferred representations, as long as ground truth data exists. However, given
the nature of user-location data, ground truth labels associated with users (e.g., age, gender, and so on) are not available due to privacy concerns. Social ties among users, on the other hand, are often available and are portrayed in social graphs, and thus this work advocates for utilizing them in evaluation.

- The second contribution of this dissertation is the empirical evaluation of two popular approaches for defining the new feature space: feature engineering vs latent feature models. Although feature engineering is very powerful and has been used extensively with various data types including user-location data, it is limited by what the experts ‘think’ is useful. On the contrary, latent feature models infer the hidden patterns, without guidance from the experts, which enable capturing unknown correlations from the raw data. The experimental evaluation of generic latent feature models vs feature engineering shows the superiority (and sometimes equivalence) of the former method on social link prediction. For details on the results, please refer to Chapter 3.

- More importantly, the main contribution in this dissertation lies in proposing three latent feature models designed to infer mobility patterns from user-location data. Generic latent feature models can infer patterns from user-location data, as shown in Chapter 3, yet without accounting for the special attributes associated with such data.

**Model 1** HuMoR incorporates discretized time into the inference model to add context to the inferred patterns as well as to balance the power-law distribution of trajectory lengths (by introducing sequences).

**Model 2** M-TIMES provides a more flexible approach to incorporate time at different granularities (hours of day, day of week, and so on) into the learning process. This is achieved by modeling time as a discrete multi-scale value, with a Dirichlet prior over each time scale. The model avoids over-
fitting to the temporal patterns due to the imposed Dirichlet prior. Additionally, M-TIMES proposes an efficient inference method that enables utilizing it in large scale user-location data.

**Model 3** STAR provides a mean to infer patterns from users with very short trajectories, which was not possible in previous models. This is achieved by incorporating links in the social graph, i.e., by boosting ones trajectory using trajectories of his/her friends.

### 7.2 Future Research Directions

The dissertation emphasizes the importance of designing latent feature models to handle the special characteristics of user-location data. This topic of research is still in its early stages with unlimited directions for future work. Potential future research directions are provided next:

- All models proposed in this dissertation are unsupervised which enable learning latent patterns from raw user-location data only. Proposing semi-supervised models would enable harvesting the knowledge provided in labels, without the need to have labels for all samples. Such models have the potential to learn ‘specific’ latent patterns in the data that targets one set of labels, and can provide more meaningful representation spaces.

- Individual digital traces when put together can offer huge insight into the details of our lives - including details one assumed were private. Inferred latent patterns from user-location data can reveal information that end-users do not wish to share, especially when descriptive metadata are available (location coordinates or names). Security is a big issue that needs to be considered in such studies. For that reason, all models proposed in this dissertation learn from anonymized locations, to minimize the privacy concerns. This being said, geogrpahical con-
strains are an important aspect to consider when modeling user-location data, given that users check-ins are bounded by travel distance [114]. A privacy-aware model that incorporates such geolocation data is much needed.

- All models proposed in this dissertation as well as generic models assume that the inferred patterns are static (do not change over time). However, such assumption is a simplified version of the reality. Given the nature of humans, patterns governing their mobility are dynamic and constantly change over time. Models that enable inferring dynamic patterns would capture more realistic representation spaces, and provide an interesting direction for future work. Additionally, depending on the application of interest, models that capture both local and global patterns for users might provide better understanding of the raw data [115].
REFERENCES


APPENDICES

A Papers Published and Submitted

A.1 Published


2. Basma Alharbi, Abdulhakim Qahtan, Xiangliang Zhang, “Minimizing User Involvement for Learning Human Mobility Patterns from Location Traces”. Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI 2016), Phoenix, Arizona, February 12-17, 2016 (acceptance rate = 549/2132 = 25.7%).

3. Abdulhakim Qahtan, Basma Alharbi, Suojin Wang, Xiangliang Zhang, “A PCA-Based Change Detection Framework for Multidimensional Data Streams”. Proceedings of the 21st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2015), Sydney, Australia, August 10-13, 2015 (acceptance rate = 159/819 = 19.4%).

A.2 Submitted

1. Basma Alharbi, Xiangliang Zhang, “Unfolding Spatial Mobility Patterns over Multiple Time-Scales: An Efficient Representation Learning Model for User-Location Data”. *Submitted to the 26th International Joint Conference on Artificial Intelligence (IJCAI 2017).*