Latent representation models for mining geo-spatial data

Liu, Yiding

2020

Liu, Y. (2020). Latent representation models for mining geo-spatial data. Doctoral thesis,
Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/137358>

<https://doi.org/10.32657/10356/137358>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0
International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 21:10:56 SGT

LATENT REPRESENTATION MODELS FOR MINING GEO-SPATIAL DATA

YIDING LIU

School of Computer Science and Engineering

2020

LATENT REPRESENTATION MODELS FOR MINING GEO-SPATIAL DATA

YIDING LIU

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2020

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

[Input Date Here]

15 July 2019

.....

Date

[Input Signature Here]

.....

Yiding Liu

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

[Input Date Here]

15 July 2019

.....
Date

[Input Supervisor Signature Here]

.....
Prof Gao Cong

Authorship Attribution Statement

This thesis contains material from 3 paper(s) published in the following peer-reviewed conferences in which I am listed as an author.

Chapter 3 is published as **Yiding Liu**, Tuan-Anh Nguyen Pham, Gao Cong, Quan Yuan. An Experimental Evaluation of Point-of-interest Recommendation in Location-based Social Networks. *In Proceedings of the VLDB Endowment (VLDB 2017)*, Volume 10, Issue 10, Pages 1010-1021, June 2017.

The contributions of the co-authors are as follows:

- Dr Quan Yuan, Prof Gao Cong and I provide the idea and initial direction.
- I designed the evaluation procedures. Dr Tuan-Anh Nguyen Pham and Prof Gao Cong provided useful comments for designing the evaluation procedures.
- I implemented all of the source code and conducted all experiments and prepared the manuscript drafts.
- Dr Tuan-Anh Nguyen Pham, Dr Quan Yuan and Prof Gao Cong assisted in the revision of the manuscript drafts.

Chapter 4 is published as **Yiding Liu**, Kaiqi Zhao, Gao Cong, Zhifeng Bao. Online Anomalous Trajectory Detection with Deep Generative Sequence Modeling. In Proceedings of the 36rd IEEE International Conference on Data Engineering (**ICDE 2020**), Dallas, Texas, USA, April 2020.

The contributions of the co-authors are as follows:

- I proposed the initial idea and formulate the problem. Prof Gao Cong and Prof Zhifeng Bao assisted in formulating the problem.
- I designed the methodologies. Dr Kaiqi Zhao and Prof Gao Cong provided useful comments for designing methodologies.
- I designed the evaluation procedures. I implemented all of the source code and conducted all experiments. I prepared the manuscript drafts.
- Dr Kaiqi Zhao assisted in the collection and preprocessing of data.
- Dr Kaiqi Zhao, Prof Gao Cong and Prof. Zhifeng Bao assisted in the revision of the manuscript drafts.

Chapter 5 is published as **Yiding Liu**, Kaiqi Zhao, Gao Cong. Efficient Similar Region Search with Deep Metric Learning. *In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (SIGKDD 2018)*, Pages 1850-1859, London, United Kingdom, August 19 - 23, 2018.

The contributions of the co-authors are as follows:

- Prof Gao Cong and I proposed the initial idea and formulated the problem,
- I designed the methodologies. Dr Kaiqi Zhao and Prof Gao Cong provided useful comments for designing methodologies.
- I designed the evaluation procedures. I implemented all of the source code and conducted all experiments. I prepared the manuscript drafts.
- Dr Kaiqi Zhao assisted in the collection and preprocessing of data.
- Dr Kaiqi Zhao and Prof Gao Cong assisted in the revision of the manuscript drafts.

[Input Date Here]

[Input Signature Here]

15 July 2019

.....
Date

.....
[Yiding Liu]

Acknowledgments

I would like to express my sincere gratitude to my supervisor Prof. Gao Cong, for his patient guidance and elaborative supervision. Under his mentorship, I have developed a better “taste” in research, including how to study new research areas, select research topics, develop novel methods and design experimental procedures. More importantly, his passion and concentration on research have a great influence on me, and always encourage me to pursue greater achievements. I would also like to express gratitude to Prof. Zhifeng Bao (RMIT), Prof. Guosheng Lin (NTU) and Prof. Junsong Yuan (SUNY Buffalo) for their valuable comments to my research.

I would like to express special appreciations to Dr. Tuan-Anh Nguyen Pham, Dr. Kaiqi Zhao and Dr. Renjie Wan, who gave me invaluable advices and suggestions for my research, and led me through the most difficult times of my postgraduate study. I would also like to acknowledge my collaborators, colleagues and friends for dining, drinking (coffee) and having pleasant discussions with me. They include Dr. Quan Yuan, Dr. Xiucheng Li, Dr. Xin Cao, Dr. Lisi Chen, Dr. Shanshan Feng, Dr. Kaiyu Feng, Dr. Tao Guo, Dr. Zhida Chen, Prof. Xutao Li, Dr. Miao Xie, Dr. Zongcheng Ji, Dr. Guangtao Wang, Dr. Shaokai Wang, Prof. Lu Chen, Dr. Yixiang Fang, Dr. Tay Yi, Dr. Seyed Ali Majid Zonoozi, Jin Yao Chin, Tran Dang Quang Vinh, Zheng Wang, Yue Chen, Yile Chen, Junghoon Kim, Nguyen Thanh Tung, Xiangyu Zhao, Huaxiu Yao, Peng Han, Dr. Jing Li, Dr. Yequan Wang, Dr. Di Yao, Grace E. Lee, Ting Lin, Dr. Xin Zheng, Minh C. Phan, Keke Huang, Xiangyu Ke, Dr. Jingwen Zhao, Dr. Hongkui Tu, Dr. Jilian Zhang, Dr. Dennis Sng, Qian Wang and Alfred Wu. In addition, I also thank all the researchers who shared their minds with me through email or at conferences, including Prof. Sibow Wang (CUHK), Prof. Ada Waichée Fu (CUHK), Prof. Xiaokui Xiao (NUS), Dr. Wenqing Lin (Tencent), Dr. Bin Liu (IBM), etc.

Finally, I would like to thank my parents, my sisters and brothers, for being such a wonderful family and support me as always. Most importantly, I express my deepest gratitude to my girlfriend Hui Tan for her unconditional, unselfish and unlimited love and care. I dedicate this dissertation to her.

Summary

With the proliferation of mobile devices and location-based services (LBS), huge amounts of geo-spatial data (e.g., check-ins, points of interest, GPS traces) are becoming available from numerous sources. These data provides us unprecedented opportunities to uncover the complex semantics and dynamics of geographical space and human mobility. It is promising to develop data-driven approaches for harnessing geo-spatial data and discovering pivotal knowledge to support a wide spectrum of real-world applications, such as business analytics, spatial information retrieval, traffic management, etc.

In this dissertation, we design latent representation models for mining geo-spatial data. In particular, we aim at discovering knowledge and learning representations for four types of geo-spatial entities: point-of-interest (POI), trajectory, region and spatiotemporal activity, to respectively support four location-based applications: POI recommendation, anomalous trajectory detection, similar region search, and spatiotemporal activity modeling. Correspondingly, our solutions include four types of latent representation models: Matrix Factorization (MF), Variational Sequence Auto-Encoder, Convolutional Neural Network (CNN) and Multimodal Embedding (ME). More importantly, we investigate and improve existing latent representation models by considering the unique properties of different types of geo-spatial data. The detailed contributions are listed as follows:

- **Learning POI/user representation for POI recommendation.** We provide an all-around evaluation of 11 state-of-the-art POI recommendation models. From the evaluation, we obtain several important findings, such as the superiority of modeling check-in as implicit feedback with ranking-based/weighted Matrix Factorization, and how to model the geographical influence for different types of users. Based on these findings, we can better understand and utilize POI recommendation models in various scenarios. We provide an overall picture of the cutting-edge research on POI recommendation.
- **Learning route representation for anomalous trajectory detection.** We study the problem of anomalous trajectory detection and propose a novel VAE-based model, namely Gaussian Mixture Variational Sequence AutoEncoder (GM-VSAE). Our GM-VSAE model is able to (1) capture complex sequential information

enclosed in trajectories, (2) discover different types of normal routes from trajectories and represent them in a continuous latent space, and (3) support efficient online detection via trajectory generation. Our experiments on two real-world datasets demonstrate that GM-VSAE is more effective than the state-of-the-art baselines and is efficient for online anomalous trajectory detection.

- **Learning region representation for similar region search.** We study the problem of similar region search on geographical space. We propose a novel solution that is equipped by (1) a deep metric learning approach to learn the similarity that considers the relative locations among the geo-spatial objects within each of the regions; and (2) an efficient branch and bound search algorithm for finding top-N similar regions. Moreover, we propose an approximation method to further improve the efficiency by slightly sacrificing the accuracy. Our experiments on three real world datasets demonstrate that our solution improves both the accuracy and search efficiency by a significant margin compared with the state-of-the-art methods.
- **Learning multimodal representation for spatiotemporal activity modeling.** We study the problem of spatiotemporal activity modeling, which aims at jointly learning representations of different regions, time units (e.g., hours) and activities (i.e., keywords). We propose a Meta Multimodal Embedding (MME) method that can capture rich semantics of regions, time units and activities from geo-textual data. More importantly, it can explicitly learn temporally transferable priors for different regions in a geographical space, which create specialized activity models for different time periods. We further extend the MME model to a Spatial-Aware MME model that incorporates geographical influence between nearby regions. Our experiments on two real world datasets demonstrate that our proposed methods can better answering region prediction queries on nonstationary geo-textual data.

Contents

Acknowledgments	i
Summary	ii
List of Figures	ix
List of Tables	xii
1 Introduction	1
1.1 Background and Motivations	1
1.2 Research Problems and Methodologies	6
1.2.1 Point-of-Interest Recommendation	6
1.2.2 Anomalous Trajectory Detection	7
1.2.3 Similar Region Search	10
1.2.4 Spatiotemporal Activity Modeling	12
1.3 Summary of Contributions	15
2 Literature Survey	17
2.1 Mining Geo-spatial Data	17
2.2 Location-based Applications	20
2.2.1 POI Recommendation	20
2.2.2 Detecting Anomalous Trajectory	23
2.3 Similar Region Search	24
2.3.1 Measuring Region Similarity	24
2.3.2 Efficient Search for Similar Regions	25
2.3.3 Metric Learning	26
2.3.4 Spatiotemporal Activity Modeling	26
2.4 Latent Representation Models	27

2.4.1	Matrix Factorization	27
2.4.2	Recurrent Neural Network	28
2.4.3	Convolutional Neural Network	29
2.4.4	Word2vec	29
3	Point-of-Interest Recommendation	31
3.1	POI Recommendation	31
3.2	Models for Evaluation	32
3.2.1	Matrix Factorization Models	33
3.2.2	Poisson Factor Models	35
3.2.3	Link-based Models	36
3.2.4	Hybrid Models	36
3.3	Evaluation Setting	38
3.3.1	Datasets	38
3.3.2	Evaluation Metrics	39
3.3.3	Performance Evaluation Procedure	40
3.4	Evaluation Results	42
3.4.1	Performance on Different Types of Data	42
3.4.2	Performance on Different Users	48
3.4.3	Different Modeling Methods	50
3.4.4	Scalability	54
3.4.5	Summary of New Insights	60
4	Anomalous Trajectory Detection	63
4.1	Problem Formulation	64
4.1.1	Problem Definition	64
4.1.2	Overview of Our Solution	65
4.2	Gaussian Mixture Variational Sequence Autoencoder	66
4.2.1	Latent route inference.	68
4.2.2	Modeling latent route with Gaussian mixture distribution.	68
4.2.3	Recurrent trajectory generation.	69
4.2.4	Training GM-VSAE	70

4.3	Efficient Online Anomalous Trajectory Detection via Trajectory Generation	71
4.4	Improving Efficiency with Approximate Inference of Route Type	74
4.5	Experiments	75
4.5.1	Experimental Setup	75
4.5.2	Effectiveness Evaluation	78
4.5.3	Efficiency Evaluation	81
4.5.4	Case study	83
4.5.5	Visualization of Route Types	83
4.6	Summary	84
5	Similar Region Search	87
5.1	Problem Definition	88
5.2	Region Similarity Learning	89
5.2.1	Ranking Metric Learning Model	90
5.2.2	Training Robust Triplet Network	91
5.3	Region Search Algorithms	92
5.3.1	Problem Transformation via Sharing Computation	93
5.3.2	Exact Algorithm for <i>SFRS</i>	94
5.3.3	Approximate Algorithm for <i>SFRS</i>	97
5.4	Experiments	98
5.4.1	Experimental Setup	98
5.4.2	Effectiveness of Triplet	100
5.4.3	Efficiency of ExactSFRS and ApproxSFRS	103
5.5	Conclusion	105
6	Modeling Nonstationary Spatiotemporal Activity	107
6.1	Problem Description	109
6.1.1	Spatiotemporal Activity Modeling.	109
6.1.2	Period-specific Spatiotemporal Activity Modeling.	109
6.2	Preliminary	110
6.2.1	Multimodal Embedding	110
6.2.2	Limitation and Challenge of Multimodal Embedding	111

6.3	Period-Specific Modeling via Temporal Transfer Learning	111
6.3.1	Transferable Priors of Regions	112
6.3.2	Meta Multimodal Embedding	113
6.4	Spatial-aware Prior Estimator	115
6.4.1	Incorporating Spatial-aware Prior Estimator for MME	115
6.4.2	Posterior Inference via Spatial-Aware Stein Variational Gradient Decent	117
6.5	Experiments	119
6.5.1	Experimental Setup	119
6.5.2	Quantitative Experiments	121
6.5.3	Parameter Sensitivity	121
6.6	Summary	123
7	Conclusions and Future Work	124
7.1	Conclusions	124
7.2	Future Directions	126
A	List of Publications	127
	Appendix - List of Publications	127
	References	129

List of Figures

1.1	Examples of different geo-spatial entities.	2
1.2	The overall workflow of using latent representation model.	4
1.3	Overview of the latent representation models and their applications. . . .	5
1.4	Examples of anomalous trajectory detection. (S_1, D_1) and (S_2, D_2) are two source-destination pairs. r_1 and r_2 are two normal routes between S_1 and D_1 ; r_3 and r_4 are two normal routes between S_2 and D_2 . T_1 and T_2 are two anomalous trajectories of (S_1, D_1) and (S_2, D_2) , respectively. . .	9
1.5	Examples of the activities sampled during different time periods within a university area in New York City (#1: daily life; #2: Halloween; #3: graduation).	13
3.1	Spatial and social influence.	33
3.2	Varying K on Gowalla.	42
3.3	Varying K on Foursquare.	42
3.4	Varying K on Yelp.	43
3.5	Varying data sparsity.	43
3.6	Performance of POI recommendations for users with different numbers of check-in POIs	43
3.7	Performance of POI recommendations for users with different average distance among check-in POIs.	44
3.8	Varying K on Foursquare Singapore.	46
3.9	Varying K on Foursquare Tokyo.	47
3.10	Varying K on Gowalla (dense).	48
3.11	Varying K on Gowalla (large).	49

3.12	Varying K on Foursquare, Foursquare (tourist) and Florence.	50
3.13	Performance of geographical modeling methods.	51
3.14	Performance of geographical modeling methods for users with different numbers of check-in POIs.	52
3.15	Performance of social modeling methods.	53
3.16	Performance of social modeling methods for users with different numbers of friends.	54
3.17	Performance of user preference modeling methods.	55
3.18	Performance of user preference modeling methods for users with different numbers of check-in POIs.	56
3.19	Relative training and querying time on Gowalla.	57
4.1	The overall architecture of GM-VSAE.	67
4.2	The framework of anomalous trajectory detection using GM-VSAE. . . .	73
4.3	The framework of anomalous trajectory detection using SD-VSAE. . . .	73
4.4	Examples of generated anomalies.	76
4.5	GM-VSAE10 vs. SD-VSAE10 (D: $d = 3$, $\alpha = 0.3$; RS: $\beta = 0.3$).	83
4.6	Efficiency comparison for anomalous trajectory detection.	83
4.7	Accumulated time cost when a trajectory is generated sequentially. . . .	84
4.8	Case study on Porto data (the most anomalous trajectory : orange dashed lines; other trajectories : blue solid lines).	85
4.9	Visualization of trajectories sampled from different route types (#15: tra- jectories in downtown area; #8: trajectories on highways; #19: trajecto- ries on the west side of Porto; #4: trajectories across city).	86
5.1	An example of defining $sim(\cdot)$ (R_q represents the query region, and R_1 is the most similar to R_q , as they have similar object attributes and relative locations; R_2 is less similar to R_q than R_1 , because the relative locations is dissimilar; R_3 is the most dissimilar to R_q).	89
5.2	The workflow of region similarity learning.	90
5.3	Sharing $\mathcal{X}_P$ for computing $sim(R_q, R_c)$	94

5.4	An example of search space, which is represented by four intervals $[t_{min}^*, t_{max}^*]$, $[b_{min}^*, b_{max}^*]$, $[l_{min}^*, l_{max}^*]$ and $[r_{min}^*, r_{max}^*]$	95
5.5	Varying noise rate.	101
5.6	Varying shift distance.	101
5.7	Comparing different triplet networks	102
5.8	Case study – finding top-5 similar regions (black rectangle: query region; red rectangle: found similar region).	103
5.9	Runtime on different datasets.	104
5.10	Runtime of ExactSFERS by varying N	104
5.11	Trade-off of ApproxSFERS by varying M	104
6.1	Meta Multimodal Embedding for period-specific spatiotemporal activity modeling.	113
6.2	Construction of prior estimator using nearby regions (red square: target region; blue squares: nearby regions weight by distance).	116
6.3	Parameter studies for SA-MME.	122
6.4	Varying d on NYC and LA data.	123

List of Tables

2.1	Summary of existing POI recommendation papers (sorted by publication years)	21
3.1	Summary of the POI recommendation models in our evaluation.	38
3.2	Data descriptions	47
3.3	Training time (hour) of the models	54
3.4	Avarage querying time (second) per user of the models for users with different numbers of check-in POIs	54
3.5	List of notations.	59
3.6	Time complexity of the POI recommendation models.	60
4.1	Description of notations.	66
4.2	Trajectory datasets after preprocessing.	75
4.3	Description of notations used in experiments.	76
4.4	Performance comparison for anomalous trajectory detection (detour) in terms of PR-AUC on Porto dataset.	79
4.5	Performance comparison for anomalous trajectory detection (route-switching) in terms of PR-AUC on Porto dataset.	80
4.6	Performance comparison for anomalous trajectory detection (detour) in terms of PR-AUC on Beijing dataset.	81
4.7	Performance comparison for anomalous trajectory detection (route-switching) in terms of PR-AUC on Beijing dataset.	82
5.1	Notations for similarity learning.	89
5.2	Notations for the search algorithms.	93

5.3	Spatial object datasets.	98
5.4	Average runtime (sec.) for query regions with different numbers of objects on SG.	105
5.5	Average runtime (sec.) for query regions with different areas (km ²) on SG.	105
6.1	Description of frequently-used notations.	112
6.2	Tweet datasets.	121
6.3	Performance of different methods for region prediction (metric: MRR). .	121

Chapter 1

Introduction

1.1 Background and Motivations

With the prominence of location-aware social media (e.g., Foursquare, Yelp), mobile devices and sensor technologies, a huge amount of geo-spatial data, such as check-ins, geo-textual posts and location traces, are being generated at an unprecedented speed from numerous sources. These data provides us great opportunities for understanding different semantics of urban space and human dynamics. In particular, there is a variety of geo-spatial data types that one can encounter in different real-world applications. Including those associated with temporal information (e.g., timestamp), geo-spatial data can be categorized into four types [AKK18]: 1) *point reference data*, i.e., measurement of reference points in space, 2) *trajectory data*, which denotes the paths traced by objects moving in space over time 3) *spatial map data*, i.e., measurements from a set of locations, and 4) *event data*, which comprises discrete events occurring at point locations and times. For those geo-spatial data types, there are four geo-spatial entities that can be considered as the basic building blocks of analyzing a wide spectrum of problems and location-based applications:

- **Point-of-interest (POI).** A POI represents a particular place that people may find it useful or interesting (e.g., restaurant, hotel). Each POI might be associated with check-in data from its visitors, which reveals the *properties about the POI*, such as the types of the customers the POI attract. It can be considered as the point reference of the semantics on a particular location.
- **Trajectory.** A trajectory can be represented by a sequence of chronologically ordered locations points, where each point is represents by GPS coordinates. Trajectories are usually collected via monitoring the movement of pedestrians or vehicles, and thus contains rich information about their *mobility behaviors* on geographical space.

Figure 1.1: Examples of different geo-spatial entities.

- **Region.** A region is a type of spatial map, which is represented by a bounded area (i.e., subspace) that belongs to a larger geographical space (e.g., a city). A region may contain a set of geo-tagged objects (e.g., POIs, geo-tagged contents), which characterize the *region functions and semantics* (e.g., shopping area, university).
- **Spatiotemporal activity.** A spatiotemporal activity is usually represented by geo-textual posts that associated with specific GPS locations, timestamps and textual content (e.g., keywords), which indicates *real-time local semantics or events* at particular places (e.g., basketball game at Staples Center, 6:30pm Oct 5).

Figure 1.1 shows examples of each type of the geo-spatial entities. The geo-spatial data associated with these entities reflect geo-spatial knowledge in different granularity (e.g.,

location-level, region-level) and from different perspectives (e.g., static semantics, mobility behaviors). Therefore, it is of great interests for mining geo-spatial data and discovering pivotal knowledge behind these geo-spatial entities.

To achieve this, extensive studies have been conducted in the recent decade, which enable many novel applications and location-based services. For example, previous studies leverage large-scale check-in data collected from location-based social networks (LBSN) to discover the preferences and properties of users and POIs for supporting POI recommendations [LZX⁺14, LCL⁺15, LWSM14, YYL10, YYLL11, ZZKL17, WSOC18, MZWL18, ZCZ15, ZCYZ15, YCZ⁺16, YBZ⁺17, WFW⁺17], mobility prediction [FLZ⁺18, WYL⁺15, LXZ⁺15, LZX13, HE13, YCM⁺13b, YCZ⁺15] and visitor prediction [FCAC17]. Moreover, there are also rich literatures that mine trajectory data for various tasks, such as map matching [HQY⁺13, BPSW05, LZZ⁺09, YZZ⁺10b], trajectory classification [LHLG08, ZCL⁺10, ZGT⁺18, BCTH12] and anomalous trajectory detection [LHL08, ZLZ⁺11, ZJL⁺15, LXZ⁺17, CZC⁺13, GXLZ11, SWX⁺18, WSZ17]. Another line of researches focus on discovering region functions [ZZG⁺18, YZX12, SZZ⁺18] and topics [YCH⁺11, KKSS14, HAG⁺12, YCM⁺13b] to support region-level applications, such as regional crime or popularity prediction [WL17, ZT17, HZZC18, ZZG⁺18, FWD⁺19, SZZ⁺18], similar region search [SZH⁺10] and region recommendation [PLC17]. In addition, many studies utilize geo-textual posts for discovering the temporal dynamics of the semantics in different places, which aims at supporting spatial query processing [ZZY⁺17a, ZLL⁺18, ZZY⁺17b], local event detection [WK13, ZLL⁺17, ZLY⁺18, ZZY⁺16a] and spatial event forecasting [ZZB⁺17, ZCLR15, ZYC⁺16, ZCY19, ZCLR16, ZSY⁺15].

For designing approaches to support these applications, an important problem is how to *represent* the knowledge behind different geo-spatial entities. For example, Memory-based Collaborative Filtering for POI recommendations usually represents a user as the POIs he/she has visited, and a POI as its historical visitors to indicate their preferences and properties, respectively. Another example is that when discovering region functions and mining regional topics, a region is usually represented as a probability distribution over a set of keywords.

Recent years have witnessed the huge successes of *latent representation models* [BCV13] (e.g., deep learning [LBH15, GBC16], word2vec [MSC⁺13], graph embedding [CWPZ18, PARS14], etc.) for solving real-world problems in various domains, such as recommender systems [ZYST19], natural language processing [YHPC18], visual recognition [KSH12, Gir15], etc. In particular, latent representation models aim at extracting and representing useful knowledge associated with data entities (e.g., images, words) in *latent representations*, each of which is denoted by a low-dimensional vector in a latent embedding space. The latent representations can therefore be considered as features for supporting downstream applications. Figure 1.2 shows how a latent representation model bridges the gap

Figure 1.2: The overall workflow of using latent representation model.

between data sources and applications via learning latent representations. Those models are powerful in extracting semantic information from massive data in a data-driven manner and encoding such information in the latent representations.

However, there are only a handful of location-based applications that leverage latent representation models to extract and represent knowledge in geo-spatial data. In particular, many location-based applications (e.g., anomalous trajectory detection) are still powered by hand-crafted feature engineering, which is not flexible and expressive for modeling complex geo-spatial data. This also largely limits novel applications and methods to be rapidly constructed. Although some applications (e.g., POI recommendations) are already benefited from using latent representation models, there is still a lack of in-depth investigation about how, why or when the latent representation models would provide promising results.

In this dissertation, we contribute a systematic study on latent representation models for mining geo-spatial data. In particular, we formulate latent representations of the four types of geo-spatial entities, which encode informative knowledge extracted from massive geo-spatial data:

- **POI/user representation.** Via exploiting check-in data, we can learn latent representation for users and POIs, such that a user representation encodes the user's preferences on different POIs, and a POI representation embeds the information about its underlying properties.
- **Route representation.** For trajectory data, we infer and represent the *route* of each trajectory as a latent representation, which expressively encodes the information about its traveling path and the sequential dependency between different trajectory segments.

Figure 1.3: Overview of the latent representation models and their applications.

- **Region representation.** For a region that contains a set of geo-spatial objects (e.g. POIs), we encode both the semantic information (e.g., different POI categories) and the geographical information (e.g., how different categories of POIs are geographically distributed) of the geo-spatial objects into its representation.
- **Multimodal representation.** We encode the multimodal attributes (i.e., spatial, temporal and textual attributes) of spatiotemporal activity data (e.g., geo-textual posts) in latent representations, which reveals the semantic correlations of those attributes.

More importantly, we explore different types of latent representation models that can optimally capture useful information for different geo-spatial entities. We also investigate and improve existing latent representation models by considering the unique properties of different types of geo-spatial data. By doing this, we aim at effectively supporting four types of real-world applications:

- **POI recommendation**, which recommends interesting or useful POIs to LBSN users that they might want to visit.
- **Anomalous trajectory detection**, which detects trajectories that do not follow normal routes of their travel itineraries.
- **Similar region search**, which answers queries of finding regions with similar functions or semantics on a graphical space.

- **Spatiotemporal activity modeling**, which reveals the semantics and correlations of different regions, time units (e.g., hours) and keywords, where downstream applications such as spatial query processing can be built based on the activity model.

By supporting the four applications, we can greatly help people to explore massive geographical space and seek information more effectively. In particular, we are able to reveal important and useful POIs, trajectories, regions and activities that meet particular needs, which would provide new insights and improve the operational efficiency for various LBS services, businesses, governments and the entire society.

Figure 1.3 shows the latent representation models and their applications included in this dissertation. In the rest of this chapter, we give more detailed introduction about the latent representation models for modeling each type geo-spatial entity, as well as how to use the latent representations for supporting the above-mentioned applications. We summarize the main contributions in Section 1.3.

1.2 Research Problems and Methodologies

1.2.1 Point-of-Interest Recommendation

With the availability of vast amount of users' visiting history (e.g., check-ins), the problem of POI recommendations has been extensively studied in recent years. It has been found that 60%–80% of users' visits are in POIs that were not visited in the previous 30 days [NSLM12]. POI recommendations can greatly help users to find new POIs of their interests, which is beneficial to both users and businesses. However, compared with other recommendation problems (e.g., product, movie), POI recommendations face new challenges as follows:

- **Rich contexts.** First, user's mobility preference is affected by geographical distance: users usually visit POIs within a small number of activity regions (e.g., near home or work place). Second, users may visit same POIs everyday (e.g., home, work place). Third, users' preference is time-dependent. For example, a user is very likely to visit different places in early morning and late night. Fourth, users' visiting preferences might be affected by their social ties. Other types of context may include reviews on POIs, social posts on POIs, etc.
- **Data scarcity problem.** POI recommendations suffer from much worse data scarcity problem than other recommendation problems. The number of POIs visited by a user is usually only a small portion of all the POIs. For example, the density of the data used in experimental studies for POI recommendations is usually around 0.1%, while the density of Netflix data for movie recommendations is 1.2% [BK07].

POI recommendations received extensive research attention in the last five years, and many approaches have been proposed. Those studies differ in problem settings, recommendation models and evaluation data. The papers that present the newly proposed methods often report on experimental studies that suggest that the proposed methods perform better than some selected baselines on certain datasets. However, it is not clear whether they perform better on different types of data (e.g., more sparse) or different types of users (e.g., users with very few historical data). Worse still, these newly proposed methods are usually compared with other methods using similar framework (e.g., matrix factorization), and the new methods are not empirically compared with each other. Furthermore, these proposed methods may make use of different types of context information, and adopt different frameworks to capture user preferences. There is a lack of empirical study on different methods of utilizing the same type of context information or capturing user preference for POI recommendations. This state of affairs makes it difficult to decide which method is the most suitable in a particular setting. Therefore, there is a clear need for a benchmark that offers in-depth insight into the performance of the existing POI recommendation methods.

Approaches and Methodology. To meet the need, we design an evaluation procedure to evaluate 12 representative POI recommendation models, including those recent proposals, aiming to gain a general picture of POI recommendation models from multiple aspects. Specifically, these models aim at utilizing different approaches and considering different information to formulate the representations of users and POIs. We experiment these models on datasets of different sources, and different sparsity, as well as users with different sizes of historical data. This evaluation offers new insight on the relative merit of these POI recommendation methods, and the applicable scenarios of these models. We also evaluate the different recommendation techniques for user preference modeling in POI recommendations, such as Matrix Factorization, and modeling methods for context information, such as geographical context. This evaluation offers insights of which method performs better for each component, for designing more accurate POI recommendation methods in the future.

1.2.2 Anomalous Trajectory Detection

With the availability of large-scale trajectory data, automatically detecting anomalous trajectories has become a critical concern in many real-world scenarios. For example, many tourists are victims of taxi driving frauds, i.e., the taxi drivers take unnecessary detours to overcharge the passengers. Detecting such kind of malicious behavior from taxi trajectories would help to protect passengers' interests and safety. Another example is to detect real-time traffic condition from vehicle trajectories—if many vehicles choose

to take a new and different route, it may indicate that there is an accident or road construction in the normal route.

Intuitively, a trajectory is considered as anomalous if it does not follow the *normal routes* of a specific travel itinerary (i.e., from a source S to a destination D) [ZLZ⁺11, WSZ17], where a *route* represents the path taken by a trajectory in the real world. Figure 1.4 shows two examples. Suppose we have a set of trajectories between a source S_1 and a destination D_1 , and most of them follow two routes r_1 (purple line) and r_2 (blue line), respectively. We can see that the trajectory T_1 (red dashed line) is an anomaly because it does not follow the normal routes r_1 and r_2 . Another case shown in Figure 1.4 is that the normal trajectories between S_2 and D_2 usually follow r_3 and r_4 (solid lines), while an anomalous trajectory T_2 (red dashed line) switches its route from r_3 to r_4 , i.e., it does not follow r_3 or r_4 completely. This may indicate an unusual condition (e.g., road construction, car accident) on the sub-route r'_3 .

However, detecting such anomalous trajectories is non-trivial due to the following two challenges:

- **Discovering normal routes.** The key of trajectory anomaly detection is to effectively discover and represent the normal routes of trajectories. This is challenging because the normal routes of trajectories may vary across different places due to the complexity of transportation systems, and thus are difficult to be predefined and represented. Also, the sequential correlation between different segments of a normal route should also be considered, as it is important for detecting route-switching anomalies as shown in Figure 1.4.
- **Efficient online detection.** To allow actions to be promptly taken when anomalies occur, it is more desirable to support efficient *online detection* of anomalous trajectories, i.e., updating the anomaly score of a trajectory while it is being generated sequentially. However, it is challenging because trajectories are often generated at a very high speed and in a massive scale. For example, 790 million GPS points of taxi trajectories were generated in Beijing in a period of 3 months [YZXS11, YZZ⁺10a].

In recent years, anomalous trajectory detection has attracted extensive research attention and various methods have been proposed [ZLZ⁺11, CZC⁺13, LHL08, WSZ17, ZJL⁺15, ZJL⁺17, LXZ⁺17]. However, none of them solves the above two challenges simultaneously. In particular, most of the previous studies [ZLZ⁺11, LHL08, CZC⁺13, ZJL⁺15, ZJL⁺17, LXZ⁺17] use hand-crafted features that usually fail at discovering normal routes of trajectories. Their heuristic definitions of “normal routes” cannot handle the *variety*, *complexity* and *sequential correlation* of the routes traveled by real-world trajectories. For example, Zhu *et al.* [ZJL⁺15] propose to compare a target trajectory

Figure 1.4: Examples of anomalous trajectory detection. (S_1, D_1) and (S_2, D_2) are two source-destination pairs. r_1 and r_2 are two normal routes between S_1 and D_1 ; r_3 and r_4 are two normal routes between S_2 and D_2 . T_1 and T_2 are two anomalous trajectories of (S_1, D_1) and (S_2, D_2) , respectively.

with top-k popular transition sequences. However, it has been argued that such definition may easily miss some normal routes that are not very popular [LXZ⁺17]; Chen *et al.* [CZC⁺13] consider normal routes as sets of frequently-visited locations, but neglect the sequential correlation between them; Recently, Wu *et al.* [WSZ17] propose a linear probabilistic model that learns the anomalousness of trajectories from data. However, the linearity of the model is suboptimal to handle the complexity and variety of the trajectories. Worse still, there is very few research work on tackling the challenge of online detection. Chen *et al.* [CZC⁺13] are the first to study the problem of online anomalous trajectory detection. However, their proposed method needs to refer to the existing trajectories when updating the anomaly score for a trajectory being generated, which is very inefficient. Wu *et al.* [WSZ17] also try to detect potential anomalies with partial observations. However, their method needs to complete a partial trajectory with the shortest path algorithm before the detection, which increases the computational cost significantly and cannot support “on-the-fly” online detection.

Approaches and Methodology. To tackle the aforementioned challenges, we propose a novel solution for online anomaly detection of trajectories based on a deep generative model, namely Gaussian Mixture Variational Sequence AutoEncoder (GM-VSAE). GM-VSAE consists of a *route inference network* and a *route-guided generative network*, which are designed to handle the two aforementioned challenges, respectively.

To address the challenge of discovering normal routes, we propose an inference network that employs a recurrent neural network (RNN) to encode the complex sequential information underlying the trajectories, and represent the route of each trajectory as a

vector in a continuous latent space. Meanwhile, a Gaussian mixture model is jointly learned to model the probability distribution of routes represented in the latent space. The distribution characterizes the probability of *being normal* of all the routes of trajectories. We also reveal that different Gaussian components can be interpreted as different types of routes, such as streets in downtown areas and highways that connect distant locations. As a result, in the latent space, routes that are close to the center of a Gaussian component are more likely to be normal routes, because they have higher probability to be traveled.

To address the challenge of efficient online detection, we design a novel *detection-via-generation* scheme using the generative network. The intuition is that the anomalous trajectories do not follow normal routes, and thus cannot be *well-generated* from normal routes using the generative network. Therefore, we detect a trajectory by computing its likelihood of being generated from normal routes, where the likelihood can be computed online and updated in $O(1)$ time. Nevertheless, the time cost of trajectory generation using GM-VSAE would increase by a factor of the number of components in the Gaussian mixture distribution. To this end, we further propose an approximate posterior inference method that selects only one component from the mixture distribution to generate and detect a trajectory, which significantly reduces the time cost of GM-VSAE. By doing this, the proposed detection-via-generation scheme enables efficient online anomaly detection.

1.2.3 Similar Region Search

As various types of geo-spatial objects (e.g., POI and tweet) are becoming available, it is of great interests to leverage such data for understanding the information in a particular urban region. However, the explosive expansion and complication of cities make it difficult to understand the information. People may only familiar with a small part of a particular city. Their knowledge are limited within the areas they live and may not be applicable to other areas. Therefore, there is a need to *search similar regions* on geographical space, which provides an easy way to explore new areas and transfer knowledge across different areas. Consider the following scenarios of searching similar regions:

- **Scenario 1: Air pollution control.** Suppose the government has developed a solution for solving the air pollution problem in a specific region. It would be desirable to find other regions with similar pollution sources (e.g., factories), where the solution can also be applied.
- **Scenario 2: Business site selection.** Suppose a restaurant is going to start a new branch. The owners may want to look for regions that are similar to their current surrounding, where they can reuse the existing strategies and experience on the new branch.

- **Scenario 3: Improving location-based services.** Location-based services often collect user’s mobility/interests in different regions. Such information can be leveraged to suggest other similar regions for users to visit, and thus improve services like POI recommendation [LPCY17] and region recommendation [PLC17].

Therefore, it is of great significance to search similar regions on geographical space, which is fundamental to support a variety of real-world applications.

However, searching similar regions faces two challenges. The first challenge is how to model region similarity. A straightforward method is to represent a region as the sum of the attribute vectors (e.g., categories, keywords) of the objects inside, and compute the similarity between the attribute vectors of two regions. However, this method completely misses the spatial information of the objects. According to Tobler’s first law of geography [Tob70], “everything is related to everything else, but near things are more related than distant things”. The nearby objects often have underlying relationships and influence each other [EKHT14, KHT13, YZX12, WZY16], which are important for real-world applications. Let us consider the aforementioned Scenario 1 as an example. Assume there are two regions R_A and R_B and both of them contain the same number of factories and residential buildings. The factories in R_A are closed to the residential buildings, while those in R_B are not. The solution of tackling pollution in R_B may not be applicable to R_A , where we need to further consider the influence to the nearby residents. Therefore, the relative spatial locations between objects is an important dimension for measuring region similarity. In addition, spatial data is usually noisy. It is also necessary yet challenging to develop a metric that is robust to noise.

The second challenge is the efficiency of similar region search. Given a 2-d geographical space, there are numerous potential regions with various sizes and scales. Apparently, it is infeasible to exhaustively compare a query to all the regions. Furthermore, after getting the search results, users may want to modify the location of the query region and re-initiate a new query interactively. This poses a higher demand for the efficiency.

A recent study [SZH⁺10] tries to tackle the challenges. It proposes to measure the similarity between two regions as the cosine similarity of their spatial feature vectors, each of which is computed based on the distances of the objects to five predefined “reference points” in the region. It also proposes a search algorithm based on quadtree index. However, such hand-crafted spatial features do not capture the relative locations between objects, and they are not robust to noise. Moreover, the search algorithm is very time-consuming to be applied to large-scale datasets.

Approaches and Methodology. We propose a novel solution that addresses the challenges of similar region search. To address the first challenge, we leverage a deep metric learning model, namely triplet network, to learn a ranking metric for comparing regions. It extracts features using Convolutional Neural Network (CNN), which can inherently

capture local relationships between objects for similarity learning. We propose a method to generate training data with hard negative example mining, which makes the similarity metric robust to small noise and object shift. We also propose a ratio-based training loss that is robust to highly skewed spatial data. To address the second challenge with the learned similarity metric, we first reduce the time cost of computing similarity by transforming the problem. We next propose a branch and bound algorithm, namely Exact Similar Feature Region Search (ExactSFRS), that greatly reduces the search space. In addition, we also propose an approximation method, namely Approximate Similar Feature Region Search (ApproxSFRS), whose worst-case time complexity is adjustable. It can make trade-off between accuracy and efficiency to support different applications.

1.2.4 Spatiotemporal Activity Modeling

The massive volume of geo-textual data (e.g., geo-tagged tweets) offers us great opportunities for *spatiotemporal activity modeling*, which aims at characterizing different urban spaces and revealing the spatiotemporal dynamics of urban residents. In particular, it usually models the spatial, temporal and textual attributes of spatiotemporal activities. Spatiotemporal activity modeling is a fundamental problem of mining geo-textual data, which is beneficial to a wide spectrum of applications, such as discovering regional functions [YZX12], revealing geographical semantics [HAG⁺12] and detecting local event [ZLL⁺17]. For example, we can represent the functions and semantics of a region based on its activity model, and build prediction or recommendation models for human mobility inference [FLZ⁺18] or region recommendation [PLC17].

However, spatiotemporal activity is inherently different from other activities (e.g. users' online behaviors), which brings two major challenges to spatiotemporal activity modeling:

- **Temporal nonstationarity.** The geo-textual posts of social media users are *temporally nonstationary*. For example, Twitter users are usually more interested in tweeting about breaking news and interesting events, which may vary dramatically across time. As shown in Figure 1.5, the activities in a university area during different time periods are significantly different. It is non-trivial to understand the information underlying nonstationary spatiotemporal activities.
- **Data scarcity.** Due to the temporal nonstationarity, it is important to separately model the activities from different time periods (e.g., days or weeks). For example, people's activities during public holidays can be largely different from their activities during workdays. It is ineffective to fuse those activities from periods with different semantic topics. However, this often face a serious data scarcity problem [ZLL⁺18]. In particular, the data generated during a *specific time period* at a *specific location*

with the heuristics, but fail in many other scenarios. Another work from Zhang *et al.* [ZLL⁺18] aim at handling the data scarcity problem when modeling the activities of a particular time period. They augment the information of a specific time period with additional external data sources, such as WordNet¹ or auxiliary data from other cities. However, their approach requires extra efforts in collecting and processing the external information, which limits its deployment for real-world applications.

Approaches and Methodology. We propose a novel solution for spatiotemporal activity modeling, which can effectively address the above-mentioned challenges without using hand-crafted heuristics or external information. The basic intuition is that we can consider the nonstationary information behind spatiotemporal activities as a fusion of *static information* and *dynamic information*. The dynamic information represents the information of spatiotemporal activities that varies dramatically across different time periods, while the static information represents those inherent attributes that keep steady across a long timespan. For example, we can consider the nonstationary information of a shopping mall during Christmas as the fusion of its static information (i.e., “shopping”) and dynamic information (“Christmas”), which together reveal the topics like “Christmas sales” or “buying Christmas gifts”. Moreover, we consider that the static information is *transferable* across different periods. Given a specific time period, it should be aggregated with the dynamic information provided by a limited number of observed activities in the period, to overcome the data scarcity problem.

Based on the above intuitions, we formulate the spatiotemporal activity modeling as a *temporal transfer learning* problem and propose a novel end-to-end model, namely Meta Multimodal Embedding (MME) for solving it. Similar to conventional multimodal embedding methods, our method learns rich semantics of regions, hours and keywords. In addition, our MME method explicitly learns *transferable priors* for different geographical regions, which reveal their static information (e.g., region functions and geographical semantics). To fuse both static and dynamic information, we update the transferable priors containing static information with only few activities in a specific time period, which is preferable under data scarcity. Furthermore, we notice that the spatiotemporal activities of a region is usually influenced by nearby regions. To consider such spatial influence, we propose a spatial-aware multi-representational (SAMR) prior to integrate the prior knowledge of a region and its nearby regions. Based on the SAMR prior, we propose a spatial-aware Stein Variational Gradient Decent (SA-SVGD) method to optimize the model in an end-to-end manner.

¹<https://wordnet.princeton.edu/>

1.3 Summary of Contributions

In this dissertation, we explore and develop different latent representation models for different real-world applications via learning latent representations for different types of geo-spatial entities. The contributions are summarize as follows:

- We study the problem of learning user/POI representations for POI recommendation. We conduct a comprehensive experimental evaluation for POI recommendation methods, which utilize different approaches and considering different information to formulate the representations of users and POIs. We design an all-around evaluation procedure with three large-scale real-world datasets and various types of contextual information to evaluate 12 representative POI recommendation models from multiple perspectives. In particular, we experiment these models on datasets of different sources, different scales, and different sparsity. Moreover, we explore the recommendation accuracy of different methods and contextual information for different types of users. This evaluation offers new insight on the relative merit of these POI recommendation methods, and the applicable scenarios of these models. We also evaluate different recommendation techniques for user preference modeling in POI recommendations, such as Matrix Factorization, and modeling methods for context information, such as geographical context. We draw several important findings from the evaluation results, which offer new insights for deploying and designing POI recommendation methods in real-world scenarios. This work contributes the first all-around evaluation for 12 representative POI recommendation models.
- We study the problem of learning route representations for detecting anomalous trajectory. We first develop a deep generative model GM-VSAE, which is powerful at capturing the sequential information of trajectories and represent their routes in a continuous latent space. We jointly use a Gaussian mixture distribution to model the route representations, which allows us to discover different types of normal routes to support effective anomalous trajectory detection. Based on GM-VSAE, we propose a novel paradigm, *detection-via-generation*, for anomalous trajectory detection using GM-VSAE. To the best of our knowledge, we are the first to use the generation scheme to detect anomalous trajectories, which can be computed online and updated in $O(1)$ time. In addition, we propose an approximate posterior inference method that largely reduces the time cost of GM-VSAE and enables a more efficient online detection. We design an all-around evaluation procedure on two large-scale real-world trajectory datasets. We conduct extensive experiments on detecting anomalous trajectories with both partial and complete observations of the trajectories. The results show that our proposed solution is more effective than the state-of-the-art baselines and is efficient for online detection.

- We study the problem of learning region representations for efficient similar region search. We propose the first deep metric learning method to learn similarity between regions. The similarity not only considers the attributes of objects in a region, but also captures the relative locations between the objects. We propose an efficient search algorithm called ExactSFRS that solves the similar region search problem. We also propose an approximation method called ApproxSFRS that can strike a balance between accuracy and efficiency to support different applications. We apply the proposed methods to three large-scale datasets. The effectiveness experiments demonstrate our deep metric learning method is significantly better than the baselines for measuring region similarity. The efficiency experiments show that ExactSFRS is more than $10^5\times$ faster than the exhaustive search method and over $45\times$ faster than the best baseline method. Furthermore, we demonstrate ApproxSFRS can largely reduce the runtime by slightly sacrificing the accuracy.
- We study the problem of spatiotemporal activity modeling, which aims at learning multimodal representations of spatial, temporal and textual attributes of spatiotemporal activities. We are the first to formulate spatiotemporal activity modeling as a *temporal transfer learning* problem, which allows us to extract static region semantics from nonstationary data and use them for modeling activities for different time periods under data scarcity. We propose a novel method based on multimodal embedding, namely Meta Multimodal Embedding (MME), which can learn rich semantics of regions, texts and time units. More importantly, the method is able to extract static information of regions as transferable priors, which is effective in adapting to different time periods. We further extend MME to a Spatial-Aware MME (SA-MME) model, which considers the spatial influences between nearby regions. We model the spatial-aware prior knowledge of each region by leveraging the prior representations from nearby regions, and propose a novel spatial-aware Stein Variational Gradient Decent method for an end-to-end training. We conduct extensive experiments on two large-scale real-world datasets. The experimental results on region prediction task show that our proposed methods are more effective in modeling spatiotemporal activities with nonstationary geo-textual data.

Chapter 2

Literature Survey

My research is related to latent representation learning for mining geo-spatial data, which is applied to different location-based applications. In this chapter, we first give a general overview of mining different types of geo-spatial data in Section 2.1. In Section 2.2, we present detailed literature reviews of the location-based applications that we study in this dissertation. In section 2.4, we introduce different types of latent representation models that we leverage in our studies.

2.1 Mining Geo-spatial Data

Mining geo-spatial data aims at obtaining useful knowledge about different geo-spatial entities, which can benefit various location-based applications. In recent years, extensive studies are conducted for mining different types of geo-spatial data. In this section, we categorize the existing studies into four types based on the data types they utilize, where a general overview of mining each type of the geo-spatial data is introduced as follows.

User and Point-of-Interest (POI) profiling via mining check-in and geo-tagged content. Location-based social networks (LBSN) (e.g., Foursquare and Yelp) have gained rapid development in the past decade. A large number of check-ins and geo-tagged contents associated with locations are being collected from these platforms. Particularly, check-in data usually reveals people’s visiting behaviors on different POIs, and geo-tagged contents are further associated with textual comments and ratings that indicate people’s sentiments and experiences on these POIs. Therefore, many research studies have been conducted to 1) understand users’ mobility from check-in data, and 2) reveal users’ fine-grained semantics and interests by exploiting geo-tagged contents. Via mining check-in data, previous studies mainly focus on understanding users’ mobility behaviors for POI recommendations [LZX⁺14, LCL⁺15, LWSM14, YYL10, YYLL11, ZZKL17, WSOC18, MZWL18, ZCZ15, ZCYZ15, YCZ⁺16, YBZ⁺17, WFW⁺17] and mobility prediction [FLZ⁺18, WYL⁺15, LXZ⁺15, LZX13, HE13, YCM⁺13b, YCZ⁺15]. For example,

various recommendation and prediction models have been proposed for discovering different types of patterns in user mobility behaviors, such as social impact [YYLL11], constraints of geographical distance [YYLL11, WSOC18], mutual influence between different regions [LZX⁺14], POI-level and region-level user preferences [LXP⁺15], geographical hierarchy [ZXL⁺17, CPP⁺18], sequential dependencies between consecutive check-ins [ZZKL17, FLZ⁺18], neighbor-aware influence [MZWL18], regularity (i.e., periodicity) and conformity [WYL⁺15, LXZ⁺15] etc. To uncover richer semantics and provide better understanding of human mobility, geo-tagged contents (e.g., geo-tagged reviews, geo-tagged tweets) have been widely applied to further comprehend these applications [ZCZ15, ZCYZ15, YCZ⁺16, YBZ⁺17, WFW⁺17, HE13, YCM⁺13b, YCZ⁺15]. From geo-tagged contents, existing studies often extract topical regions [ZCYZ15, YCM⁺13b, YCZ⁺15, YCZ⁺16, LFYX13, LXP⁺15], location aspects [ZCYZ15] and fine-grained sentiments [YBZ⁺17, ZCZ15, WFW⁺17] of users on different POIs, which are subsequently integrated into user preferences and POI properties. These studies on mining check-ins and geo-tagged contents can greatly help LBSN users to explore vast and complicated geographical space, which is beneficial to both users and businesses.

Trajectory data mining. Trajectory data is ubiquitous and usually contains rich information about the mobility behaviors of pedestrians and vehicles, which is of immense potential to be exploited for real-world applications, such as traffic management, taxi scheduling and optimization of public transportation. It draws a lot of attentions from both academia and industry and various research problems have been studied [ZCZY14]. In particular, the existing studies on trajectory data mining can be categorized into different topics. First, *trajectory preprocessing* is a vital phase in trajectory data mining, as the raw trajectories collected from GPS devices often contain uncertainty, noise or other issues that may cause problems and troubles for downstream applications. To this end, different types of preprocessing methods have been proposed, such as trajectory calibration [Man93, LLL⁺12, SZW⁺13, SZH⁺15], map matching [HQY⁺13, BPSW05, LZZ⁺09, YZZ⁺10b] and privacy-aware trajectory publication [HCM⁺15, CAC12, CFDS12, SJK⁺13, TM08]. Second, the massiveness of trajectory data also calls an efficient *trajectory management*. To meet this need, many trajectory management systems have been built [DCG⁺18, Ric18, BMNS08, CMWM10, WZX⁺14, XMC⁺16], which optimize the indexing and storage of trajectory data. Subsequently, these systems can also be applied to support efficient *trajectory query processing* [MYQZ09, AEA⁺18], such as trajectory similarity search and join [BHKT05, BT06, CP09, DTS08, LZC⁺18a, YCZB19, CÖO05, XLP17]. Moreover, to further extract useful knowledge from trajectory data, a vast number of studies attempt to *mine trajectory patterns*, such as mining frequent patterns [LZC⁺11, LCI09], topical & semantic patterns [YLWT11, KHY15], gathering patterns [ZZY⁺13] and convey patterns [JYZ⁺08,

JSZ08]. The knowledge gained from trajectory pattern mining also enables many downstream applications, such as trajectory clustering [YSZ⁺17, LHLG08, LLLH10, LHW07], classification [LHLG08, ZCL⁺10, ZGT⁺18, BCTH12], prediction [MZZ⁺18, WHKD12], regression (e.g., estimating travel time) [LCSC19, IS11, WFZ⁺18, ZN13], and anomalous trajectory detection [LHL08, ZLZ⁺11, ZJL⁺15, LXZ⁺17, CZC⁺13, GXLZ11, SWX⁺18, WSZ17].

Region analysis and profiling. With the rapid progress of urbanization all over the world, geographical spaces of urban cities are becoming increasingly complex for people to understand, which creates a need of analyzing and discovering the functions, semantics and topics of different geographical regions. For example, geographical topic discovery has gained a lot of attentions in this field, where many geographical topic models have been proposed [ZCYZ15, YCH⁺11, HAG⁺12, YCM⁺13b, YCZ⁺15] to represent regions as multivariate distribution over a geographical space, and the topics of regions as probability distributions over keywords. Moreover, multimodal data from multiple sources have also been leveraged to discover region functions (e.g., education area, recreation area) [YZX12, YFL⁺18, ZZG⁺18]. Subsequently, these semantical and functional information of regions can be applied to applications like popularity prediction [ZZG⁺18, FWD⁺19] and crime prediction [WL17, ZT17, HZZC18]. In recent years, some other novel tasks for region analysis have also emerged, including similar region search [SZH⁺10], detecting periodic regions [YZZ⁺17], region partitioning [WJW⁺19], regions profiling from map query logs [SZZ⁺18], region recommendation [PLC17], regional data refinement [TIT⁺18] and discovering transit-oriented development regions [KXM⁺19]. Meanwhile, new machine learning techniques, such as graph embedding and deep learning, have also been employed for improving the effectiveness or provide new insights for region analysis and profiling [WL17, FWD⁺19, WFZ⁺18].

Spatiotemporal activity modeling. Geo-tagged posts (e.g., geo-tagged tweets) are a type of geo-spatial data associated with GPS coordinates, timestamps and textual contents, which are being generated as a continual data stream. Different from geo-tagged reviews that are usually recorded *after* users visiting POIs, geo-tagged posts can better reveal the *real-time* semantics on geographical space, and thus can be considered as a useful data source for modeling spatiotemporal activities and real-time local events. In particular, spatiotemporal activity modeling is a fundamental problem of mining geo-tagged posts, which aims at understanding the correlations among spatial, temporal and textual attributes of these data. Previous studies [Siz10, WWXM07, HAG⁺12, KKSS14, YCH⁺11, YCM⁺13b] on geographical topic mining with geo-tagged posts can be considered as a special case of spatiotemporal activity modeling, while temporal dimension is usually unexplored in these studies. Recently, Zhang *et al.* [ZZY⁺17a, ZLL⁺18, ZZY⁺17b] propose several embedding methods to capture the relationships among regions (i.e., spatial attributes), hours (i.e., temporal attributes) and keywords

(i.e., textual attributes) via exploiting geo-tagged posts, which show promising performances for spatial query processing. In addition to spatiotemporal activity modeling, two other real-world applications, i.e., spatiotemporal event detection [WK13, ZLL⁺17, ZLY⁺18, ZZY⁺16a, ASG13, FZZ⁺15, KH15, LWS11, SOM10] and forecasting [ZZB⁺17, ZCLR15, ZYC⁺16, ZCY19, ZCLR16, ZSY⁺15], have also been extensively studied, where the targets are detecting and predicting unusual spatiotemporal activities revealed by the geo-tagged posts, respectively.

2.2 Location-based Applications

In this section, we review existing researches that are related to the location-based applications included in this dissertation, i.e., POI recommendation (Section 2.2.1), anomalous trajectory detection (Section 2.2.2), similar region search (Section 2.3) and spatiotemporal activity modeling (Section 2.3.4).

2.2.1 POI Recommendation

2.2.1.1 Conventional POI Recommendation

In recent years, extensive studies have been conducted for POI recommendations, which use different approaches and different information to represent user preferences and POI properties. Some early studies usually leverage User-based Collaborative Filtering method [SFHS07]. For example, Ye et al. applied User-based CF (UCF) and Friend-based CF (FCF) to model user preference and include social influence, respectively. Subsequently, UCF and FCF have been utilized in many POI recommendation models [YYL10, YYLL11, ZC13, ZCL14, ZCZ15] as the based model, where various types of context information are integrated (e.g., geographical influence, content information). Additionally, item-based CF (ICF) is also used for POI recommendations [LSEM12, SLEM14] and ICF is shown to perform worse than UCF [YYLL11]. To further explore the relationships and common interests among users, link-based methods are also applied for POI recommendations. For example, Ying et al. [YKTL14] propose a HITS-based model that considers the links between users and POIs, and Noulas et al. [NSLM12] include both user-user friendships and user-POI links. Wang et al. [WTM13] further include relations between spatially similar users as the edges in user-user graph, and use them for recommendations.

With the development of conventional recommender systems, Matrix Factorization (MF) [KBV09] has become the most popular techniques and usually achieve the state-of-the-art performances for conventional item recommendations in e-commerce websites. For POI recommendation, there are also various MF methods being applied, such as

Table 2.1: Summary of existing POI recommendation papers (sorted by publication years)

Related work	Problem					Methodology					Information					
	POI rec.	Time-aware	Next POI	In-/Out-of-town	Others	Link-based	CF	Factorization	Probabilistic	Hybrid	Geographical	Social	Textual	Categorical	Sequential	Temporal
Ye et al. - GIS'10 [YYL10]	✓						✓					✓				
Ye et al. - SIGIR'11 [YYLL11]	✓						✓			✓	✓	✓				
Noulas et al. - SocialCom'12 [NSLM12]	✓					✓						✓				
Cheng et al. - AAAI'12 [CYKL12]	✓								✓	✓		✓				
Levandoski et al. - ICDE'12 [LSEM12]	✓						✓				✓					
Gao et al. - RecSys'13 [GTHL13]	✓							✓								✓
Hu et al. - RecSys'13 [HE13]	✓								✓				✓			
Zhang et al. - GIS'13 [ZC13]	✓						✓			✓	✓	✓				
Wang et al. - GIS'13 [WTM13]	✓					✓					✓	✓				
Yuan et al. - SIGIR'13 [YCM ⁺ 13a]		✓					✓				✓					✓
Liu et al. - KDD'13 [LFYX13]	✓							✓			✓		✓			
Liu et al. - CIKM'13 [LLAM13]					✓			✓			✓			✓		
Ference et al. - CIKM'13 [FYL13]				✓			✓				✓	✓				
Kurashima et al. - WSDM'13 [KIH ⁺ 13]	✓		✓						✓		✓		✓			
Cheng et al. - IJCAI'13 [CYLK13]			✓					✓			✓				✓	
Yin et al. - KDD'13 [YSC ⁺ 13]				✓					✓		✓		✓			
Yuan et al. - KDD'13 [YCM ⁺ 13b]					✓				✓		✓		✓		✓	
Zhang et al. - GIS'14 [ZCL14]	✓						✓			✓	✓	✓			✓	
Liu et al. - CIKM'14 [LWSM14]	✓							✓		✓	✓					
Yuan et al. - CIKM'14 [YCS14]		✓				✓					✓					✓
Lian et al. - KDD'14 [LZX ⁺ 14]	✓							✓			✓					
Ying et al. - TIST'14 [YKTL14]	✓					✓						✓		✓		
Sarwat et al. - TKDE'14 [SLEM14]	✓						✓				✓					
Li et al. - SIGIR'15 [LCL ⁺ 15]	✓	✓									✓					✓
Zhang et al. - SIGIR'15 [ZC15a]	✓							✓			✓	✓		✓		
Zhang et al. - TIST'15 [ZC15b]	✓									✓	✓				✓	✓
Zhang et al. - CIKM'15 [ZCZ15]	✓						✓			✓	✓	✓				
Liu et al. - TKDE'15 [LXP ⁺ 15]	✓								✓		✓					
Gao et al. - AAAI'15 [GTHL15]	✓							✓					✓			
Lian et al. - ICDM'15 [LGZ ⁺ 15]	✓							✓					✓			
Li et al. - ICDM'15 [LHZG15]	✓							✓			✓	✓				
Zhao et al. - ICDE'15 [ZCYZ15]	✓								✓				✓	✓		
Feng et al. - IJCAI'15 [FLZ ⁺ 15]			✓						✓		✓				✓	
Wang et al. - KDD'15 [WYC ⁺ 15]				✓					✓		✓		✓	✓		
Yin et al. - CIKM'15 [YZS ⁺ 15]				✓					✓		✓		✓	✓		✓
Yuan et al. - TOIS'15 [YCZ ⁺ 15]					✓				✓		✓		✓		✓	
Chen et al. - AAAI'15 [CZC ⁺ 15]					✓		✓				✓		✓			✓
Li et al. - KDD'16 [LGRZ16]	✓							✓			✓	✓		✓		
Zhao et al. - AAAI'16 [ZZY ⁺ 16b]			✓					✓			✓					✓
He et al. - AAAI'16 [HLL ⁺ 16]			✓					✓			✓				✓	
Liu et al. - KDD'16 [LLL ⁺ 16]			✓						✓						✓	✓

conventional MF [GTHL13], Weighted MF [LWSM14, LZX⁺14, LGRZ16], Bayesian Non-negative MF [LFYX13] and ranking-based MF [LCL⁺15]. Most of them incorporate region-level or POI-level geographical influence to further improve their effectiveness. In addition, Bhargava et al. [BPZL15] also apply tensor decomposition to model user preference.

Moreover, probabilistic models are usually represented graphically to describe the interplay among variables, such as the mutual effects between geographical influence and user interests in POI recommendations. On the one hand, Poisson Factor Model (PFM) is first adopted for POI recommendation [CYKL12] and then extended to include context information [LXP⁺15]; On the other hand, spatial topic models are used in modeling user's latent interests and integrating users' context preferences [KIH⁺13, HE13,

LFYX13, ZCYZ15].

2.2.1.2 Other POI Recommendation Problems

We next introduce three variants of the POI recommendation problem, which take additional information as part of input. They aim to fulfill more specific needs of users.

Next POI Recommendation. Given a user and his/her current location, next POI recommendation aims at *recommending new POIs that are likely to be visited by the user in the next time interval (e.g., in the next 6 hours)* [CYLK13, KIH⁺13, FLZ⁺15, ZZY⁺16b, HLL⁺16, LLL⁺16]. Most of the next POI recommendation models employ sequential information between consecutive check-ins to recommend. Cheng et al. [CYLK13] embed personalized Markov Chain in modeling users' transitions between POIs. Kurashima et al. [KIH⁺13] extend their model, which is based on probabilistic latent semantic analysis (PLSA), for next POI recommendations by considering the Euclidean distance from current location to the candidate POIs. Feng et al. [FLZ⁺15] propose a personalized ranking metric embedding method, considering transition probability between POIs as the Euclidean distance between their latent representation in a low-dimensional space. Zhao et al. [ZZY⁺16b] and He et al. [HLL⁺16] propose ranking-based tensor factorization models for next POI recommendations.

Additionally, models developed for location prediction can also be applied to next POI recommendation [MPTG09, SMM⁺11, YCM⁺13b, YCZ⁺15, MNSB15, WYL⁺15]. Monreale et al. [MPTG09] use a decision tree that is learned from users' trajectory patterns to predict their next visit POIs; Scellato et al. [SMM⁺11] propose a location prediction approach based on nonlinear time series analysis; Yuan et al. [YCM⁺13b, YCZ⁺15] propose a unified probabilistic graphical model that can perform location prediction; Muntean et al. [MNSB15] employ Gradient Boosted Regression Trees and Ranking Support Vector Machine (SVM) on tourism related data to predict the mobility behavior of tourists; Wang et al. [WYL⁺15] design a MF-based method to model users' conforming and regular behaviors for prediction.

Time-aware POI Recommendation. Considering that user preference varies with time, given a user a time (e.g., 5 p.m.), time-aware POI recommendation *returns new POIs that are most likely to be visited at the time to users* [YCM⁺13a, YCS14, LCL⁺15, DAM⁺15]. Time-aware POI recommendation is first proposed by Yuan et al. [YCM⁺13a], and a CF method with time-dependent similarity between users is introduced to solve the problem. Subsequently, Yuan et al. [YCS14] propose a Geographical-Temporal influences Aware Graph method to further improve the recommendation accuracy. Li et al. [LCL⁺15] extend their POI recommendation model by including time-related latent factors, and a tensor decomposition is applied for time-aware recommendations. Deveaud et al. [DAM⁺15] propose a POI-centric method to model the time-aware properties of POIs.

In-town/Out-of-town POI Recommendation. Given a user’s hometown and current location or city, in-town/out-of-town POI recommendation *returns new POIs to the user, using different recommendation strategies when the user is in-town or out-of-town* [FYL13, YSC⁺13, YZS⁺15, WYC⁺15]. Ference et al. [FYL13] first show that the performances of POI recommendation models for out-of-town users are worse than in-town users by a large margin, and use a unified CF method with different parameter settings for in-town and out-of-town recommendations. Yin et al. [YSC⁺13, YZS⁺15] use local users’ preferences and POI contents for out-of-town users, to overcome users’ “cold-start” problem in a new city. Wang et al. [WYC⁺15] design a novel geographical sparse additive generative model and apply different latent topic distributions for both in-town and out-of-town users. Except for the earliest work [FYL13], all the other models for in-town/out-of-town POI recommendation rely on content information (e.g., tags and categories).

There are other types of POI recommendation, such as category-aware [LLAM13] and requirement-aware [YCM⁺13b, YCZ⁺15] POI recommendation.

2.2.2 Detecting Anomalous Trajectory

We introduce two categories of studies on anomalous trajectory detection. The first category is usually based on hand-crafted features and contains two steps: 1) defining “normal routes” with representative trajectories and 2) comparing a target trajectory with representative trajectories based on distance or density metrics. The second type is learning-based methods, which try to automatically extract useful information from data. We elaborate each category of methods in the following and then introduce recent advances in general anomaly detection using deep representation learning.

2.2.2.1 Metric-based Anomalous Trajectory Detection

An early study [LHL08] propose to detect anomalous trajectory segments via computing the distance of each segment of the target trajectory to other trajectories. Zhang *et al.* [ZLZ⁺11] propose to detect trajectory anomalies by checking how many of the locations in a trajectory are not frequently covered by other trajectories with the same source and destination. This method is further extended by considering the order of the locations in a trajectory [CZC⁺13]. Lv *et al.* [LXZ⁺17] find representative trajectories using clustering algorithms and compare a target trajectory with the representatives using a novel edit distance, which is similar to the dynamic time wrapping (DTW) metric. Zhu *et al.* [ZJL⁺15] also designed a new edit distance for trajectory comparison, while the representative trajectories are found by mining frequent transition patterns between locations. In addition, Ge *et al.* [GXLZ11] tried to combine distance and density metrics to detect taxi frauds. However, this category of methods have two drawbacks. Firstly, their

heuristic definitions of representative trajectories are usually hand-crafted with many predefined parameters, such as the frequency or density thresholds of being “representative”. As trajectories between different places are usually very different, those definitions might only be effective for several scenarios but fail for many other situations. Secondly, the trajectory comparison is usually time-consuming and cannot support efficient online detection.

2.2.2.2 Learning-based Anomalous Trajectory Detection

Wu *et al.* [WSZ17] leveraged a probabilistic model that learns to detect anomalies for both the whole trajectories and partial trajectories. However, the linearity of the model makes it hard to capture complex information behind the data. Also, it does not take sequential information of trajectories into consideration. Another recent proposal [SWX⁺18] develops a supervised model with recurrent neural network (RNN) for anomalous trajectory detection. However, the model requires labeled data that is usually unavailable in real applications. In contrast, our method employs an unsupervised learning method that does not require any labeled data, where the anomalous trajectories are considered as those trajectories that are significantly different from the majority. Other studies [GXZ⁺10, YCRW14, BYR16] that use different problem settings from ours, such as detecting visually and temporally anomalous trajectories, will not be discussed further.

2.2.2.3 Anomaly Detection with Deep Representation Learning

Due to the powerful capacity of extracting complex information from large-scale data, many deep learning methods have been applied for anomaly detection [MRA⁺16, ZP17, CSAT17, ZSM⁺18]. Most of them leverage autoencoder (AE) to learn the latent features of data and compute anomaly scores using reconstruction (i.e., decoding) error. However, conventional autoencoders are both ineffective and inefficient for online anomalous trajectory detection. First, they lack the means of mining useful information about trajectory anomalousness in the latent embedding space, i.e., they may only embed trajectories in a low-dimensional space, without exploring which part in the latent space represents *normal* and which part stands for *anomalous*. Second, as a new GPS point of a trajectory comes, it needs to re-encode and re-decode the whole trajectory to update the anomaly score, and thus is inefficient for online detection.

2.3 Similar Region Search

2.3.1 Measuring Region Similarity

There are several recent proposals trying to measure the similarity between regions. Sheng et al. [SZH⁺10] first studied the similar region search problem. They measure the

similarity between two regions as the cosine similarity of their spatial feature vectors. The spatial feature vector of a region is defined as the average distances between the objects and five predefined reference points. Le Falher et al. [LFGM15] and Kafsi [KCTS15] et al. both studied the problem of finding similar neighborhoods across different cities. The neighborhoods are predefined areas within cities. Besides, Preoțiuc-Pietro [PPCY13] et al. studied a problem of measuring city similarity, considering a city as a large region and representing it as grids or predefined neighborhoods for comparison.

Our work differs from the existing studies [KCTS15, LFGM15] in measuring similarity. They only consider the aggregated object attributes of a region, while our work measures the similarity considering both the spatial locations and attributes of objects. Sheng et al.’s work [SZH⁺10] is the most germane to our paper. It also considers the spatial locations of objects. However, its similarity modeling is very sensitive to noise. The similarity can easily be affected by adding, deleting or shifting few objects in the region, which are commonly-seen noise in spatial object databases. Worse still, it only considers the spatial distance of the objects to the center and boundary of the region, missing how different types of objects are co-located. This means the underlying influence and relationships between objects are not included in the region similarity.

There are also some work focus on modeling the relationships between regions [WKGL16, ZT17, WL17, YWK⁺18, YTW⁺18], where the relationships (e.g., traffic flow) can be used to define similarity.

2.3.2 Efficient Search for Similar Regions

Many of the above-mentioned studies [KCTS15, PPCY13, WKGL16, ZT17, WL17] aim at measuring the similarity between areas with predefined boundary (e.g., neighborhoods, cities). They are different from our proposal, which supports searching regions with various sizes and scales at an arbitrary location on the map. Another recent proposal [SZH⁺10] that supports searching similar regions proposes a quadtree-based search method. It stores the spatial objects using a quadtree and consider the tree nodes as candidate regions. When a query comes, it searches top- N similar tree nodes (i.e., regions) and then expand their sizes greedily to further improve the results. However, the region expansion is very time-consuming. It needs to recompute the similarity every time a region expands, while our proposed search method can directly localize the most similar regions without adjusting the size and recomputing the the similarity. Also, the quadtree based method gives no guarantee about the running time, while our proposed approximation method has a worst-case complexity guarantee.

There are also some studies in Computer Vision about finding regions on images, namely visual instance retrieval [RSCM14, TSJ15]. They usually sample many candidate regions on an image and check them exhaustively. However, this is not applicable on large geographical space, which contains at least billions of region samples to consider regions with various scales at arbitrary locations.

2.3.3 Metric Learning

For applications that rely on distances or similarities, such as information retrieval and clustering, learning a good metric is important [K⁺13]. Recently, Convolutional Neural Networks (CNNs) based metric learning has achieved great performance on many Computer Vision tasks, such as image classification [WHG17] and human re-identification [HBL17]. The ability of capturing local spatial correlation makes CNN also inherently applicable to geo-spatial data. Our proposed model is based on a triplet architecture of CNNs, namely triplet network [HA15], which learns a ranking metric for region comparison. To apply triplet network on spatial data, we design a method to generate training data with hard negative example mining [WZL17]. We also use a ratio-based training loss that is robust to highly skewed spatial data.

2.3.4 Spatiotemporal Activity Modeling

2.3.4.1 Spatiotemporal Topic Modeling for Geo-tagged Social Media

Mining geo-tagged social media has become an important problem for many real-world applications. The early studies of mining geo-tagged social media data [Siz10, WWXM07, HAG⁺12, KKSS14, YCH⁺11, YCM⁺13b] focus on learning topics from geo-tagged social media using geographical topic models, which are mainly based on Latent Dirichlet Allocation (LDA) [BNJ03]. In particular, Wang *et al.* [WWXM07] are the first to apply topic models for modeling geo-tagged data, which explicitly model the relationships between locations and words. Sizov [Siz10] proposes a multimodal Bayesian model that integrates spatial semantics. More recently, Yin *et al.* [YCH⁺11] incorporate region discovery jointly with topic modeling, which can find various semantical regions on geographical space. Moreover, Hong *et al.* [HAG⁺12] further extend the geographical topic model considering users' interests and preferences on different locations. Besides, Quan *et al.* [YCM⁺13b] also propose a topic model that mines the mobility behaviors of social media users from multiple perspectives, which is further improved to a nonparametric Bayesian approach [YCZ⁺15]. These methods are mainly focus on discovering topics for different regions and revealing preferences for social media users.

2.3.4.2 Spatiotemporal Activity Modeling with Multimodal Embedding

Multimodal embedding [PCS18, MXJY16] is a technique that learns latent embeddings for different data modalities to capture the cross-modal relationships. In recent years, there are several studies aim at applying multimodal embedding for spatiotemporal activity modeling. In particular, Zhang *et al.* [ZZY⁺17a] are the first to consider different spatiotemporal attributes, i.e., regions, hours and keywords, as different data modalities, and try to learn their embeddings via exploiting spatiotemporal activity data. However,

their model can hardly be used to model the spatiotemporal activity of a specific time periods. To this end, Zhang *et al.* [ZZY⁺17b] further propose a recency-aware model to capture the up-to-date information at each time period. The recency-aware buffer stores some recent historical data to alleviate the data scarcity problem when modeling the activities in a small time period. Also, an alternative way to handle the data scarcity when modeling period-specific activities is introduced in a very recent work [ZLL⁺18], which proposes to transfer useful knowledge from external sources. Compared with using geographical topic models for spatiotemporal activity modeling, this line of researches are benefited from the flexibility of multimodal embedding technique, which does not need to pose any distributional assumptions on the data. Therefore, they have achieved the state-of-the-art performances for modeling spatiotemporal activities.

2.4 Latent Representation Models

Latent representation models are a type of machine learning models that extract useful knowledge from data and encode it as latent representations (i.e., vectors in a continuous latent space), which can be useful for building classifiers or other predictors. In this section, we review the latent representation models that are investigated and leveraged in our studies.

2.4.1 Matrix Factorization

Matrix Factorization (MF) [KBV09] is a popular technique for developing recommender systems. The objective of MF is to learn latent representations of users and items (e.g., POIs in POI recommendation), such that a similarity measure (e.g., inner product) can be applied to the representations to reveal the interests of users on different items. In particular, conventional MF method first constructs a user-item matrix $\mathbf{C} \in \mathbb{R}^{M \times N}$, where each entry C_{ij} represents the historical behaviors (e.g., rating, visiting frequency) of a user i on the item j . Subsequently, it decomposes $\mathbf{C}$ into a user representation matrix $\mathbf{U} \in \mathbb{R}^{M \times \mathcal{K}}$ and a item representation matrix $\mathbf{L} \in \mathbb{R}^{N \times \mathcal{K}}$, where M , N and $\mathcal{K}$ are the number of users, items and latent factors, respectively. The latent representations of each user i and item j are represented by $\mathbf{u}_i$ and $\mathbf{l}_j$, respectively. Here, $\mathbf{u}_i$ and $\mathbf{l}_j$ are the i -th and j -th rows of $\mathbf{U}$ and $\mathbf{L}$, respectively. The recommendation score of user i for item j is thus modeled as the inner product $\hat{C}_{ij} = \mathbf{u}_i \mathbf{l}_j^\top$, and the objective function is formulated as:

$$\min_{\mathbf{U}, \mathbf{L}} \|\mathbf{C} - \mathbf{U}\mathbf{L}^\top\|_F^2 + \lambda_1 \|\mathbf{U}\|_F^2 + \lambda_2 \|\mathbf{L}\|_F^2,$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix, λ_1 and λ_2 are regularization parameters.

In recent years, many variants of MF have also been developed to adapt different types of data. For example, Weighted Matrix Factorization [HKV08, PZC⁺08] constructs a 0-1 user-item matrix and assigns different weights to different entries based on the interaction frequencies between users and items. It reports better recommendation accuracy when modeling implicit feedback data (e.g., purchase and browsing history). Another implicit feedback model, i.e., Bayesian Personalized Ranking method [RFGST09], designs a novel ranking-based objective function, such that the item recommendations for a given user are considered as a ranking problem, where the items that are more likely to be preferred by the user should be ranked higher in the recommendation list.

2.4.2 Recurrent Neural Network

Recurrent Neural Network (RNN) [GMH13] is a class of deep neural networks for processing sequence data. Given a sequence $x_1, x_2, \dots, x_T$, it performs the same non-linear operation at each time step $t = 1, 2, \dots, T$, which recurrently encodes the sequence as hidden state vectors (i.e., latent representations). We next introduce two widely-used implementations of RNN, i.e., Long Short-Term Memory (LSTM) [HS97] and Gated Recurrent Unit (GRU) [CVMG⁺14].

Long Short-Term Memory (LSTM) is a variant of RNN that aims at tackling gradient vanishing and exploding problems. It is equipped with three gating functions when processing the sequence at each time step. In particular, the non-linear operations of LSTM at time step t can be defined as

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \\ f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \\ c_t &= f_t \odot c_{t-1} + i_t \odot \sigma(W_c \cdot [h_{t-1}, x_t] + b_c), \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \\ h_t &= o_t \odot \tanh(c_t), \end{aligned}$$

where i_t , f_t and o_t are named as the input, forget and output gates. Here, W and b represent the weight matrix and bias of the non-linear operations, $[\cdot, \cdot]$ denotes vector concatenation, σ represents a sigmoid activation function and $\odot$ represents Hadamard product. By doing these, information of sequence $x_1, x_2, \dots, x_t$ would be captured in the hidden state h_t and the cell state c_t .

Gated Recurrent Unit (GRU) is a simplified variant of LSTM, which only relies on two gating functions and does not have cell state. In particular, it introduces an update

gate and a reset gate that work at each time step as

$$\begin{aligned} z_t &= \sigma(W_z \cdot [h_{t-1}, x_t] + b_z), \\ r_t &= \sigma(W_r \cdot [h_{t-1}, x_t] + b_r), \\ \tilde{h}_t &= \tanh(W_h \cdot [r_t \odot h_{t-1}, x_t] + b_h), \\ h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t. \end{aligned}$$

At time step t , h_t is the hidden state (i.e., latent representation) that encodes sequential information.

2.4.3 Convolutional Neural Network

Convolutional Neural Network [KSH12] is a class of deep neural networks for processing data with grid-like topology, such as image and video data. In particular, it usually consists of multiple convolutional layers, each of which contains a set of learnable filters (i.e., kernels) (each denoted as K) that perform convolution function to extract features from the local areas of its input. For example, the convolution function performed by the first convolutional layer on the input X is defined as

$$F_{i,j} = \sum_m \sum_n X_{i-m,j-n} K_{m,n}, \quad (2.1)$$

where m and n are the indexes for the locations in a filter; i and j represent a location on the input grids X and the output grids F . The output F of each convolutional layer is called feature map, which is usually considered as the input for the next convolutional layer. Therefore, a CNN usually stacks multiple convolutional layers and produces high-level non-linear abstraction of the original input data. There are extensive studies that explore different architectures of building CNN models for different tasks [LCJB⁺89, KSH12, SLJ⁺15, SZ14, HZRS16, Cho17, HZC⁺17, HA15, BGL⁺94, Gir15, RHGS15, HGDG17].

2.4.4 Word2vec

Word2vec [MSC⁺13] is a popular technique to learn high-quality distributed (latent) representations of words, where the representations encode syntactic and semantic meaning of words. Compare to other neural architecture of learning word representations, word2vec does not include dense matrix multiplications and is therefore very efficient. Previous study [MSC⁺13] also find interesting constitutionality of the learned word representations, which provides interpretability to the model. A commonly-used implementation of word2vec is the skip-gram architecture, whose objective is to predict the

surrounding context words w_o given a target word w_i , i.e.,

$$\arg \max \sum_{w_o \in C(i)} \log p(w_o | w_i), \quad \text{where} \quad p(w_o | w_i) = \frac{\exp(w_o^\top w_i)}{\sum_{w \in \mathcal{W}} \exp(w^\top w_i)}. \quad (2.2)$$

In Eq. 2.2, w_i , w_o and w are the latent vectors, $C(i)$ represents the set of context words of w_i and $\mathcal{W}$ represents all the possible words in the corpus. To avoid expensive computation for $\sum_{w \in \mathcal{W}} \exp(w^\top w_i)$, hierarchical softmax [MB05] and negative sampling [GH12] have been applied to significantly reduce the training time cost. The latent word representations can therefore be applied to downstream tasks in natural language processing.

In addition to learning word representations, the idea of word2vec has also been widely employed for modeling other types of data. Two typical fields are graph representation learning [PARS14, GL16, TQW⁺15, HLH17, CWPZ18, RSF17, HYL17] and multimodal representation learning [DMMD18, ZZY⁺17a, ZLL⁺18, ZZY⁺17b, ZLL⁺17]. In graph representation learning, many studies try to define the contextual relationships between the nodes in a graph, such as considering the neighbors of a given node as its context or conducting random walk to sample context nodes. By doing this, they construct objectives that are similar to skip-gram and conduct optimization to learn node representations, which capture the structure and topological information of the graph. In multimodal representation learning, previous studies usually consider the co-occurrence of different data modalities (e.g., text and image) in the same data instances. Therefore, the objective is to predict the value of a particular modality given the values of other data modalities that appear in the same data instances. Multimodal representation learning is powerful in capturing the relationships across different data modalities.

Chapter 3

Point-of-Interest Recommendation

With the availability of vast amount of users' visiting history (e.g., check-ins), point-of-interest (POI) recommendation has become a popular location-based application, which helps people to explore interesting and useful places. As such, various approaches have been proposed in the recent decade to effectively support POI recommendations. However, they usually represent users and POIs with different models and leverage different types of context information. Their experiments are also conducted on different datasets (e.g., datasets with different sparsity). Thus, it is unclear which representation model is the most suitable in a particular scenario for POI recommendations.

Motivated by this, we contribute a systematic evaluation for 12 representative POI recommendation methods, which employ different types of representation models and use different kinds of context information. We design an all-round evaluation procedure using real-world large-scale datasets, which provides several new insights for using the existing methods, as well as designing new approaches for POI recommendations.

This chapter is organized as follows. Section 3.1 first gives an introduction of POI recommendations. Section 3.2 categorizes and presents representative models included in our evaluation. Section 3.3 and 3.4 present the experiments for POI recommendation models, from which some notable findings are uncovered.

3.1 POI Recommendation

Given a set of POIs $\mathcal{L}$, and a set of users $\mathcal{U}$ each associated with a set of POIs $\mathcal{L}^u$ visited by the user, the problem of POI recommendations is to recommend for each user $u \in \mathcal{U}$ new POIs, i.e., in the set of $\mathcal{L}/\mathcal{L}^u$, that are likely to be visited by user u .

This chapter is published as *Yiding Liu, Tuan-Anh Nguyen Pham, Gao Cong, Quan Yuan. An Experimental Evaluation of Point-of-interest Recommendation in Location-based Social Networks. In Proceedings of the VLDB Endowment (VLDB 2017), Volume 10, Issue 10, Pages 1010-1021, June 2017 [LPCY17].*

POI recommendations are significantly influenced by rich contexts such as geographical distance, social relations and time. To demonstrate their influence, we next show some statistical analysis results on the datasets from three LBSNs, i.e., Gowalla, Foursquare and Yelp. The details of these data are introduced in Section 3.3.1.

Observation 1: spatial influence. We consider user’s consecutive check-ins as transitions between POIs, and compute the distribution of transition distances of users. Figure 3.1(a) shows the cumulative distribution functions (CDFs) of transition distances in Gowalla, Foursquare and Yelp. We can see that all the three curves rise dramatically when the distance is small. In Gowalla and Foursquare, 90% of users’ transition distances are less than 50km. These indicate that users tend to visit nearby POIs.

Observation 2: social influence. We choose the Gowalla users whose check-ins are within Austin, Texas, and compute the numbers of common check-in POIs between friends and between randomly sampled non-friends. For friends and non-friends, the average numbers of common check-in POIs are 5.69 and 0.91, respectively. Particularly, Figure 3.1(b) shows the CDFs of the number of common check-in POIs. We can see that more than 60% non-friends have no common check-in POI, while the number is only 16% between friends. Moreover, around 85% friends have fewer than 10 common check-in POIs and over 80% non-friends have only 1 or no check-in POI in common. These indicate that most friends have small overlapping on their check-in POIs, but the overlapping is significantly larger than non-friends.

Observation 3: temporal influence. On the one hand, two users may behave differently with respect to time. For example, one often checks in restaurants during lunch time, while the other likes bars and often checks in at midnight. On the other hand, different POIs have different opening hours and peak hours (e.g. restaurants vs. bars), and thus their check-in patterns over time are also different.

3.2 Models for Evaluation

In this section, we introduce 12 POI recommendation models included in the evaluation. They represent the state-of-the-art methods, and include those recent proposals. They cover (i) four popular recommendation techniques and (ii) five types of context information such as geographical influence. These models are summarized in Table 3.1. Next, we group them based on their recommendation techniques, and introduce how they model and incorporate context information.

Figure 3.1: Spatial and social influence.

3.2.1 Matrix Factorization Models

Matrix Factorization (MF) [KBV09] decomposes the check-in matrix $\mathbf{C} \in \mathbb{R}^{M \times N}$ into user matrix $\mathbf{U} \in \mathbb{R}^{M \times \mathcal{K}}$ and POI matrix $\mathbf{L} \in \mathbb{R}^{N \times \mathcal{K}}$, where M , N and $\mathcal{K}$ are the number of users, POIs and latent factors, respectively. Latent features of each user i and POI j are represented by $\mathbf{u}_i$ and $\mathbf{l}_j$. The recommendation score of user i for POI j is thus modeled as the inner product $\hat{C}_{ij} = \mathbf{u}_i \mathbf{l}_j^\top$, and the objective function is formulated as:

$$\min_{\mathbf{U}, \mathbf{L}} \|\mathbf{C} - \mathbf{U}\mathbf{L}^\top\|_F^2 + \lambda_1 \|\mathbf{U}\|_F^2 + \lambda_2 \|\mathbf{L}\|_F^2,$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix, λ_1 and λ_2 are regularization parameters. We next introduce the MF-based models considered in the evaluation.

3.2.1.1 LRT

LRT [GTHL13] is a time-enhanced MF model. Based on the observation that user's check-in behavior varies with time, LRT models each user by different latent vectors for different time slots, and the final recommendation score is computed from all the latent vectors.

Temporal influence. To model users' preferences at different time, LRT factorizes a check-in matrix $\mathbf{C}^{(t)}$ for each time slot t separately, where $t \in \{0, 1, \dots, 23\}$ is an hour in a day. Furthermore, inspired by the intuition that users' interests in close time slots tend to be similar, a regularization term is added into the objective function of MF, formulated as $\sum_{t=1}^T \sum_{i=1}^m \psi_i(t, t-1) \|\mathbf{u}_i^{(t)} - \mathbf{u}_i^{(t-1)}\|_2^2$, where $\psi_i(t, t-1)$ is the similarity between $\mathbf{C}_i^{(t)}$ and $\mathbf{C}_i^{(t-1)}$. LRT sums up the recommendation scores of all time slots as the final score, i.e., $\hat{C}_{ij} = \sum_t \mathbf{u}_i^{(t)} \mathbf{l}_j$.

3.2.1.2 IRenMF

IRenMF [LWSM14] is based on Weighted Matrix Factorization (WMF) [HKV08, PZC⁺08]. The intuitions behind IRenMF are (i) user has similar preferences on neighboring POIs (location-level influence) [HSL14], and (ii) POIs in the same geographical region may share similar user preferences (region-level influence).

Geographical influence. For modeling location-level influence, the recommendation score of user i on POI j further includes the influence of the neighboring POIs $\mathcal{N}(l_j)$. Formally, the model defines the recommendation score $\hat{C}_{ij}$ as $\hat{C}_{ij} = \alpha \mathbf{u}_i \mathbf{l}_j^\top + (1 - \alpha) \frac{1}{Z(l_j)} \sum_{l_k \in \mathcal{N}(l_j)} \text{Sim}(l_j, l_k) \mathbf{u}_i \mathbf{l}_k^\top$, where $\alpha \in [0, 1]$ is used to control the influence of neighboring POIs, $\text{Sim}(l_j, l_k)$ is a distance-based weight between l_j and l_k , and $Z(l_j)$ is a normalization term.

For modeling region-level influence, IRenMF first clusters all POIs into G regions based on their geographical locations. By assuming the latent factors of POIs from the same region share the same sparsity pattern, they add a lasso penalty in the objective function as $\sum_{g=1}^G \sum_{k=1}^K \omega_g \|\mathbf{L}_{(g)}^k\|_2$, where $\mathbf{L}_{(g)}^k$ contains k^{th} latent factors of all POIs in region g and ω_g is a weight assigned to $\mathbf{L}_{(g)}^k$.

3.2.1.3 GeoMF

GeoMF [LZX⁺14] is a geographical WMF model. In order to capture the spatial clustering phenomenon (i.e., POIs visited by same users are likely to be in the same region [YYLL11]), GeoMF integrates geographical influence by modeling users' activity regions and the influence propagation on geographical space.

Geographical influence. GeoMF divides the whole geographical space into R grids, each of which represents a geographical region. For each POI, its influence is propagated to surrounding regions, attracting nearby users to visit. In particular, two matrices are introduced, namely user activity areas $\mathbf{X} \in \mathbb{R}^{M \times R}$ and POI influence areas $\mathbf{Y} \in \mathbb{R}^{N \times R}$. The entry $Y_{jr} = \frac{1}{\sigma} K(\frac{d(r,j)}{\sigma})$ denotes POI j 's influence on region r , where $d(r, j)$ is the distance between POI j and region r , $K(\cdot)$ is standard normal distribution and σ is the standard deviation. X_{ir} denotes the possibility of user i appearing in region r . User i 's geographical preference on POI j is estimated by $\mathbf{x}_i \mathbf{y}_j^\top$, and the final recommendation score is $\hat{C}_{ij} = \mathbf{u}_i \mathbf{l}_j^\top + \mathbf{x}_i \mathbf{y}_j^\top$.

3.2.1.4 RankGeoFM

RankGeoFM [LCL⁺15] is an ranking-based MF model that (i) learns users' preference rankings for POIs, and (ii) includes the geographical influence of neighboring POIs.

Geographical influence. RankGeoFM uses another latent matrix $\mathbf{U}^{(2)}$ to represent users' geographical preferences, in addition to user preference matrix $\mathbf{U}^{(1)}$. The recommendation score thus is computed as $\hat{C}_{ij} = \mathbf{u}_i^{(1)} \mathbf{l}_j^\top + \mathbf{u}_i^{(2)} \cdot \sum_{l_k \in \mathcal{N}(l_j)} w_{jk} \mathbf{l}_k^\top$. The first term

models the user preference score, while the second term models the geographical influence score that a user likes a POI because of its neighbors, where $\mathcal{N}(l_j)$ refers to the neighboring POIs of j and w_{jk} is the distance-based weight assigned to POI k .

3.2.1.5 ASMF

ASMF [LGRZ16] is a two-step POI recommendation framework that (i) learns potential locations from users' friends and (ii) incorporates potential locations into WMF to overcome cold-start problem.

Social influence. For each user i , ASMF considers the locations that are visited by three types of friends, i.e., social friends, location friends and neighboring friends, as his/her potential locations pot_i , and assigns a small value $\alpha \in [0, 1]$ to C_{ik} , where $k \in pot_i$.

Categorical influence. ASMF uses a category-based weight when computing the recommendation score, i.e., $\hat{C}_{ij} = (Q_{ic_j} + \epsilon)\mathbf{u}_i\mathbf{l}_j^\top$, where c_j is the category of POI j , Q_{ic_j} is the preference of user i on c_j and ϵ is a tuning parameter.

Geographical influence. A distance-based geographical score p_{ij}^G is fused with the result of WMF as the overall recommendation score, i.e., $\hat{C}_{ij} \propto p_{ij}^G \times \hat{C}_{ij}$, where p_{ij}^G is computed based on the distance distribution between users' home and their check-in POIs.

3.2.2 Poisson Factor Models

Poisson Factor Model (PFM) [MLKL11] is a probabilistic model that factorizes the user-POI check-in matrix $\mathbf{C}$ as $\mathbf{C} \sim \text{Poisson}(\mathbf{UL}^\top)$. We include two PFM-based models in our evaluation.

3.2.2.1 MGMPFM

MGMPFM [CYKL12] is a fusion model combining the outputs of PFM and a geographical modeling method, namely Multi-center Gaussian Model (MGM).

Geographical influence. Based on the observation that user's check-ins are usually distributed around several centers, such as home and workplace, MGM learns regions of activity for each user using multiple Gaussian distributions.

Recommendation. The recommendation score is defined as $P_{ij} = P(C_{ij}) \cdot P(l_j|R_i)$, where $P(C_{ij}) \propto \mathbf{u}_i\mathbf{l}_j^\top$ is the output of PFM, $P(l_j|R_i)$ is computed by MGM, and R_i is the regions of user i .

3.2.2.2 GeoPFM

The idea of GeoPFM [LXP⁺15] is that user’s geographical preference and interest preference are mutually affected, and user’s preference is related to both of them. Hence, GeoPFM jointly learns both geographical preference and interest preference for users.

Geographical influence. Latent regions are integrated into PFM, represented by two-dimensional Gaussian distributions on the spatial space, and each user has a multinomial distribution on regions.

3.2.3 Link-based Models

LFBCA [WTM13] is a link-based model that constructs a graph to model LBSN users and their relations. In the graph, both **user preference** and **social influence** are modeled by different types of edges. Particularly, users with similar check-in behaviors are linked to model the “similarity relations” and edges representing “friendship relations” are added to connect friends in the graph. Based on the constructed graph, the Bookmark-Coloring Algorithm algorithm (BCA) [Ber06] is executed for each user to compute his/her similarity to every other users, and then the User-based Collaborative Filtering (UCF) [AT05] is performed based on the similarities.

3.2.4 Hybrid Models

Hybrid model combines the outputs of two or more recommendation methods and each method models user preference or a type of context information. For example, MGMPFM (see Section 3.2.2.1) is a hybrid model of PFM and MGM. We introduce more hybrid models included in our evaluation next. Note that for recommendations, USG linearly combines the context influence, while other models use the multiplication of context influence as the final recommendation scores.

3.2.4.1 USG

USG [YYLL11] models user preference, social influence and geographical influence simultaneously for POI recommendations.

User preference. The model adopts UCF to model user preference, where the similarity between two users is computed based on their common check-ins.

Social influence. To exploit the social influence, USG proposes Friend-based Collaborative Filtering (FCF) to make POI recommendations based on similar friends. The similarity between friends is based on their common check-in POIs and common friends.

Geographical influence. Given a POI j and a user i , geographical influence is estimated as the probability of visiting j based on the user’s historical visited POIs $\mathcal{L}_i$, i.e., $P_{ij}^g = \prod_{l_k \in \mathcal{L}_i} Pr(d(l_j, l_k))$, where $d(l_j, l_k)$ is the distance between l_j and l_k . $Pr(\cdot)$ estimates the probability that users travel a distance of $d(l_j, l_k)$.

3.2.4.2 iGSLR

iGSLR [ZC13] exploits geographical preference and social influence for POI recommendations.

Social influence. Similar to USG, iGSLR also uses FCF to leverage friends' check-ins, where the similarity between friends is computed based on the distance of their residences. In our datasets, since residence locations of users are not available, we take users' most frequent check-in POIs as their residences.

Geographical influence. For each user, iGSLR learns a distance distribution from his/her check-in history using Kernel Density Estimation (KDE). The probability of user i visiting a new POI j is thus estimated based on the KDE values of the distances between POI j and the POIs visited by user i .

3.2.4.3 LORE

Different from other models, LORE [ZCL14] considers sequential influence, in addition to social and geographical influence.

Social influence. FCF is adopted to model social influence, where social similarities are defined as in iGSLR.

Geographical influence. For each user, LORE models a check-in probability distribution over a two-dimensional space using KDE. The geographical probability of visiting a new POI is then estimated based on its location on the check-in probability distribution.

Sequential influence. LORE employs additive Markov chain (AMC) [Usa09] to exploit sequential influence between POIs. The sequential probability of a user visiting a POI is based on the transition probability between all the user's visited POIs and the target POI.

3.2.4.4 GeoSoCa

GeoSoCa [ZC15a] models three types of context information, namely geographical, social and categorical correlations.

Geographical influence. GeoSoCa also uses two-dimensional KDE for geographical modeling. Different from LORE, where σ is shared by all users, GeoSoCa adds a local (i.e., user-dependent) bandwidth to make the geographical modeling more personalized.

Social influence. GeoSoCa estimates a power-law distribution (denoted as $f_S(x_{ij})$) of users' social check-in frequency. The social check-in frequency x_{ij} refers to the check-in frequency on POI j made by user i 's friends. GeoSoCa uses the cumulative distribution of $f_S(x_{ij})$ as the social influence in recommendations.

Table 3.1: Summary of the POI recommendation models in our evaluation.

Models	Methodology					Information-used					
	CF	MF	PFM	Link	Hybrid	User Pref.	Geo.	Soc.	Temp.	Seq.	Cat.
USG [YYLL11]	✓				✓	✓	✓	✓			
MGMPFM [CYKL12]			✓		✓	✓	✓				
LRT [GTHL13]		✓				✓			✓		
iGSLR [ZC13]	✓				✓		✓	✓			
LFBCA [WTM13]				✓		✓		✓			
LORE [ZCL14]	✓				✓		✓	✓		✓	
IRenMF [LWSM14]		✓				✓	✓				
GeoMF [LZX ⁺ 14]		✓				✓	✓				
RankGeoFM [LCL ⁺ 15]		✓				✓	✓				
GeoPFM [LXP ⁺ 15]			✓			✓	✓				
GeoSoCa [ZC15a]					✓		✓	✓			✓
ASMF [LGRZ16]		✓				✓	✓	✓			✓

Categorical influence. Similar to social influence modeling, GeoSoCa estimates a power-law distribution (denoted as $f_C(y_{ic})$) for users’ categorical check-in frequency. The categorical check-in frequency y_{ic} denotes the check-in frequency of user i on all the POIs with category c . The cumulative distribution of $f_C(y_{ic})$ is used as categorical influence in recommendations.

3.3 Evaluation Setting

In this section, we introduce the setting of our evaluation. More details about the evaluation, such as model implementations and parameter settings can be found in <http://spatialkeyword.sce.ntu.edu.sg/eval-vldb17/>.

3.3.1 Datasets

Our experiments are conducted on three public check-in datasets.

Gowalla dataset. The Gowalla check-in data¹ was generated worldwide from February 2009 to October 2010. We filter out those users with fewer than 15 check-in POIs and those POIs with fewer than 10 visitors. The filtered dataset comprises 18,737 users, 32,510 POIs, 1,278,274 check-ins. The sparsity of user-POI check-in matrix is 99.865%.

Foursquare dataset. The Foursquare data² [YZQ16] includes check-in data from April 2012 to September 2013. We use the records generated within United States (except Alaska and Hawaii) and eliminate those users with fewer than 10 check-in POIs, as well as those POIs with fewer than 10 visitors. The filtered dataset contains 24,941 users, 28,593 POIs and 1,196,248 check-ins. The sparsity of user-POI check-in matrix is 99.900%.

¹<http://snap.stanford.edu/data/loc-gowalla.html>

²<https://sites.google.com/site/yangdingqi/home/foursquare-dataset>

Yelp dataset. The Yelp data³ contains a large number of geo-tagged businesses (considered as POIs) and reviews within several cities. We eliminate those users with fewer than 10 check-in POIs, as well as those POIs with fewer than 10 visitors. This yields a dataset with 30,887 users, 18,995 POIs and 860,888 reviews. The sparsity of user-POI check-in matrix is 99.860%.

Notably, the selected three datasets have different geographical granularities (i.e., Gowalla: worldwide, Foursquare: across US and Yelp: within several cities) and different sparsity, and thus reflects different scenarios of POI recommendations. Therefore, the evaluation on these datasets would be more all-around than just using a single dataset.

We partition each dataset into training set, tuning set and test set. For each user, we use the earliest 70% check-ins as the training data, the most recent 20% check-ins as the test data and the remaining 10% as the tuning data. For each model, we tune the parameters based on the tuning data to find the optimal values that maximize Pre@10 and Rec@10 (see Section 3.3.2), and subsequently use them in the test data.

3.3.2 Evaluation Metrics

To evaluate the models, we use 4 widely-used metrics, i.e., precision (Pre@K), recall (Rec@K), normalized discounted cumulative gain (nDCG@K) and mean average precision (MAP@K). To our knowledge, none of the previous work uses all the 4 metrics for experiments.

Pre@K and Rec@K. Given the top-K returned POIs for user u , Pre@K and Rec@K are defined as: $Pre@K = \frac{tp_u}{tp_u + fp_u}$ and $Rec@K = \frac{tp_u}{tp_u + tn_u}$, respectively, where tp_u is the number of recommended POIs that are visited by u (i.e., correct recommendations); fp_u is the number of recommended POIs that are not visited by u (i.e., incorrect recommendations); tn_u is the number of POIs visited by u but not in the top-K recommendations. The average of precision (recall) values of all users is reported.

nDCG@K. For each user, nDCG@K is defined as: $nDCG@K = \frac{DCG@K}{IDCG@K}$, where $DCG@K = \sum_{i=1}^K \frac{2^{rel_i} - 1}{\log_2(i+1)}$. IDCG@K is the DCG@K value when the recommended POIs are ideally ranked, and rel_i refers to the graded relevance of the result ranked at the position i . nDCG@K is in the range 0 to 1 and higher value means better results. The average of nDCG values of all users is reported.

MAP@K. MAP@K is the arithmetic mean of top-K average precision (AP@K) over all users, which is computed by $MAP@K = \sum_{u=1}^M AP@K / M$, where M is the number of users. For each user, $AP@K$ is defined as $AP@K = \frac{1}{K} \sum_{i=1}^K Pre@i$, where $Pre@i$ refers to the precision for top- i recommended POIs.

³Yelp dataset challenge round 7 (access date: Feb 2016), https://www.yelp.com/dataset_challenge

3.3.3 Performance Evaluation Procedure

To systematically evaluate all the models, we devise an all-around evaluation procedure with the following four components.

3.3.3.1 Evaluation on Different Types of Data

To evaluate the effect of different data properties on accuracy, we design the following two experiments.

Different datasets. We evaluate all the models on Gowalla, Foursquare and Yelp data. We vary K from 5, 10, 20 to 50.

Data density. To investigate the effect of training data density, for each of Foursquare and Gowalla datasets, we generate training sets with different density levels, i.e. 0.0010, 0.0008, 0.0006, 0.0004 and 0.0002, by randomly eliminating non-zero entries of the check-in matrix (to make it sparser) or randomly moving data from tuning set to training set (to make it denser). Note that the density of Foursquare dataset is 0.0010, and thus we can only generate training data with density less than 0.0010.

3.3.3.2 Evaluation for Different Types of Users

We design three experiments to study the effect of different types of user properties on recommendation models.

Number of check-in POIs of users. The number of check-in POIs is expected to affect the accuracy of recommendation models. We divide users into groups based on the number of check-in POIs in training data. Particularly, we divide Gowalla users into five groups: “<15”, “15–30”, “30–50”, “50–100” and “>100”, which contain 6164, 7201, 2979, 1672 and 721 users, respectively. We also divide Foursquare users into five groups: “<10”, “10–20”, “20–30”, “30–50” and “>50”, which contain 6045, 9689, 4882, 3341 and 984 users, respectively. We train the models using all users and evaluate them on different groups of users separately.

Activity range of users. As geographical factor is important in POI recommendations, we design this experiment to study the effect of users’ activity range on recommendation models. In particular, we divide users of both Gowalla and Foursquare data into five groups: “<10”, “10–50”, “50–200”, “200–800” and “>800” based on the average distance between their check-in POIs (in kilometers), which reflects the activity range. In Gowalla, there are 3664, 3263, 3536, 4213 and 4061 users in the five groups, respectively. In Foursquare, there are 6814, 3189, 3419, 5939 and 5580 users in the five groups, respectively. Note that the numbers of check-in POIs in these groups are similar, and thus will not affect the results.

Recommendation for tourist users. To evaluate the models for a special type of users, namely tourists, we compare the models with a simple baseline on two additional datasets.

3.3.3.3 Evaluation for Different Modeling Methods

As discussed in Section 3.2.4, these hybrid POI recommendation models differ in how they model user preferences and each type of context information, particularly the geographical and social influence. It is very useful if we can evaluate the individual component of these methods to answer questions such as how good is the user preference modeling component for POI recommendations. We design the following experiments to compare the user preference, geographical and social components used in the models. The names of these individual modeling components are shown in Table 3.1.

Comparing geographical modeling methods. We evaluate 6 types of geographical modeling methods used in these models. Moreover, we also evaluate the accuracy for users with different numbers of check-in POIs. Note that it is difficult to isolate the geographical and social parts from those MF-based models and GeoPFM, and thus we do not include them here.

Comparing social modeling methods. We evaluate 5 types of social modeling methods used in these models. We also evaluate the accuracy for users with different numbers of friends.

Comparing user preference modeling methods. We evaluate 7 types of user preference modeling methods, without any context information (e.g., geographical and social information) included.

3.3.3.4 Scalability Evaluation

Scalability is also an important dimension for the practical interest of a recommender system. Therefore, we evaluate both the training and querying (i.e., recommendation) scalability of the models. To the best of our knowledge, this is the first work to evaluate both training and querying scalability of POI recommendation models.

Training scalability. To explore the training scalability, we use 20%, 40%, 60% and 80% of the Gowalla dataset as the training sets to test the training time of 7 models (i.e., MGMPFM, LRT, LFBCA, IrenMF, GeoMF, RankGeoFM and GeoPFM).

Querying scalability. To explore the querying scalability, we divide Gowalla users into five groups: “<15”, “15–30”, “30–50”, “50–100” and “>100” based on the number of users’ check-in POIs, and test the average querying time of the models for each group.

Time complexity analysis. We analyze the training and querying time complexity of different models.

Figure 3.2: Varying K on Gowalla.

Figure 3.3: Varying K on Foursquare.

3.4 Evaluation Results

In this section, we show the experimental results⁴. Note that for some experiments, results on nDCG and MAP are similar with precision and recall, and we do not show them for saving space.

3.4.1 Performance on Different Types of Data

3.4.1.1 Performance on Different Datasets

Figures 3.2, 3.3 and 3.4 depict the overall comparison of the 12 models with respect to top-K recommendations on Gowalla, Foursquare and Yelp, respectively. Note that the Foursquare data does not have social information and thus we only report results for those methods without utilizing social information. Additionally, GeoSoCa and ASMF are only evaluated on Yelp since Gowalla and Foursquare do not have the categorical information it needs.

⁴The related datasets and source code are available at: <http://spatialkeyword.sce.ntu.edu.sg/eval-vldb17/>

Figure 3.4: Varying K on Yelp.

Figure 3.5: Varying data sparsity.

Figure 3.6: Performance of POI recommendations for users with different numbers of check-in POIs

For each model, the accuracy are similar on Gowalla and Foursquare data. However, all the models perform worse on Yelp. For example, the Pre@5 of RankGeoFM on Gowalla and Foursquare are 0.069 and 0.063, respectively, while the value is only 0.032 on Yelp. This might be because that the activity range of the Yelp users are larger than the Gowalla and Foursquare users (see Figure 3.1(a)), which makes it difficult to model

Figure 3.7: Performance of POI recommendations for users with different average distance among check-in POIs.

users’ geographical preferences (which is discussed in Section 3.4.2.2). Considering the relative performances, most of the models perform consistently in the three datasets. We discuss their relative performances as follows.

Hybrid models. Among hybrid models, USG exhibits better recommendation quality than iGSLR, LORE and GeoSoCa. Take the results on Yelp as an example, in terms of Pre@5, USG outperforms the other three by 105.93%, 6.71% and 43.29%, respectively. One possible reason is that although iGSLR, LORE and GeoSoCa leverage geographical, social, sequential and categorical information to indirectly characterize user preference, they still miss direct modeling of user preference as USG does (i.e., using UCF). Therefore, for hybrid models, it would be better to utilize other information on top of user preference modeling for POI recommendations.

IRenMF, GeoMF and RankGeoFM. They are the top-3 best models, and RankGeoFM normally performs the best. Compared with USG, RankGeoFM and IRENMF achieve approximate 14% and 10% improvement, respectively, on Gowalla in terms of all the 4 metrics and different K values. GeoMF also outperforms USG by 5%-10% and 15%-20% on Gowalla and Yelp, respectively. All the three models are designed for implicit feedback data. This indicates that modeling user’s check-ins as implicit feedback is more appropriate in POI recommendations.

Comparing IRENMF and GeoMF. IRENMF is better than GeoMF on Gowalla and Foursquare data, but worse on Yelp data. This might be because that IRENMF assumes users tend to visit those POIs near their visited locations, while GeoMF considers those POIs as negative samples. Figure 3.1(a) shows that the distances between users’ check-in POIs in Gowalla and Foursquare are smaller than Yelp, which means users are more likely to visit nearby POIs in Gowalla and Foursquare. Thus, the assumption of IRENMF is more likely to hold on Gowalla and Foursquare than Yelp, which leads to their different performances on these datasets.

ASMF. ASMF performs similarly to USG (<5% difference). However, it is not as good as the other geographical-enhanced MF models. For example, GeoMF is 8%–23% better than ASMF on all the evaluation metrics and K values. This might be because ASMF focuses on utilizing social information, while learning geographical influence might be a better way to improve MF model.

LRT. LRT gives the worst performance among these models. It only considers temporal information, without modeling geographical and social influence. In addition, partitioning the check-in matrix based on time slots makes the data sparser, causing negative effects on learning user preferences.

LFBCA. Ye et al. [YYLL11] show that the performance of a link-based method is not as good as USG. However, in our evaluation, LFBCA, which is also a link-based method, achieves similar accuracy with USG on Gowalla, and it is 5%–10% worse than USG on Yelp. This is because LFBCA not only utilizes information of friends, but also considers the effects of other spatially similar users. This tells us that, for link-based models, users’ relations with respect to their spatial behaviors should also be considered as “links”.

Poisson Factor Models. In the experiments, GeoPFM consistently beats MGMPFM. The difference between them is that GeoPFM jointly learns geographical influence and user preference, while MGMPFM simply fuses the outputs of the two components. This implies that joint learning would be a better approach for leveraging context information than separately modeling.

3.4.1.2 Performance on Different Density

Figure 3.5 shows the accuracy on Gowalla and Foursquare, under different data densities. LRT and iGSLR are not included here since their performances are not good in the previous experiments. The results on Yelp are qualitatively similar and omitted.

GeoPFM. In general, except for GeoPFM, every model is significantly jeopardized by lower data density. For example, the performances of these models in terms of Pre@10 decrease at least 35% on Gowalla, when the data density drops from 0.0008 to 0.0002. Although GeoPFM is not as good as RankGeoFM, IRenMF and GeoMF when data is dense, it is very robust to low data density. In terms of Pre@10, GeoPFM only experiences 13.66% and 12.25% loss when the density decreases from 0.0008 to 0.0002. One possible explanation is that GeoPFM uses a hierarchical way to profile user preferences, i.e., each user has preferences on latent regions and preferences on the POIs within each region. This might be helpful in overcoming data scarcity problem.

RankGeoFM. RankGeoFM reports the best performance when the density is greater than 0.0008, outplaying the second best one (i.e., IRenMF) by 5%-10%. However, when the density downs to 0.0002, its accuracy declines dramatically, which is worse than GeoMF and GeoPFM. It might be because that RankGeoFM learns to rank the positive

Figure 3.8: Varying K on Foursquare Singapore.

examples higher than negative examples, but there are fewer positive examples available to learn the rankings in sparse data. As a consequence, the result is less reliable.

GeoMF. The performance of GeoMF is slightly worse than IRenMF and RankGeoFM at 0.0008 and 0.0010, but turns to be the most effective model when the density is 0.0002, outperforming the second best one by 5%–10% on both datasets. Hence, GeoMF is preferable than the other models for sparse data.

Comparing USG and LFBCA. The results show that USG performs better than LFBCA on sparse data (0.0002 – 0.0006) by at least 7%, while slightly worse on dense data (0.0010) by 2%–3%. This might be because that for user preference modeling, LFBCA also considers indirect similar users, e.g., similar users of the similar users. The indirect users might provide more information when data is dense, while cause bias when data is sparse.

LORE. LORE does not directly model user preference, but it still benefits from higher density. This indicates that context information such as social influence, is also sensitive to data density.

3.4.1.3 Performance on Other Datasets

In addition, we also conduct extensive experiments on some other datasets. The details of the datasets are shown in Table 3.2.

Performances on different cities. Figures 3.8 and 3.9 show the results on two city-level data, Singapore and Tokyo. The details of the datasets are given in Table 3.2. From the figures, we can see that the relative performances of the models are qualitatively

⁵<http://hpc.isti.cnr.it/~nardini/datasets/LearNext.tar.gz>

⁶<https://sites.google.com/site/yangdingqi/home/foursquare-dataset>

Figure 3.9: Varying K on Foursquare Tokyo.

Table 3.2: Data descriptions

Name	# user	# POI	# check-in	Sparsity
Tourist datasets				
Foursquare (tourist)	11,555	28,295	408,958	99.875%
Florence ⁵	3,011	927	30178	98.919%
City datasets				
Singapore [LCL ⁺ 15]	2,321	5,596	95,118	99.268%
Tokyo ⁶	2,287	7,055	389,029	99.203%
Other different datasets				
Gowalla (dense)	2,426	2,195	221,487	97.527%
Gowalla (large)	41,297	67,738	2,062,996	99.952%

similar to the results shown in Figures 3.2, 3.3 and 3.4. RankGeoFM, IRenMF and GeoMF are better than the other models, and RankGeoFM usually performs the best. This indicates that RankGeoFM, IRenMF and GeoMF are better for global-level data (Gowalla and Yelp), country-level data (Foursquare) and city-level data (Singapore and Tokyo).

The results also show that the precision and recall values on the city-level datasets are higher than the global-level and country-level datasets. For example, the Pre@5 value of RankGeoFM is 0.11 on Singapore and Tokyo, while are 0.07, 0.06 and 0.03 on Gowalla, Foursquare and Yelp, respectively. There are two possible explanations: (i) the data density of Singapore and Tokyo data are higher than the Gowalla, Foursquare and Yelp data, and (ii) the activity ranges of the users in the city-level datasets are smaller than those in the global-level and country-level datasets.

Figure 3.10: Varying K on Gowalla (dense).

Performance on denser and larger datasets. Figure 3.10 shows the results on a denser dataset, namely Gowalla (dense), whose sparsity is 97.527%, which is significantly smaller than the density of all the other datasets in our study. More details of the dataset can be found in Table 3.2. The relative performances of the models on the Gowalla (dense) are similar to on the Gowalla data, except that all the models achieve higher precision and recall values, which could largely be attributed to the higher density of the data.

Figure 3.11 shows the results on a larger scale dataset, namely Gowalla (large), which has over 40,000 users and 60,000 POIs. The details of the dataset are given in Table 3.2. GeoMF and RankGeoFM run out of memory on this data, and thus their results are not available. The results are qualitatively similar to the results on the Gowalla.

3.4.2 Performance on Different Users

3.4.2.1 Users with Different Numbers of Check-in POIs

Figure 3.6 shows the results for different groups of users with different numbers of check-in POIs. The models considering social information are only evaluated on Gowalla, and results on Yelp are similar and are omitted. We make the following observations: (1) USG, LFBCA, IrenMF, GeoMF and RankGeoFM consistently outperform the other models in general, among which IrenMF, GeoMF and RankGeoFM are usually better than the other two by 5%–15% on both datasets. The improvement is even larger for the groups of users with fewer check-in POIs. This observation confirms the superiority of the 5 models for both active and cold-start users. (2) LORE experiences the largest increase as the number of check-in POIs goes up as shown in Figure 3.6(a). In particular,

Figure 3.11: Varying K on Gowalla (large).

for the group of users “<15”, the Pre@10 of LORE is only half of USG, while it becomes comparable to the Pre@10 of USG for the group of users “100<”. It is worth noting that, the increase of other models using geographical, social and temporal information is not as significant as LORE. Thus, LORE’s improvement on active users could largely be attributed to its sequential influence modeling.

3.4.2.2 Users with Different Activity Ranges

Figure 3.7 shows the results for groups of users with different activity ranges on Gowalla and Foursquare. The results on Yelp are qualitatively similar and are omitted. We make two observations: (1) RankGeoFM, GeoMF, IRenMF, USG and LFBCA outperform the others by a large margin (>30%) on all bases, and RankGeoFM usually performs the best. (2) All the models with geographical modeling experience at least 10% loss from the first group (<10) to the last group (800<) on both datasets in terms of Rec@10. This means larger range of user’s activity causes side effects for geographical-enhanced POI recommendations. Geographical modelings may not work well for users with wide activity ranges.

3.4.2.3 Tourist Users

The POI recommendation models are designed to uncover users’ interests and preferences, based on which recommendations are made. However, the behaviors of tourists are irregular, which might not follow closely with their interests and preferences. For example, tourists usually tend to visit popular places in a new city. To verify this, we have conducted experiments on the original Foursquare dataset (containing both tourists’ and natives’ check-ins), a tourist Foursquare dataset (the training and test data are constructed

Figure 3.12: Varying K on Foursquare, Foursquare (tourist) and Florence.

based on a previous study [WYC⁺15]) and a tourist dataset named Florence [MNSB15]. In addition to the models that are included in the evaluation paper, we also add a MostFreq baseline in the comparison, which always recommends the top-frequent POIs to a user.

Figure 3.12 shows that the MostFreq baseline performs poorly on original Foursquare data but comparable to the best models on Foursquare (tourist) data and Florence data. Particularly, on original Foursquare data, RankGeoFM is better than MostFreq by 150% in terms of Pre@5. However, on Foursquare (tourist) and Florence, RankGeoFM is better than MostFreq only by 30% and 13%, respectively. Moreover, MostFreq outperforms all the models except RankGeoFM when K is 20 or 50 on the tourist datasets. This shows that conventional POI recommendation models cannot properly capture the preferences of tourists. This is because conventional POI recommendation models are not specialized for tourists, whose behaviors are more irregular than the majority users of traditional POI datasets, i.e., native users. Therefore, conventional POI recommendation models are not much better than simple baselines such as MostFreq.

3.4.3 Different Modeling Methods

3.4.3.1 Geographical Modeling

Figures 3.13 and 3.14 show the evaluation results of 6 geographical modeling methods on Gowalla data. Similar results can be found on Foursquare and Yelp, and thus are omitted.

PD & PD+. PD is the geographical component of USG (see Section 3.2.4.1) and PD+ is an improved version of PD [YCM⁺13a]. As shown in Figures 3.13 and 3.14, PD+ outperforms all the other models by at least 25% with respect to different K values and user groups, and PD is the second best method in most cases.

Figure 3.13: Performance of geographical modeling methods.

MGM. MGM is the geographical component of MGMPFM (see Section 3.2.2.1). Figure 3.13 shows that the overall performance of MGM is not satisfactory, while Figure 3.14 shows that it performs well for users with more than 100 check-in POIs, better than PD by 10% on Pre@10 and only 1% worse on Rec@10. It indicates that when a user has many check-in POIs, the distribution of his/her POIs is more likely to be consistent with the assumption of MGM, i.e., following multi-centered Gaussian distribution.

1dKDE. 1dKDE is the geographical component of iGSLR (see Section 3.2.4.2). It performs the worst in our experiments.

2dKDE & AKDE. 2dKDE and AKDE are the geographical component of LORE and GeoSoCa (see Section 3.2.4.3 and Section 3.2.4.4), respectively. AKDE consistently outperforms 2dKDE by 15%–40%. This is because AKDE uses both global and personalized bandwidth of the kernel function for each user, and thus can perform better. Compared to PD, AKDE is 8%–17% worse for the first four user groups (<100), while becoming similar (<2% difference) for the last user group (>100). This means that, similar to MGM, AKDE is also better for active users.

3.4.3.2 Social Modeling

Figures 3.15 and 3.16 show the results of 5 social modeling methods:

FBCA. FBCA is the social component of LFBCA (see Section 3.2.3). In general, FBCA is the best social modeling method, outperforming the second best one, i.e., CI-FCF, by 6%–12% in terms of recall on Gowalla, while their precision values are similar. This indicates that link-based method might be a good choice for modeling social influence. Specifically, Figure 3.16 shows that FBCA is more powerful for cold-start users but not effective for active users. This is because for users with few friends, FBCA can utilize

Figure 3.14: Performance of geographical modeling methods for users with different numbers of check-in POIs.

his/her friends' friends and so on, which is helpful for aggregating more information for cold-start users, while the other models only consider the direct friends of users. However, for active users, FBCA might be inaccurate due to including the effects of too many indirect friends.

CI-FCF & CN-FCF. CI-FCF and CN-FCF are two social components of USG (see Section 3.2.4.1). From Figure 3.15, we can see that CI-FCF is more effective than CN-FCF by 22.77% in terms of Pre@5. It means that spatial similarity, which is based on check-in POI and frequency, is better than online similarity such as the number of common friends, for FCF method. This result is consistent with the previous work [YYLL11].

D-FCF. D-FCF is the social component of iGSLR and LORE (see Sections 3.2.4.2 and 3.2.4.3). Figure 3.15 shows that it is the worst among all methods.

SC. SC is the social component of GeoSoCa (see Section 3.2.4.4). In terms of Pre@10, SC outperforms CN-FCF and D-FCF by 15.23% and 20.87%, while it is outperformed by CI-FCF, which also aggregates social check-ins for recommendations. This means CI-FCF might be a better choice for social check-in aggregation.

It is worth noting that, in Figure 3.16, the recall values of all models increase in the beginning, but slightly decrease afterwards. This is because users with more friends also tend to check-in at more POIs. Specifically, users in the first group (<2) has 21.57 check-in POIs on average, while this value of the last group (20<) is 38.80.

3.4.3.3 User Preference Modeling

We compare 7 user preference modeling methods employed in POI recommendation models, including User-based Collaborative Filtering (UCF), Location-based Bookmark-coloring Algorithm (LBCA) Matrix Factorization (MF), Poisson Factor Model (PFM),

Figure 3.15: Performance of social modeling methods.

Weighted Matrix Factorization (WMF), Non-negative Matrix Factorization (NMF) [LFYX13] and Bayesian Personalized Ranking (BPRMF) [RFGST09]. To focus on user preference modeling only, all of these methods are based on user-POI check-in matrix, without utilizing any context information. Figures 3.17 and 3.18 depict the accuracy of these methods.

LBCA. LBCA is the link-based model in Section 3.2.3 without social links in the user-user graph. LBCA is the best method for user preference modeling. For example, it outperforms UCF, WMF and BPRMF by 11%, 7% and 13%, respectively, in terms of Pre@5. One possible reason is that it utilizes both direct and indirect similar users (i.e., similar users of similar users), and thus can aggregate more information for recommendations.

MF. Matrix Factorization, which performs well in traditional recommendation problems, fails for POI recommendations. This is because the data in POI recommendations is much sparser than in traditional recommendation problems, and MF does not fit well with check-ins, which is implicit feedback data. The finding is consistent with the result reported in previous work [LCL⁺15].

PFM & NMF. They do not perform well for POI recommendations. This also explains why MGMPFM and GeoPFM, which are based on PFM, are not good in our previous experiments.

UCF, WMF & BPRMF. They perform similarly (<3% difference) and are second to the best methods for modeling user preference. For example, they outperform NMF and PFM by over 140% on Pre@5. Moreover, they are better for both cold-start users and active users. This also explains why the models based on these methods, e.g., USG, IrenMF, GeoMF and RankGeoFM, perform good.

Figure 3.16: Performance of social modeling methods for users with different numbers of friends.

Table 3.3: Training time (hour) of the models

Train size	MGMFM	LRT	LFBCA	IRenMF	GeoMF	RankGeoFM	GeoPFM
20%	0.03	0.27	5.23	0.13	7.34	2.84	0.10
40%	0.06	0.32	8.25	0.21	7.38	4.00	0.21
60%	0.09	0.36	12.82	0.28	7.48	5.22	0.32
80%	0.12	0.41	19.20	0.35	7.45	6.29	0.44

3.4.4 Scalability

3.4.4.1 Training Scalability

Table 3.3 shows the training time of the POI recommendation models on training sets of different sizes. Note that we do not evaluate the training scalability of USG, iGSLR, LORE and GeoSoCa, since they do not need training. For ASMF, the training time on the Gowalla data (without category information) is the same as training WMF and thus is omitted. For LFBCA, we report the training time of the Bookmark-Coloring

Table 3.4: Average querying time (second) per user of the models for users with different numbers of check-in POIs

# of POIs	USG	MGM-PFM	LRT	iGSLR	LF-BCA	LORE	IRen-MF	Geo-MF	Rank-GeoFM	Geo-PFM	Geo-So	AS-MF
<15	3.95	3.40	1.45	1.96	2.05	7.15	0.80	0.28	9.65	0.25	4.98	0.19
15-30	4.79	4.57	1.45	3.53	2.05	10.66	0.80	0.28	9.64	0.25	7.82	0.19
30-50	6.38	6.06	1.45	11.04	2.04	17.63	0.80	0.28	9.61	0.25	13.27	0.19
50-100	9.2	6.48	1.45	56.61	2.04	30.28	0.80	0.28	9.63	0.25	22.79	0.19
100<	18.20	5.08	1.44	1292.57	2.04	72.54	0.79	0.28	9.58	0.25	42.83	0.19

Figure 3.17: Performance of user preference modeling methods.

Algorithm (BCA). To compare the scalability of each model with different training data sizes, we show the relative training time in Figure 3.19(a), which is defined as the ratio comparing to the running time using 20% training data.

We observe that (1) GeoMF is almost not affected by the size of the training set, with a variation of less than 2%. This is because the training time of GeoMF is based on $\|\mathbf{C} - \mathbf{X}\mathbf{Y}^\top\|_0$ instead of N_C (Table 3.6, Section 3.4.4.3 [LPCY17]). The value of $\|\mathbf{C} - \mathbf{X}\mathbf{Y}^\top\|_0$ is not influenced by the size of training data. Therefore, GeoMF is more scalable than the other models. (2) MGMPFM, LRT, IRenMF, RankGeoFM and GeoPFM scale linearly with the size of training data. Among these models, LRT is more scalable than the others. When the size of the training data increases by 300% (from 20% to 80%), the training time of LRT only increases by 50%, while IRenMF and RankGeoFM increase by over 170% and 120%, respectively. MGMPFM and GeoPFM perform similarly and worse than LRT, IRenMF and RankGeoFM. (3) The training time of LFBFA increases super-linearly with the size of training data. Therefore, LFBFA is not preferable for very large datasets.

3.4.4.2 Querying Scalability

Table 3.4 shows the average querying time per user of all the models for different user groups, each with different numbers of check-in POIs. In the experiments, querying time represents *the average time of computing the recommendation scores of a user u on all the POIs*. Ranking the POIs is not included in the querying time. For GeoSoCa, since categorical information is not available in the Gowalla data, we test the querying time of its geographical and social components (namely GeoSo) instead. To compare the scalability of different methods with the number of check-in POIs, we show the relative

Figure 3.18: Performance of user preference modeling methods for users with different numbers of check-in POIs.

querying time in Figure 3.19(b), which is defined as the ratio comparing to the time cost for the first user group (i.e., <15).

iGSLR. iGSLR has the worst scalability for recommendations. Particularly, it takes only 1.958s for users with fewer than 15 POIs, but 1292.566s for users with more than 100 POIs. This is because the querying time of iGSLR cubically increases with the number of the user’s check-in POIs (Table 3.6, Section 3.4.4.3 [LPCY17]). Therefore, iGSLR is not suitable for active users.

Hybrid models vs. Joint-learning models. On the one hand, the querying time of hybrid models usually increases with the number of check-in POIs of a user (except MGMPFM). For example, the querying time of GeoSo increases by 10 times, from 4.977s (for <15) to 42.828s (for 100<). The reason is that the hybrid models separately model geographical information based on users’ visited POIs. At recommendation time, the geographical component needs to iterate all the user’s visited POIs, and thus the querying time is usually polynomial to the number of visited POIs. On the other hand, the querying time of all the joint-learning models, e.g., IRenMF and GeoMF, is constant for different user groups. This is because these models jointly embed users’ preferences and contextual information into fixed-size matrices, and thus the querying time is the same. Therefore, the querying scalability of joint-learning models is usually better than hybrid models.

MGMPFM. Its querying time varies between 3.4s and 6.5s for all 5 user groups, which is more scalable than the other hybrid models. This is because the geographical modeling method of MGMPFM (i.e., MGM) is based on the Gaussian centers $\mathcal{G}_u$ of the user instead of the visited POIs $\mathcal{L}_u$. Even for active users, their check-in POIs are likely to locate around a few number of centers. Hence, MGMPFM can scale better to active users.

Figure 3.19: Relative training and querying time on Gowalla.

3.4.4.3 Time Complexity Analysis

USG. USG recommends by summing up the preferences of all the similar users of u on all the POIs. Since u can have at most M similar users, the querying time complexity is $O(MN)$.

MGMPFM. MGMPFM has two individual components, namely MGM and PFM. The training process of MGMPFM is to train PFM, of which the complexity of $O(rN_C K)$ [MLKL11]. When making recommendations for user u , the querying complexity PFM is $O(KN)$, and MGM needs to compute the geographical score of each candidate POI, which is based on the Gaussian centers of u , and thus the querying complexity is $O(|\mathcal{L}_u|N)$. The overall querying complexity is $O((|\mathcal{L}_u| + K)N)$.

LRT. Gao et al. claim that the training complexity of LRT is $O(rMNK)$ [GTHL13]. However, due to the sparsity of the check-in matrix, it can be rewrite as $O(rN_C K)$. The recommendation process is to sum up all the recommendation scores in different time slots, where the number of time slots is usually a small constant. Therefore, the querying complexity is $O(KN)$.

iGSLR. iGSLR consists of a geographical component (namely 1dKDE) and a social component (namely D-FCF). The querying process of 1dKDE is to compare the distances of the candidate POIs to u 's visited POIs (totally $|\mathcal{L}_u|N$ values) and the distances between every pair of u 's visited POIs (totally $|\mathcal{L}_u|^2$ values). Hence, the querying complexity of 1dKDE is $O(|\mathcal{L}_u|^3 N)$. D-FCF needs to iterate all the friends of u to aggregate their preferences on N POIs, and thus the complexity is $O(|\mathcal{F}_u|N)$. The overall querying complexity of iGSLR is $O((|\mathcal{L}_u|^3 + |\mathcal{F}_u|)N)$.

LFBCA. In each training iteration of BCA, the PPR value of a user is transferred to other connected user. Due to the fact that a user can have at most M connected users, the

overall training time complexity of LFBCA is $O(rM^2)$. After obtaining the PPR values of all users, LFBCA recommends POIs to users in a same way with UCF. Therefore, the querying complexity is $O(MN)$.

LORE. LORE consists of a geographical component (namely 2dKDE), a social component (namely D-FCF) and a sequential component (namely AMC). The querying complexity of D-FCF is the same to in iGSLR, i.e., $O(|\mathcal{F}_u|N)$. 2dKDE takes u 's visited POIs as reference points for each candidate POI to compute kernel function values, and thus its complexity is $O(|\mathcal{L}_u|N)$. AMC computes the transition probability of each visited POI to each candidate POI, whose complexity is $O(|\mathcal{L}_u|N)$. Therefore, the overall querying complexity is $O((|\mathcal{L}_u| + |\mathcal{F}_u|)N)$.

IRenMF. In each training iteration, IRenMF first fixes $\mathbf{L}$ and uses the alternating least squares (ALS) algorithm to update $\mathbf{U}$, of which the time complexity is $O(N_c K^2)$ [HKV08, PZC⁺08]. Subsequently fixing $\mathbf{U}$, IRenMF uses the accelerated proximal gradient (APG) method [TY10] to update $\mathbf{L}$. Note that solving the APG optimization problem is also an iterative process and within each iteration, IRenMF performs line search to find an optimal learning rate. The complexity of updating $\mathbf{L}$ is $O(r_L r_{ls}(NK + N_C))$, where r_L is the number of iterations of APG optimization and r_{ls} is the number of iterations for line search. Therefore, the overall training complexity is $O(r(N_C K^2 + r_L r_{ls}(NK + N_C)))$. When making recommendations, IRenMF includes the user's preference on k nearest POIs. The querying complexity is $O(KkN)$.

GeoMF. The main computation of GeoMF is to iteratively update $\mathbf{U}$, $\mathbf{L}$. In each iteration, the complexity of updating $\mathbf{U}$ and $\mathbf{L}$ is $O(\|\mathbf{C} - \mathbf{X}\mathbf{Y}^\top\|_0 K^2)$, where $\|\cdot\|_0$ is l_0 norm that used to count the non-zero values in a matrix [LZX⁺14]. When making recommendations, the complexity of computing preference scores and geographical scores are $O(KN)$ and $O(RN)$, respectively. Thus, the querying complexity is $O((K + R)N)$.

RankGeoFM. The main computation of RankGeoFM is to make pairwise comparison of POIs to find incompatibility in the ranking-based learning process. For each positive example from training data, RankGeoFM randomly chooses a POI to construct a negative example and compute the recommendation scores of both examples for comparison. Based on Section 3.2.1.4, the complexity of computing a recommendation score is $O(Kk)$. Hence, the complexity of making a pairwise comparison till finding incompatibility is $O(Kk\bar{s})$, where $\bar{s}$ is the average number of sampling trials. On the other hand, the complexity of updating the latent matrices is $O(Kk)$. Thus, the overall training complexity is $O(rN_C Kk\bar{s})$. In the querying step, RankGeoFM is similar to IRenMF, considering the user's preference on k nearest POIs. The querying complexity is $O(KkN)$.

GeoPFM. GeoPFM uses a Expectation Maximization (EM) algorithm to estimate the parameters. In E-step, GeoPFM updates the probability distribution of each POI assigned to regions, whose time complexity is $O(N_C KR)$; In M-step, GeoPFM first recomputes the Gaussian distributions of the regions, whose complexity is $O(N)$, and subsequently updates $\mathbf{U}$ and $\mathbf{L}$ iteratively (for r_M iterations), whose complexity is $O(r_M N_C K)$.

Table 3.5: List of notations.

M	the number of users
N	the number of POIs
K	the number of dimensions of the latent space
$\mathbf{C}$	the user-POI check-in matrix
$\mathbf{X}$	the user activity areas matrix (GeoMF)
$\mathbf{Y}$	the POI influence areas matrix (GeoMF)
N_C	the number of non-zero entries in user-POI check-in matrix
r	the number of iterations for learning a model
r_M	the number of iterations for updating $\mathbf{U}$ and $\mathbf{L}$ in the M-step (Geo-PFM)
r_L	the number of iterations for APG optimization (IRenMF)
r_{ls}	the number of iterations for line search (IRenMF)
$ \mathcal{L}_u $	the number of POIs visited by user u
$ \mathcal{F}_u $	the number of friends of user u
$ \mathcal{G}_u $	the number of centers of user u (MGMPFM)
$ \mathcal{C}_u $	the number of categories visited by user u (GeoSoCa)
k	the number of nearby POIs (IRenMF & RankGeoFM)
R	the number of regions (GeoPFM & GeoMF)
$\bar{s}$	the number of trials to find an incompatible example (RankGeoFM)
$\ \cdot\ _0$	l_0 norm

Therefore, the overall training complexity of GeoPFM is $O(rN_C K(r_M + R))$. The recommendation score of GeoPFM is a geographical-weighted recommendation score of PFM (see Section 3.2.2.2). The complexity of computing the geographical weight is $O(1)$. Therefore, the querying complexity is $O(KN)$.

GeoSoCa. LORE consists of a geographical component (namely AKDE), a social component (namely SC) and a categorical component (namely CC). The querying complexity of AKDE is same to the 2KDE of LORE, i.e., $O(|\mathcal{L}_u|N)$. SC compute the social check-in frequency of u by aggregating all the check-ins from his/her friends, and thus the complexity is $O(|\mathcal{F}_u|N)$. By using similar method, the complexity of CC to aggregate categorical check-ins is $O(|\mathcal{C}_u|N)$. Therefore, the overall querying complexity is $O((|\mathcal{C}_u| + |\mathcal{F}_u| + |\mathcal{L}_u|)N)$.

ASMF. The time complexity of training ASMF is $O(N_C K)$ [LGRZ16]. The main computation of ASMF in the query step is the same as MF, i.e., multiplying user and POI latent vectors. Therefore, the time complexity of recommendation is $O(KN)$.

Table 3.5 shows the definitions of the notations used in the complexity analysis. For the notations that are used in specific models (e.g., the number of nearby POIs used in IRenMF and RankGeoFM, denoted as k), we indicate the model names in the brackets.

Table 3.6: Time complexity of the POI recommendation models.

Models	Training	Querying
USG	N/A	$O(MN)$
MGMPFM	$O(rN_CK)$	$O((\mathcal{G}_u + K)N)$
LRT	$O(rN_CK)$	$O(KN)$
iGSLR	N/A	$O((\mathcal{L}_u ^3 + \mathcal{F}_u)N)$
LFBCA	$O(rM^2)$	$O(MN)$
LORE	N/A	$O((\mathcal{L}_u + \mathcal{F}_u)N)$
IRenMF	$O(r(N_CK^2 + r_L r_{ls}(NK + N_C)))$	$O(KkN)$
GeoMF	$O(r\ \mathbf{C} - \mathbf{XY}^\top\ _0 K^2)$	$O((K + R)N)$
RankGeoFM	$O(rN_CKk\bar{s})$	$O(KkN)$
GeoPFM	$O(rN_CK(r_M + R))$	$O(KN)$
GeoSoCa	N/A	$O((\mathcal{C}_u + \mathcal{F}_u + \mathcal{L}_u)N)$
ASMF	$O(N_CK^2)$	$O(KN)$

Table 3.6 summarizes the training and querying time complexity of the POI recommendation models. For the training time complexity, we can conclude that: (1) The complexity of LFBCA is significantly higher than the other models; (2) The complexity of MGMPFM, LRT, IRenMF, RankGeoFM and GeoPFM are linear to the number of non-zero values in the check-in matrix (denoted as N_C), i.e., the size of training data. For the querying time complexity, we can conclude that: (1) The complexity of all the models are linear to the number of POIs in the dataset. Because the models need to compute the recommendation score of the given user u on each POI. (2) The complexity of the MF/PFM-based models are linear to the number of dimensions of the latent space (denoted as K). (3) The complexity of the hybrid models are related to the contextual information and the number of context-related observations. For example, LORE models geographical and social information, and thus its querying complexity is related to the number of check-in POIs and the number of friends, i.e., $O((|\mathcal{L}_u| + |\mathcal{F}_u|)N)$; GeoSoCa models geographical, social and categorical information and its querying complexity is $O((|\mathcal{C}_u| + |\mathcal{F}_u| + |\mathcal{L}_u|)N)$. On the contrary, the querying complexity of joint-learning models (i.e., those jointly learn contextual information, such as GeoMF) are independent to the number of context-related observations. Because they already embed contextual information into user preferences in the training steps and do not need to iterate historical data for querying.

3.4.5 Summary of New Insights

From the evaluation, we observe many interesting findings that have never been reported in any existing work. Those findings are important for understanding POI recommen-

dation models, which helps us to choose and design a suitable model for a particular scenario (e.g., sparse data). We summarize the key findings below.

- RankGeoFM, IRenMF and GeoMF outperform the other models, on different datasets and types of users, and RankGeoFM usually performs the best. Those models (1) are based on implicit feedback models, such as WMF and ranking-based MF, and (2) consider geographical information. In contrast, LRT, which does not possess these two properties, performs worse. Moreover, models without directly modeling user preference based on user-POI check-in matrix (e.g., iGSLR) are not attractive (Figures 3.2, 3.3, 3.4, 3.6 and 3.7).
- RankGeoFM is the best model when the check-in data is dense (i.e., the sparsity is larger than 0.0004), and is followed by IRenMF and GeoMF. However, GeoMF is better than all other methods for sparse data (i.e., the sparsity is as low as 0.0002) (Figure 3.5).
- GeoPFM is the least sensitive model to the sparsity change, while the accuracy of all the other models decreases dramatically when the data becomes sparser (Figure 3.5).
- LORE experiences the largest increase when users have more POIs, which is mainly attributed to its sequential modeling (Figure 3.6).
- The accuracy of all geographical models decrease for larger user activity ranges (Figure 3.7).
- PD+ is the best geographical modeling method for all the user groups, and PD usually performs the second best. This indicates that utilizing power-law distribution is an effective solution for geographical modeling (Figures 3.13 and 3.14).
- Methods that model personalized two-dimensional check-in distributions (i.e., AKDE and MGM) outperform PD, only for users with more than 100 check-in POIs. Because personalized two-dimensional distribution requires more data to precisely capture users' behaviors. Thus, it is preferable for active users (Figure 3.14).
- FBCA and CI-FCF are the two best models for social modeling. FBCA performs the best for users with 1 friend, while CI-FCF is better for users with more than 5 friends. FCF methods based on common friends (i.e., CN-FCF) and geographical distance (i.e., D-FCF) do not show promising performances (Figures 3.15 and 3.16).
- LBCA performs the best (by over 7%) for user preference modeling, followed by UCF, WMF and BPRMF, among which the difference is within 6%. These 4 models are better than the other methods. This also explains why the models based on these preference modeling methods (e.g., RankGeoFM and GeoMF) outperform the other models. Hence, to develop new POI recommendation models in the future, it would be more promising to extend LBCA, UCF, WMF or BPRMF for recommendations (Figures 3.17 and 3.18).

- For training scalability, GeoMF is the most scalable model, whose training time is constant to the size of training data; LFBCA scales super-linearly to the size of training data; The other models scale linearly to the size of training data (Figure 3.19(a)).
- For querying scalability, joint-learning models (i.e., LRT, IReMF, GeoMF, RankGeoFM and GeoPFM) are usually more scalable than hybrid models (e.g., iGSLR and GeoSoCa), with respect to the number of check-in POIs of the user (Figure 3.19(b)).

Chapter 4

Anomalous Trajectory Detection

With the availability of large-scale trajectory data, detecting anomalous trajectories has become a critical concern in many real-world scenarios. For example, it can help us to reveal malicious mobility behaviors (i.e., taxi driving frauds) or unusual traffic conditions (i.e., road construction). In recent years, various methods have been proposed for anomalous trajectory detection [ZLZ⁺11, CZC⁺13, LHL08, WSZ17, ZJL⁺15, ZJL⁺17, LXZ⁺17]. However, most of them define anomalous trajectories by leveraging hand-crafted distance or density metrics, which 1) cannot handle the complexity and variety of real-world trajectory data and 2) cannot support efficient online detection.

To this end, we propose a novel solution to overcome these limitations. In particular, our solution is equipped with 1) an end-to-end *deep generative model*, namely Gaussian Mixture Variational Sequence AutoEncoder (GM-VSAE), that can discover normal routes as latent representations via exploiting massive trajectory data, and 2) a novel *detection-via-generation* framework that can support efficient online detection of anomalous trajectories.

In GM-VSAE, we propose an inference network and a generative network that employ recurrent neural networks (RNNs) to encode the complex sequential information underlying the trajectories, and represent the route of each trajectory as a vector in a continuous latent space. Meanwhile, a Gaussian mixture model is jointly learned to model the probability distribution of routes represented in the latent space. The distribution characterizes the probability of *being normal* of all the routes of trajectories. We also reveal that different Gaussian components can be interpreted as different types of routes, such as streets in downtown areas and highways that connect distant locations. As a result, in the latent space, routes that are close to the center of a Gaussian component are more likely to be normal routes, because they have higher probability to be traveled.

For the detection framework, the intuition is that the anomalous trajectories do not follow normal routes, and thus cannot be *well-generated* from normal routes using the generative network. Therefore, we detect a trajectory by computing its likelihood of

being generated from normal routes, where the likelihood can be computed online and updated in $O(1)$ time. Nevertheless, the time cost of trajectory generation using GM-VSAE would increase by a factor of the number of components in the Gaussian mixture distribution. To this end, we further propose an approximate posterior inference method that selects only one component from the mixture distribution to generate and detect a trajectory, which significantly reduces the time cost of GM-VSAE. By doing this, the proposed detection-via-generation scheme enables efficient online anomaly detection.

Overall, our contributions can be summarized as follows:

- We develop a deep generative model GM-VSAE, which is powerful at capturing the sequential information of trajectories and represent their routes in a continuous latent space. We jointly use a Gaussian mixture distribution to model the route representations, which allow us to discover different types of normal routes to support effective anomalous trajectory detection.
- We propose a novel paradigm, *detection-via-generation*, for anomalous trajectory detection using GM-VSAE. To the best of our knowledge, we are the first to use the generation scheme to detect anomalous trajectories, which can be computed online and updated in $O(1)$ time. In addition, we propose an approximate posterior inference method that largely reduces the time cost of GM-VSAE and enables a more efficient online detection.
- We design an all-around evaluation procedure on two large-scale real-world trajectory datasets. We conduct extensive experiments on detecting anomalous trajectories with both partial and complete observations of the trajectories. The results show that our proposed GM-VSAE model is more effective than the state-of-the-art baselines and is efficient for online detection.

This chapter is organized as follows. Section 4.1 describes the problem of anomalous trajectory detection, and gives an overview of our proposed solution. Section 4.2 elaborates the proposed deep generative model (i.e., GM-VSAE). Section 4.3 introduces the detection framework based on GM-VSAE, and Section 4.4 presents an approximate inference method to further improve the detection efficiency. Section 4.5 evaluates the performance of our proposed solution. Finally, Section 4.6 summarizes this chapter.

4.1 Problem Formulation

4.1.1 Problem Definition

A trajectory is defined as a sequence of chronologically ordered points $P = \{p_1 \rightarrow p_2 \rightarrow \dots \rightarrow p_n\}$, where each point p_i is represented by GPS coordinates. Intuitively, a

trajectory can be considered as a representative (i.e., sample) of a *route*, which represents the continuous path in the physical world the trajectory follows.

As illustrated in Figure 1.4, there usually exist *normal routes* within a specific travel itinerary, which have relatively high probability to be traveled by trajectories (e.g., r_1 and r_2 between S_1 and D_1 in Figure 1.4). Based on this, we can intuitively consider those trajectories that *do not* follow those normal routes as *anomalous trajectories*, such as T_1 and T_2 in Figure 1.4.

More formally, we formulate these intuitions using probabilities as follows:

- We define the probability of a route r being traveled by trajectories as $p(r)$. Higher $p(r)$ means r is more likely to be a normal route.
- We specify the probability of a trajectory T following a normal route (denoted as r_*) as $p(T|r_*)$. Lower $p(T|r_*)$ means T is more likely to be anomalous.

Next, we follow the previous studies [LZC⁺18a, ZLZ⁺11] to evenly partition a geographical space into discrete grids (denoted as $\mathcal{G}$) and map the GPS coordinates to the grids. By doing this, a trajectory can be represented by a sequence of grid tokens, i.e., $T = \{t_1 \rightarrow t_2 \rightarrow \dots \rightarrow t_n\}$. Other methods such as map matching [NK09] to tokenize GPS coordinates is also applicable [WSZ17]. We represent the *source* and the *destination* of a trajectory T as $S_T = t_1$ and $D_T = t_n$. Note that S_T and D_T are both given *before* a trajectory T starts to be recorded [ZLZ⁺11]. Finally, we formulate the problem of online anomalous trajectory detection as:

Definition 1 (Online Anomalous Trajectory Detection) *Given an ongoing trajectory T being generated, its source S_T and destination D_T , the purpose of Online Anomalous Trajectory Detection problem is to 1) discover the normal routes between S_T and D_T and, 2) compute and update the probability of T following the normal route, i.e., $p(T|r_*)$, while T is sequentially generated.*

4.1.2 Overview of Our Solution

After formulating the problem, our main goals of designing an anomalous trajectory detection method become

- discovering normal routes by modeling $p(r)$, and
- detecting anomalous trajectories by modeling $p(T|r_*)$.

Table 4.1: Description of notations.

Symbol	Description
T	the token sequence of a trajectory
t_i	the i th grid token of a trajectory
n	the length of a trajectory
(S, D)	a pair of source and destination
$\mathcal{G}$	a set of all the grids on a geographical space
$\mathbf{r}$	the representation of a route in the latent space
k	the dimensionality of the RNN hidden vectors
M	the dimensionality of the latent space of route
C	the number of route types
ϕ	the parameters of the inference network
γ	the parameters of the latent Gaussian mixture distribution
θ	the parameters of the generative network

To achieve these goals, we develop a novel solution to solve the online anomalous trajectory detection problem based on a deep generative model, namely Gaussian mixture variational sequence autoencoder (GM-VSAE), which is inspired by previous studies on variational autoencoder (VAE) [KW13, JZT⁺17].

In particular, we first learn a GM-VSAE model that can 1) infer and represent routes of trajectories as vectors in a latent embedding space, 2) model the probability distribution of the routes (i.e., modeling $p(r)$), and 3) generate trajectories given a specific route (i.e., modeling $p(T|r)$).

Based on the learned GM-VSAE model, we develop an efficient anomalous trajectory detection framework. The framework can 1) discover normal routes by exploiting the learned probability distribution of routes, and 2) detect a trajectory via computing its probability of being generated from normal routes. Furthermore, we propose an approximation module, namely SD-network (SDN), to further improve the detection efficiency.

Our solution enables an effective and efficient detection of anomalous trajectories. In the following, we introduce GM-VSAE, the detection framework and the SD-network module in Section 4.2, 4.3 and 4.4, respectively.

4.2 Gaussian Mixture Variational Sequence Autoencoder

Figure 4.1 shows the architecture of the proposed GM-VSAE, which consists of the following components:

Figure 4.1: The overall architecture of GM-VSAE.

- **Inference network.** The inference network aims to infer the route of a given trajectory T and representing it as a vector $\mathbf{r}_T$ in a continuous latent space.
- **Probability distribution of routes.** We use a probability distribution to model the routes and represent how likely a route can be considered as a normal route.
- **Generative network.** The generative network is designed with the ability of generating trajectories from a given route.

For GM-VSAE, our main goal is to jointly train these components, such that 1) the probability distribution can well characterize the routes of trajectories in the latent space, and 2) the generative network can generate trajectories from latent routes. The probability distribution of routes and the generative network will be subsequently used for online anomalous trajectory detection in Section 4.3. In the rest of this paper, we use the terms “*latent route*”, “*latent route space*” and “*latent route distribution*” to denote the vector representation of routes, the latent space and the probability distribution of routes, respectively. Table 4.1 lists the notations used in the model. In the next three subsections, we elaborate each of the three components.

4.2.1 Latent route inference.

Before we exploit routes and route distribution for anomalous trajectory detection, we need to infer and represent the routes of trajectories. To achieve this, we propose a route inference network $q_\phi(\mathbf{r}|T)$ to infer the latent route (i.e., the route representation vector) $\mathbf{r}_T \in \mathbb{R}^M$ for a given trajectory $T = (t_1, t_2, \dots, t_n)$. Here, ϕ represents the parameters of $q_\phi(\mathbf{r}|T)$ and M represents the dimensionality of the latent route space.

In particular, we implement $q_\phi(\mathbf{r}|T)$ with a recurrent neural network (RNN), which accepts variable-length trajectories as inputs and can capture the sequential information behind trajectories. Specifically at each step i , the inference RNN reads t_i and produces a hidden state $\mathbf{h}_i \in \mathbb{R}^k$ by:

$$\mathbf{h}_i = f_1(t_i, \mathbf{h}_{i-1}) \quad i = 1, 2, \dots, n, \quad (4.1)$$

where f_1 is a deterministic non-linear function to be learned, which can be implemented with Long Short-Term Memory (LSTM) [HS97] or Gated Recurrent Unit (GRU) [CVMG⁺14]. By doing this, the sequential information of T is encoded in the latent representation $\mathbf{h}_n$. Note that the RNN accepts inputs in the form of real-valued vectors. Thus, we introduce a token embedding layer to embed the discrete token (i.e., t_i) in a vector.

To cope with the possible uncertainties and noise (e.g., GPS error) behind trajectory T , we infer $\mathbf{r}_T$ by drawing from a posterior distribution as

$$\mathbf{r}_T \sim q_\phi(\mathbf{r}|T) = \mathcal{N}(\boldsymbol{\mu}_T, \boldsymbol{\sigma}_T^2 \mathbf{I}) \quad (4.2)$$

where $\boldsymbol{\mu}_T \in \mathbb{R}^M$, $\boldsymbol{\sigma}_T \in \mathbb{R}^M$ and $\{\boldsymbol{\mu}_T, \boldsymbol{\sigma}_T\} = g_1(\mathbf{h}_n)$. The function $g_1(\cdot)$ is a non-linear function to be learned and $\mathbf{I}$ is an identity matrix. With the help of RNN, we are able to capture the sequential information of trajectories and represent their latent routes in the latent route space.

4.2.2 Modeling latent route with Gaussian mixture distribution.

With the ability of inferring and representing the routes of trajectories in the latent route space, we still lack knowledge about which routes in the space are *normal routes*, or which routes are more normal than the others. Motivated by this, we aim at modeling the latent routes with a probability distribution, which can be used to measure how likely a given route can be considered as a normal route. However, this is non-trivial as the routes in the real world are usually diverse and complicated. For example, vehicles may travel in different areas in an urban city (e.g., downtown, suburb) and on different types of roads (e.g., major/minor highway, ramp, street). It is non-trivial to model such complex route distribution.

To tackle this challenge, we propose to model the latent route distribution with a *Gaussian mixture distribution*, which is comprehensive for modeling various types of latent routes. In particular, we first assume there are C different *types* of routes underlying the trajectory data, where C is a hyper-parameter of the model. Different types of routes have different semantic meanings (e.g., road type, traveling area). To model different types of latent routes, we introduce two probability distributions:

- A multinomial distribution $p_\gamma(c) = \text{Mult}(\boldsymbol{\pi})$, where $\boldsymbol{\pi} \in \mathbb{R}^C$ are the parameters. It models the probability of a specific type (denoted as c) of routes being traveled by trajectories.
- A Gaussian distribution $p_\gamma(\mathbf{r}|c) = \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2 \mathbf{I})$ for each type c , where $\boldsymbol{\mu}_c \in \mathbb{R}^M$ and $\boldsymbol{\sigma}_c \in \mathbb{R}^M$ are the mean and standard deviation vectors. It models the probability of a route $\mathbf{r}$ of type c being traveled by trajectories.

Here, we use $\gamma = \{\boldsymbol{\pi}, \boldsymbol{\mu}_c, \boldsymbol{\sigma}_c\}$ to denote all the parameters in $p_\gamma(c)$ and $p_\gamma(\mathbf{r}|c)$. These two distributions together model the latent route space as $p_\gamma(\mathbf{r}) = p_\gamma(\mathbf{r}|c)p_\gamma(c)$.

After incorporating the latent route type c , we need to extend the inference network in Eq. (4.2) to $q_\phi(\mathbf{r}, c|T)$ to infer the route type c in addition to the latent route $\mathbf{r}$. Specifically, we apply the mean-field approximation to factorize $q_\phi(\mathbf{r}, c|T)$ as

$$q_\phi(\mathbf{r}, c|T) = q_\phi(\mathbf{r}|T)q_\phi(c|T), \quad (4.3)$$

where $q_\phi(\mathbf{r}|T)$ is defined in Eq. (4.2). For $q_\phi(c|T)$, it can be intuitively considered as the probability of route type c given the route of a trajectory T , i.e., $q_\phi(c|T) := p_\gamma(c|\mathbf{r}_T)$. Thus, $q_\phi(c|T)$ can be defined as

$$q_\phi(c|T) := p_\gamma(c|\mathbf{r}_T) = \frac{p_\gamma(c)p_\gamma(\mathbf{r}_T|c)}{\sum_{i=1}^C p_\gamma(c_i)p_\gamma(\mathbf{r}_T|c_i)}, \quad (4.4)$$

where $\mathbf{r}_T$ is drawn from $q_\phi(\mathbf{r}|T)$ as in Eq. (4.2). The intuition is that the probability of T traveling a route of type c (i.e., $p_\gamma(c|\mathbf{r}_T)$) is proportional to the probability $p_\gamma(\mathbf{r}_T|c)$.

4.2.3 Recurrent trajectory generation.

Next, we need to formalize the probability of a trajectory T following a given latent route $\mathbf{r}$, which enables the detection of anomalous trajectories in Section 4.3. To achieve this, we design a novel generative network $p_\theta(T|\mathbf{r})$, where θ is the set of parameters. The objective is to optimize $p_\theta(T|\mathbf{r})$ such that a trajectory T can be well generated from its inferred latent route $\mathbf{r}_T$.

However, different from other generation tasks (e.g., image generation), the generation of a trajectory is naturally sequential, i.e., the mobility of an object at a specific step

is correlated with its previous trace. Therefore, we resort to a recurrent generation process with the help from an RNN. Specifically at each step i , we consider that the generation of t_i is related to 1) the latent route $\mathbf{r}$, and 2) the previous sequence $t_{<i}$, where $t_{<i} = \{t_1 \rightarrow t_2 \rightarrow \dots \rightarrow t_{i-1}\}$. Thus, we generate t_i via $t_i \sim p_\theta(t_i|t_{<i}, \mathbf{r})$. In particular, we first recurrently encode the information of $\mathbf{r}$ and the sequence $t_{\leq i} = \{t_1 \rightarrow t_2 \rightarrow \dots \rightarrow t_i\}$ into a hidden vector $\mathbf{g}_i \in \mathbb{R}^k$ as:

$$\mathbf{g}_i = f_2(t_i, \mathbf{g}_{i-1}) \quad i = 1, 2, \dots, n \quad \mathbf{g}_0 = \mathbf{r}, \quad (4.5)$$

where f_2 is a deterministic non-linear function to be learned (e.g., GRU cell, LSTM cell). Note that the same token embedding layer that is used in the inference network is also applied here before feeding inputs to the RNN. Based on these, at each step i , we can generate t_i from a multinomial distribution as

$$t_i \sim p_\theta(t|t_{<i}, \mathbf{r}) = p_\theta(t|\mathbf{g}_{i-1}) = \text{Mult}(\text{softmax}(g_2(\mathbf{g}_{i-1}))). \quad (4.6)$$

Here, the output function $g_2(\cdot)$ converts $\mathbf{g}_{i-1}$ to a $|\mathcal{G}|$ -dimensional vector, where $|\mathcal{G}|$ denotes the total number of grid tokens. Then, we use a softmax function to get a probability vector to parameterize the multinomial distribution. By doing this at each time step, a trajectory can be recurrently generated.

4.2.4 Training GM-VSAE

Objective of joint optimization. The parameters to be learned in GM-VSAE include $\phi = \{f_1(\cdot), g_1(\cdot)\}$, $\gamma = \{\boldsymbol{\pi}, \boldsymbol{\mu}_c, \boldsymbol{\sigma}_c\}$, $\theta = \{f_2(\cdot), g_2(\cdot)\}$ and the token embeddings. Intuitively, the objective of training GM-VSAE is to jointly optimize the inference and generative networks, and the latent route distribution, such that the generative network can properly generate a trajectory with its route inferred by the inference network. More formally, the objective is to maximize the marginal log-likelihood of generating the training data:

$$\log p_\theta(T^{(1)}, T^{(2)}, \dots, T^{(N)}) = \sum_{j=1}^N \log p_\theta(T^{(j)}), \quad (4.7)$$

where $T^{(1)}, T^{(2)}, \dots, T^{(N)}$ are the N trajectories in the training data. We derive the evidence lower bound on the marginal likelihood of each trajectory T (denoted as $\mathcal{L}(\theta, \gamma, \phi; T)$)

as

$$\begin{aligned}
 \log p_\theta(T) &\geq \mathcal{L}(\theta, \gamma, \phi; T) = \mathbb{E}_{q_\phi(\mathbf{r}, c|T)} \left[\log \frac{p_{\theta, \gamma}(T, \mathbf{r}, c)}{q_\phi(\mathbf{r}, c|T)} \right] \\
 &= \mathbb{E}_{q_\phi(\mathbf{r}, c|T)} \left[-\log q_\phi(\mathbf{r}, c|T) + \log p_{\theta, \gamma}(T, \mathbf{r}, c) \right] \\
 &= \mathbb{E}_{q_\phi(\mathbf{r}|T)q_\phi(c|T)} \left[-\log q_\phi(\mathbf{r}|T) - \log q_\phi(c|T) \right. \\
 &\quad \left. + \log p_\theta(T|\mathbf{r}) + \log p_\gamma(\mathbf{r}|c) + \log p_\gamma(c) \right] \\
 &= \mathbb{E}_{q_\phi(\mathbf{r}|T)} [\log p_\theta(T|\mathbf{r})] - \mathbb{E}_{q_\phi(\mathbf{r}|T)} [D_{KL}(q_\phi(c|T)||p_\gamma(c))] \\
 &\quad - \mathbb{E}_{q_\phi(c|T)} [D_{KL}(q_\phi(\mathbf{r}|T)||p_\gamma(\mathbf{r}|c))], \tag{4.8}
 \end{aligned}$$

where $D_{KL}(\cdot||\cdot)$ represents the KL-divergence between two distributions. For the log-probability $\log p_\theta(T|\mathbf{r})$, we can further decompose it as the sum of the log-likelihood of generating each $t_i \in T$, i.e., $\log p_\theta(T|\mathbf{r}) = \sum_{i=1}^n \log p_\theta(t_i|t_{<i}, \mathbf{r})$. Therefore, we can rewrite the first term in Eq. (4.8) as

$$\mathbb{E}_{q_\phi(\mathbf{r}|T)} [\log p_\theta(T|\mathbf{r})] = \mathbb{E}_{q_\phi(\mathbf{r}|T)} \left[\sum_{i=1}^n \log p_\theta(t_i|t_{<i}, \mathbf{r}) \right]. \tag{4.9}$$

We train GM-VSAE via maximizing the evidence lower bound.

Alternative optimization scheme. To maximize $\mathcal{L}(\theta, \gamma, \phi; T)$, we need to optimize three sets of parameters, i.e., ϕ , γ and θ , which parameterize the inference network, latent route distribution and the generative network, respectively. Jointly optimizing all the parameters would be difficult and unstable. Therefore, we adopt an alternative optimization scheme, which alternatively updates two sets of parameters $\{\phi, \theta\}$ and γ . We apply reparameterization trick and Stochastic Gradient Variational Bayes (SGVB) estimator [KW13] to train the model in an end-to-end manner with alternative gradient decent.

4.3 Efficient Online Anomalous Trajectory Detection via Trajectory Generation

In this section, we design a novel *detection-via-generation* framework for online anomalous trajectory detection by exploiting the probability distribution $p_\gamma(\mathbf{r}|c)$ and the generative network $p_\theta(T|\mathbf{r})$ that are learned by GM-VSAE.

In particular, we first consider *normal routes* as those routes which are more likely to be traveled by trajectories. With the learned Gaussian probability distribution $p_\gamma(\mathbf{r}|c) = \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2 \mathbf{I})$, the normal routes of a specific type c can be considered as the routes that are close to $\boldsymbol{\mu}_c$, i.e., $\mathbf{r}_*^c \simeq \boldsymbol{\mu}_c$. Based on this, we use $\boldsymbol{\mu}_c$ as normal routes to approximate $\mathbf{r}_*^c$. We will analyze the accuracy of this approximation later.

Next, we need to detect the probability of a given trajectory T following μ_c . By leveraging the generative network $p_\theta(T|\mathbf{r})$, we propose a *detection-via-generation* scheme, where the probability of T following μ_c is implemented as the probability of T being generated from μ_c . The main advantage of this scheme is that the detection can be updated online while the trajectory is being sequentially generated, which is very efficiency. We will analyze the time complexity in the end of this section.

In particular, we define the anomaly score of a given trajectory T using $p_\theta(T|\mu_c)$ as

$$s(T) = 1 - \arg \max_c \exp \left[\frac{\log p_\theta(T|\mu_c)}{n} \right], \quad (4.10)$$

where $\exp(\cdot)$ is the exponential function and $\log p_\theta(T|\mu_c) = \sum_{i=1}^n \log p_\theta(t_i|t_{<i}, \mu_c)$. Figure 4.2 illustrates the detection procedure using GM-VSAE. Note that $\log p_\theta(T|\mu_c)$ is divided by n , i.e., the length of a trajectory T , to normalize trajectories with different lengths.

For online detection, i.e., detecting an ongoing trajectory (denoted as $t_{\leq i} = \{t_1 \rightarrow t_2 \rightarrow \dots \rightarrow t_i \rightarrow\}$), we maintain the probability $p_\theta(t_{\leq i}|\mu_c)$ for all μ_c . When a new location point is recored, i.e., $t_{\leq i}$ becomes $t_{\leq i+1}$, we can derive the new anomaly score as

$$s(t_{\leq i+1}) = 1 - \arg \max_c \exp \left[\frac{\log p_\theta(t_{\leq i}|\mu_c)p_\theta(t_{i+1}|t_{\leq i}, \mu_c)}{i+1} \right], \quad (4.11)$$

Here, we have $s(T) \in [0.0, 1.0]$ and high $s(T)$ indicates that T *cannot* be well generated from *any* type of normal route (i.e., has low $p_\theta(T|\mu_c)$ values), and thus is more likely to be an anomaly. To have a deterministic definition of anomalous trajectories, one can set a threshold Δ on the anomaly score, i.e., T is detected to be anomalous if $s(T) > \Delta$.

Approximation ratio. We analyze the accuracy of using μ_c to approximate $\mathbf{r}_*^c$. Based on Bayes' theorem, we have the following equation

$$\frac{p_\gamma(\mathbf{r}_*^c|c)p_\theta(T|\mathbf{r}_*^c)}{q_\phi(\mathbf{r}_*^c|T)} = p_{\gamma,\theta}(T|c) = \frac{p_\gamma(\mu_c|c)p_\theta(T|\mu_c)}{q_\phi(\mu_c|T)}. \quad (4.12)$$

Thus, we derive the approximation ratio τ as

$$\tau = \frac{p_\theta(T|\mu_c)}{p_\theta(T|\mathbf{r}_*^c)} = \frac{p_\gamma(\mathbf{r}_*^c|c)q_\phi(\mu_c|T)}{p_\gamma(\mu_c|c)q_\phi(\mathbf{r}_*^c|T)}. \quad (4.13)$$

Based on $\mathbf{r}_*^c \simeq \mu_c$, we denote $\mathbf{r}_*^c = \mu_c + \delta_*$, where δ_* is a small offset. We substitute the probability density functions of all the Gaussian distributions in Eq. (4.13) to finalize τ as

$$\tau = \exp \left[\frac{\delta_*^2(\sigma_c^2 - \sigma_T^2) + 2\delta_*\sigma_c^2(\mu_c - \mu_T)}{2\sigma_c^2\sigma_T^2} \right]. \quad (4.14)$$

We can see from Eq. (4.14) that τ is related to δ_* . If $\delta_* \rightarrow \mathbf{0}$, we would have a good approximation ratio, i.e., $\tau \rightarrow 1$. We consider a latent route $\mathbf{r}$ with a very high

Figure 4.2: The framework of anomalous trajectory detection using GM-VSAE.

Figure 4.3: The framework of anomalous trajectory detection using SD-VSAE.

probability $p_\gamma(\mathbf{r}|c)$ as a normal route $\mathbf{r}_*^c$. This implies $\mathbf{r}_*^c \rightarrow \mu_c$ and $\delta_* \rightarrow \mathbf{0}$. Therefore, δ has a very small value empirically, and can give us a good approximation ratio $\tau \rightarrow 1$.

Complexity analysis. To compute $s(T)$, we need to generate T for C times, i.e., using different μ_c . According to Eq. (4.10), the time complexity of computing $s(T)$ is $O(Ck^2n)$, where $O(k^2)$ is the complexity of updating the RNN hidden state $\mathbf{g}_i \in \mathbb{R}^k$ in each step $i = 1, 2, \dots, n$. Because k and C are both constants, the time complexity of computing $s(T)$ is therefore $O(n)$. For online detection, the time complexity of updating $s(T)$ for an ongoing trajectory is $O(Ck^2)$. It is because updating the log-likelihood $\log p_\theta(T|\mu_c)$ only requires to compute the RNN hidden state $\mathbf{g}_{i+1}$ for the new step $i + 1$ and then compute $p_\theta(t_{i+1}|\mathbf{g}_i)$. We omit the constants k and C and finalize the time complexity of updating $s(T)$ as $O(1)$.

4.4 Improving Efficiency with Approximate Inference of Route Type

Although GM-VSAE has a constant time complexity for online detection, the time cost is related to C , i.e., the number of Gaussian components in the latent route distribution. If C is large, the detection would still be slow. To address this issue, we proposed to infer the route type c_T^* that is the most likely to generate a trajectory T . By doing this, we only need to generate (i.e., detect) a trajectory from only *one* route type $\mu_{c_T^*}$.

A straightforward way is to use the route type c with the maximum posterior probability $q_\phi(c|T)$, i.e. $c_T^* = \arg \max_c q_\phi(c|T)$. However, this is not practical for online detection due to two reasons. First, the posterior inference $q_\phi(c|T)$ might be inaccurate with partial trajectory, i.e., while the trajectory is not complete. Second, $q_\phi(c|T)$ might be updated when a new location is recorded for the ongoing trajectory T . If $c_T^* = \arg \max_c q_\phi(c|T)$ is changed after updating $q_\phi(c|T)$, we need to re-generate T with the new c_T^* to update the anomaly score, which takes $O(n)$ time in the worst case for online detection.

To this end, we propose an approximation method to infer c_T^* for a trajectory T and achieve $O(1)$ time complexity for updating the anomaly score. We name the model SD-VSAE, which is extended from GM-VSAE. The intuition is that the normal routes between a specific source-destination pair (S, D) usually have the same or similar type. For example, vehicles usually drive through highway between two distant places. Motivated by this, we propose to infer c_T^* with S_T and D_T . As S_T and D_T are given *before* T is being generated, the inference of c_T^* can also be done beforehand.

SD-network. We propose a new module, namely SD-network (SDN), to distill the information of posterior $q_\phi(c|T)$ into a new probability distribution $q_\eta(c|S_T, D_T)$, which is parameterized by η . Specifically, we first introduce two extra embedding layers of grid tokens, i.e., S -embedding layer and D -embedding layer, respectively. Therefore, we can represent the S_T and D_T of a trajectory T as continuous vectors $\mathbf{s}_T$ and $\mathbf{d}_T$, respectively. Then, we concatenate $\mathbf{s}_T$ and $\mathbf{d}_T$ and feed it to a multi-layer perceptron as $MLP_L([\mathbf{s}_T, \mathbf{d}_T])$ where L denotes the number of layers in the MLP. We compute the probability values on the C types of routes as

$$q_\eta(c|S_T, D_T) = \text{softmax}(\mathbf{o}_L), \quad (4.15)$$

where $\mathbf{o}_L \in \mathbb{R}^C$ represents the final-layer output of the MLP. Since computing $q_\eta(c|S_T, D_T)$ can be done before T is being generated and does not need to be recurrently updated like $q_\phi(c|T)$, the time complexity of inferring c_T^* using $q_\eta(c|S_T, D_T)$ is $O(1)$, which is much more efficient than using $q_\phi(c|T)$, whose time complexity is $O(n)$.

Training SDN via knowledge distillation. We train the SDN module via knowledge distillation, i.e., using the original posterior of route type $q_\phi(c|T)$ as the supervision to

Table 4.2: Trajectory datasets after preprocessing.

Dataset	#Points	#Trajectories	#(S, D)	Avg. n
Porto	11,635,104	262,574	4,567	44.31
Beijing	909,642	52,497	6,752	17.32

train $q_\eta(c|S_T, D_T)$. In particular, we add one more step in the alternative optimization of $\mathcal{L}(\theta, \gamma, \phi; T)$. The objective of training SDN is to minimize the cross entropy between $q_\eta(c|S_T, D_T)$ and $q_\phi(c|T)$, i.e., $H(q_\phi, q_\eta) = -\sum_{c=1}^C q_\phi(c|T) \log q_\eta(c|S_T, D_T)$.

After training SD-VSAE, we can replace $q_\phi(c|T)$ with $q_\eta(c|S_T, D_T)$ to infer c_T^* . Subsequently, the online trajectory generation (i.e., detection) can be done with the inferred route type $\mu_{c_T^*}$, where $c_T^* = \arg \max_c q_\eta(c|S_T, D_T)$. SD-VSAE is therefore C times faster than GM-VSAE. Figure 4.3 summarizes the detection procedure using SD-VSAE.

4.5 Experiments

4.5.1 Experimental Setup

Datasets & Preprocessing. Our experiments are conducted on two real-world taxi trajectory datasets. In particular, the first dataset (Porto)¹ is generated from 442 taxis running in the city of Porto during Jan 07 2013 to Jun 30 2014. Each taxi reports its location at 15 second intervals. Following previous study [LZC⁺18a], we partition the geographical space into 100m×100m grids and filter those (S, D) pairs with fewer than 25 trajectories. The filtered dataset contains 262,574 trajectories. The second dataset (Beijing) is a subset of T-Drive trajectory dataset² [YZXS11, YZZ⁺10a], which contains a one-week trajectories of 10,357 taxis. We partition the geographical space into 300m×300m grids and select the grid cells with the highest trajectory density (e.g., top-200 popular cells) as S or D . We use the trips between the popular cells as the trajectories in the experiments. To choose trajectories with reasonable traveling distance and sampling rate, we further select the trajectories with a length of at least 10 and with time gaps between consecutive sample points being between 10 seconds and 10 minutes. We also filter those (S, D) pairs with fewer than 5 trajectories and finally obtain a dataset with 52,497 trajectories. Table 4.2 presents the statistics of the filtered datasets.

Note that the two datasets are very different from each other. First, the trajectories of Porto data are recorded with a fixed sampling rate (i.e., 1 point/15s) and is very dense, while the trajectories of Beijing data (i.e., T-Drive) is sampled with non-uniform time gaps, which is more sparse and noisy (even after preprocessing). Second, the urban area

¹<https://www.kaggle.com/c/pkdd-15-predict-taxi-service-trajectory-i/data>

²<https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/>

Figure 4.4: Examples of generated anomalies.

Table 4.3: Description of notations used in experiments.

Symbol	Description
α	the proportion of the detour segment of detour anomalies
d	the moving distance (i.e., offset) of the detour segment
β	the split location of generating route-switching anomalies
ρ	the proportion of a trajectory being observed for detection

of Beijing is larger than Porto, while the number of trajectories is fewer. Thus, Porto data reflects scenarios where the data are well-collected and less noisy, while Beijing data represents those where the data is sparse, noisy and spread over large geographical space. Evaluation on the two datasets can illustrate the effectiveness of different methods for different scenarios.

Ground truth. Since there is no labeled dataset available for anomalous trajectory detection, some existing studies [ZLZ⁺11, LXZ⁺17, CZC⁺13] try to manually label anomalies. However, it is very time-consuming and often limits the evaluation on just a few (S, D) pairs. Following previous work on spatial outlier detection [ZBLL17], we use *generated anomalies* for the evaluation.

In particular, we use two different perturbation schemes to generate two types of anomalous trajectories, namely detour anomalies (D) and route-switching anomalies (RS), respectively. The first perturbation scheme moves a random part of a trajectory along a direction that is roughly perpendicular to the moving direction of the trajectory. Figure 4.4(a) shows an example of a normal trajectory T_1 (blue line) and the generated anomaly T_2 (red line). We can see that it corresponds to the case where a moving object takes a long detour. The second perturbation scheme is exemplified in Figure 4.4(b). In particular, we first randomly select two trajectories T_3 and T_4 between the same

(S, D) , each of which can be divided into two sub-trajectories, i.e., $T_3 = T_3^{(1)} \rightarrow T_3^{(2)}$ and $T_4 = T_4^{(1)} \rightarrow T_4^{(2)}$, where $T^{(1)}$ and $T^{(2)}$ represent the first and the second part of T . Then, we replace $T_3^{(2)}$ with $T_4^{(2)}$, i.e., injecting an anomaly $T_5 = T_3^{(1)} \rightarrow T_4^{(2)}$ to the dataset.

We parameterize the injection of detour anomalies with two parameters d and α , where d represents the moving distance of the detour segment and α defines the proportion of the detour part. For example, $\alpha = 0.3$ and $d = 5$ means 30% of a trajectory is moved with a distance of 5 grids. We parameterize the injection of route-switching anomalies with β , which stands for the split location of a trajectory. For example, $\beta = 0.3$ means $T^{(1)}$ contains the earliest 30% points of T and $T^{(2)}$ contains the remaining 70%. The notations of the injection parameters are listed in Table 4.3. In the experiments, we vary d , α and β , and evaluate all the methods on detecting the two types of anomalies separately. Following previous studies [WSZ17, ZBLL17, ZLZ⁺11], the proportion of the injected anomalies in each dataset is set to 5%.

Baselines. We compare the following methods in our experiments:

- **TRAOD** [LHL08]. This method is based on a partition-and-detection framework. It first partitions all trajectories to segments. The anomaly score of a trajectory is based on the total length of its anomalous segments, which are distant to the other trajectories with the same (S, D) .
- **T-DBSCAN** [LXZ⁺17]. This method is a density-based method. It defines a route distance function and applies it for clustering trajectories with the same (S, D) using DBSCAN [EKS⁺96]. The anomaly score of a trajectory is computed as its distance to the closest “core trajectories”, which are found in the clustering procedure.
- **iBAT** [ZLZ⁺11]. This method detects anomaly by checking how much the target trajectory can be isolated from reference trajectories with same (S, D) pair. For a target trajectory, it randomly selects location tokens to remove the reference trajectories that do not pass them. The anomaly score is defined based on the number of sampled tokens to filter out all the reference trajectories.
- **DBTOD-embed** [WSZ17]. This method considers driving speed, road levels and turning angle as features and learns a probabilistic model that can automatically detect anomalous trajectories. However, to make a fair comparison to other baselines that do not use any side information, we use a grid token embedding vector to represent the features, which is jointly trained with the probabilistic model.
- **Sequence autoencoder (SAE)** [MRA⁺16]. This is a conventional RNN-based sequence to sequence model, which is learned via minimizing a reconstruction error. The anomaly score is defined based on the reconstruction error.

- **Variational sequence autoencoder (VSAE).** This is a degenerated model of GM-VSAE, where the latent route distribution is a Gaussian distribution.
- **GM-VSAE.** This is our proposed model. We set the number of Gaussian components in the latent route distribution as 10, 20 and 50 on Porto data, and 5, 10 and 20 on Beijing data, respectively. For example, GM-VSAE10 stands for the GM-VSAE model with 10 Gaussian components.
- **SD-VSAE.** This is extended from GM-VSAE model with SD-network module to improve the efficiency.

The baselines include representative metric based methods with different heuristic definitions (i.e., TRAOD, T-DBSCAN, iBAT), as well as learning based methods (i.e., DBTOD-embed, SAE and VSAE).

Implementation details. For GM-VSAE, we use GRU cells as f_1 and f_2 . We set the size of all embedding layers as 32, the size of GRU hidden states as $k = 256$ and the dimensionality of the latent route space $M = 256$. We use the same parameter setting for the baseline methods SAE and VSAE. For the SD-network module, we use a four-layer MLP, where the size of the layers are $\{64, 256, 256, C\}$. A ReLU non-linearity is applied between two consecutive layers. We train the models using Adam optimizer [KB14], where the minibatch size is set to 128 and the learning rate is set to 0.0001.

For each dataset, we randomly take 10% trajectories out as validation data and the remaining 90% trajectories as training data. The final model was chosen with early-stopping based on the validation loss. The parameters of other methods are also tuned to have the optimal results on the validation data. We use all the methods to detect the anomalies in the training trajectories for each dataset.

Evaluation metrics. The output of a detection method for each trajectory is an anomaly score between 0.0 and 1.0. For the effectiveness evaluation, we use Precision-Recall AUC (PR-AUC) as the metric [ZBLL17]. PR-AUC represents the area under the precision-recall curve, where the points on the curve are pairs of precision-recall values when varying the detection threshold $\Delta \in [0.0, 1.0]$ for the anomaly score. PR-AUC is appropriate to evaluate anomaly detection methods on skewed datasets [ZBLL17]. Note that the number of anomalies we inject into the trajectory datasets are only small portions of all the trajectories. We compute the PR-AUC value for each (S, D) and report the average values over all (S, D) .

4.5.2 Effectiveness Evaluation

Varying observed ratios. To evaluate the performance of *online* detection, we use each method to detect the anomalousness of a target trajectory with different *observed*

Table 4.4: Performance comparison for anomalous trajectory detection (detour) in terms of PR-AUC on Porto dataset.

	Detecting detour anomalies (D)								
Perturb params	$d=3; \alpha = 0.1$			$d=5; \alpha = 0.1$			$d=3; \alpha = 0.3$		
Observed ratio (ρ)	0.1	0.5	1.0	0.1	0.5	1.0	0.1	0.5	1.0
TRAOD	-	-	0.212	-	-	0.311	-	-	0.161
DBSCAN	-	-	0.231	-	-	0.305	-	-	0.253
iBAT	0.156	0.184	0.181	0.159	0.196	0.193	0.182	0.371	0.406
DBTOD-embed	0.148	0.146	0.277	0.148	0.141	0.279	0.159	0.297	0.384
SAE	0.152	0.46	0.717	0.154	0.502	0.824	0.176	0.666	0.921
VSAE	0.170	0.455	0.701	0.174	0.501	0.819	0.197	0.660	0.910
GM-VSAE10	0.230	0.473	0.773	0.234	0.513	0.842	0.233	0.702	0.961
GM-VSAE20	0.334	0.494	0.811	0.335	0.522	0.868	0.361	0.711	0.969
GM-VSAE50	0.338	0.498	0.811	0.337	0.523	0.868	0.358	0.714	0.963

ratios (denoted as $\rho \in [0.0, 1.0]$). For example, $\rho = 0.1$ means to detect a trajectory when we only observe the earliest 10% locations of the trajectory. We use $\rho = 1.0$ to represent the detection on complete trajectories. Note that TRAOD and T-DBSCAN cannot detect anomalous trajectories in an online manner. Therefore, we only evaluate them on detecting complete trajectories (i.e., $\rho = 1.0$).

Detecting detour anomalies. We first compare all the methods on detecting detour anomalies. In particular, we apply different parameters to inject detour anomalies in both datasets, where $d = \{3, 5\}$ and $\alpha = \{0.1, 0.3\}$. We set $\rho = \{0.1, 0.5, 1.0\}$ in the experiments. The trajectories in Beijing dataset are usually very short, i.e., 10% of a trajectory may only contain one or two points. Therefore, we only use $\rho = \{0.5, 1.0\}$ on Beijing. Tables 4.4 and 4.6 show the PR-AUC of detecting detour anomalies using different methods on Porto and Beijing datasets, respectively.

For detecting complete trajectories (i.e., $\rho = 1.0$), we can see that GM-VSAE is significantly better than all the baselines on both datasets. For example, when $d = 3$ and $\alpha = 0.1$, GM-VSAE20 outperforms the best baseline (i.e., SAE) by 13.1% on Porto, and GM-VSAE10 outperforms all the baselines by at least 8% on Beijing. We also find out that the improvements of VSAE over the distance and density based methods (e.g., TRAOD, T-DBSCAN) are more significant on Porto data than on Beijing data. For example, when $d = 5$ and $\alpha = 0.1$, VSAE improves TRAOD by 163.3% on Porto while only 56.3% on Beijing. This indicates that using RNN can more effectively capture useful information and reveal normal routes on large-scale dense data (i.e., Porto) than on sparse data (i.e., Beijing). Comparing with VSAE, GM-VSAE provides more accurate detection on both datasets. For example, GM-VSAE10 outperforms VSAE by 10.3% and 8.5%,

Table 4.5: Performance comparison for anomalous trajectory detection (route-switching) in terms of PR-AUC on Porto dataset.

	Detecting route-switing anomalies (RS)					
Perturb params	$\beta = 0.3$		$\beta = 0.5$		$\beta = 0.7$	
Observed ratio (ρ)	0.5	1.0	0.7	1.0	0.9	1.0
TRAOD	-	0.189	-	0.183	-	0.179
DBSCAN	-	0.240	-	0.245	-	0.201
iBAT	0.200	0.179	0.177	0.173	0.177	0.175
DBTOD-embed	0.138	0.285	0.171	0.292	0.240	0.292
SAE	0.469	0.396	0.459	0.424	0.440	0.426
VSAE	0.614	0.566	0.619	0.571	0.605	0.591
GM-VSAE10	0.641	0.574	0.636	0.606	0.634	0.618
GM-VSAE20	0.717	0.662	0.721	0.678	0.720	0.706
GM-VSAE50	0.715	0.660	0.725	0.705	0.718	0.707

on Porto and Beijing, respectively, when setting $d = 3$ and $\alpha = 0.1$. This is because GM-VSAE models latent routes using Gaussian mixture distribution, which can provide more useful insights (i.e., route types) to detect anomalous trajectories.

For detecting partial trajectories ($\rho = \{0.1, 0.5\}$), we can see that GM-VSAE is also the best among all the methods. For example, when $\rho = 0.5$, $d = 5$ and $\alpha = 0.1$, GM-VSAE10 outperforms iBAT and SAE by 21.5% and 7.1% on Beijing. Furthermore, at the very early stage of a trajectory (i.e., $\rho = 0.1$), the performance improvement of GM-VSAE is even better than the other models. For example, when $d = 5$ and $\alpha = 0.1$, GM-VSAE20 improves VSAE by 92.5% when $\rho = 0.1$, while only 6.0% when $\rho = 1.0$ on Porto data. This is because GM-VSAE mines the information about route types, which can provide extra confidence in anomaly detection even when the trajectory generation has just started. Hence, GM-VSAE would be more useful for launching early-warnings when detecting anomalous trajectories.

Detecting route switching anomalies. Tables 4.5 and 4.7 show the results of detecting route-switching anomalies using different methods on Porto and Beijing, respectively. In these experiments, we use $\beta = \{0.3, 0.5, 0.7\}$ to inject anomalies and report the results on detecting both partial and complete trajectories. For detecting partial trajectories, we set $\rho = \{0.5, 0.7, 0.9\}$ for different β values correspondingly. The results are qualitatively similar to the results of detecting detour anomalies. We observe that GM-VSAE outperforms all the baselines with respect to different β and ρ , in terms of PR-AUC. For example, GM-VSAE5 performs better than DBTOD-embed and T-DBSCAN by 46.8% and 51.7%, respectively, on Beijing when $\beta = 0.5$ and $\rho = 1.0$.

Table 4.6: Performance comparison for anomalous trajectory detection (detour) in terms of PR-AUC on Beijing dataset.

Perturb params	Detecting detour anomalies (D)					
	$d=3; \alpha = 0.1$		$d=5; \alpha = 0.1$		$d=3; \alpha = 0.3$	
Observed ratio (ρ)	0.5	1.0	0.5	1.0	0.5	1.0
TRAOD	-	0.333	-	0.343	-	0.265
DBSCAN	-	0.245	-	0.270	-	0.288
iBAT	0.383	0.352	0.396	0.364	0.413	0.367
DBTOD-embed	0.326	0.485	0.311	0.487	0.366	0.512
SAE	0.416	0.464	0.449	0.495	0.559	0.691
VSAE	0.421	0.506	0.461	0.536	0.588	0.741
GM-VSAE5	0.439	0.528	0.481	0.559	0.612	0.792
GM-VSAE10	0.454	0.549	0.481	0.598	0.599	0.799
GM-VSAE20	0.434	0.511	0.457	0.565	0.588	0.746

GM-VSAE vs. SD-VSAE. Next, we study the effectiveness of using SD-network (SDN) to distill the information of route types. We compare the effectiveness of GM-VSAE10 and SD-VSAE10 by detecting both detour anomalies (D) and route-switching anomalies (RS), where the injection parameters are set as $d = 3$, $\alpha = 0.3$ and $\beta = 0.3$. We can see from Figure 4.5 that SD-VSAE has comparable performance to GM-VSAE for detecting both types of anomalies on both datasets, in terms of PR-AUC. For example, when detecting detour anomalies (D) with complete trajectories ($\rho = 1.0$) on Porto data, the PR-AUC value of SD-VSAE only decreases by 0.94% compared to GM-VSAE. This indicates that our SDN module can effectively infer the route type for a given trajectory. It is worth noting that the effectiveness of SD-VSAE decreases less on Porto than Beijing. This is because each (S, D) in Porto has more trajectories, and thus can provide more information for inferring the route type between (S, D) .

4.5.3 Efficiency Evaluation

Figure 4.6 presents the average runtime of detecting a trajectory using all the methods on both datasets. Note that the y-axis of Figure 4.6 is in a *logarithmic* scale. Firstly, we can see that all the methods are faster on Beijing data, where the trajectories are shorter. Furthermore, DBTOD-embed is the fastest method for detecting anomalous trajectories, because it is a linear model with low-dimensional grid embeddings as features, where the detection can be accomplished very quickly. Among the other (i.e., non-linear) models, we can see that VSAE and SD-VSAE10 both have the lowest time cost for anomalous trajectory detection. For example, SD-VSAE10 takes only 0.433ms to detect a trajectory

Table 4.7: Performance comparison for anomalous trajectory detection (route-switching) in terms of PR-AUC on Beijing dataset.

	Detecting route-switing anomalies (RS)					
Perturb params	$\beta = 0.3$		$\beta = 0.5$		$\beta = 0.7$	
Observed ratio (ρ)	0.5	1.0	0.7	1.0	0.9	1.0
TRAOD	-	0.297	-	0.283	-	0.288
DBSCAN	-	0.422	-	0.451	-	0.458
iBAT	0.500	0.477	0.472	0.448	0.480	0.468
DBTOD-embed	0.345	0.446	0.354	0.466	0.414	0.475
SAE	0.458	0.425	0.476	0.461	0.464	0.459
VSAE	0.454	0.445	0.469	0.467	0.465	0.472
GM-VSAE5	0.665	0.650	0.667	0.684	0.659	0.663
GM-VSAE10	0.608	0.599	0.606	0.621	0.618	0.622
GM-VSAE20	0.576	0.556	0.586	0.596	0.591	0.589

on Porto, while the runtime of TRAOD and T-DBSCAN is around 40ms, which is over $100\times$ slower. VSAE is shown to be approximately $2\times$ faster than SAE, where their time costs are 0.757ms and 0.357ms, respectively, on Porto data. This is because SAE detects anomalies via trajectory reconstruction, and thus needs to process a given trajectory twice (i.e., encoding and decoding). However, VSAE uses the *detection-via-generation* scheme, which is free of the encoding (i.e., inference) step, and thus is more efficient. Moreover, we find out that SD-VSAE10 is approximately $10\times$ faster than GM-VSAE10. This is because when detecting a target trajectory, GM-VSAE10 needs to generate trajectories from 10 Gaussian components in the latent route distribution, while in SD-VSAE10, only one component selected by the SDN module is needed.

Furthermore, we compare the runtime of SAE, VSAE, GM-VSAE10 and SD-VSAE10 for detecting ongoing trajectories on both datasets. For detecting each trajectory, we record the time cost accumulated at different observed ratios (i.e., $\rho = \{0.2, 0.4, 0.6, 0.8, 1.0\}$). The reported runtime values are averaged on all the trajectories. Figure 4.7 shows that (1) the time cost of SAE increases quadratically with the generation of trajectories, while others increase linearly. This is because SAE needs to re-encode the ongoing trajectory when a new location is recorded, whose time complexity of updating the anomaly score is $O(n)$. (2) GM-VSAE10's accumulated time cost increases dramatically with more GPS points being observed for a trajectory; (3) As a trajectory is sequentially generated, the time costs for updating its anomaly score of SD-VSAE and VSAE are the lowest among all these methods. This shows the remarkable efficiency of SD-VSAE for online detection. Along with the effectiveness comparison between GM-VSAE and SD-VSAE shown in Figure 4.5, we can conclude that SD-VSAE is able to largely reduce the time cost of GM-VSAE for online anomalous trajectory detection, while having similar effectiveness.

 Figure 4.5: GM-VSAE10 vs. SD-VSAE10 (D: $d = 3$, $\alpha = 0.3$; RS: $\beta = 0.3$).

Figure 4.6: Efficiency comparison for anomalous trajectory detection.

4.5.4 Case study

We visualize the trajectories between 16 randomly selected (S, D) pairs on Porto data. Among the trajectories between each (S, D) pair, we highlight the trajectory with the highest anomaly score detected by SD-VSAE20 as shown in Figure 5.8. Note that in the case study, we do not inject generated anomalies into the dataset, i.e., all the trajectories shown are real-world trajectories. We can see that the anomalous trajectories (i.e., orange dashed lines) are intuitively very different from the majorities of other trajectories (i.e., blue solid lines that overlap) traveling from S to D . This demonstrate the superior capability of our method in detecting anomalous trajectories.

4.5.5 Visualization of Route Types

We demonstrate the different route types learned by GM-VSAE by visualizing trajectories that belong to a specific route type. Figure 4.9 visualizes four randomly selected route

Figure 4.7: Accumulated time cost when a trajectory is generated sequentially.

types from GM-VSAE20 on Porto data. We can see that the trajectories in different route types are different from each other. In particular, most of the trajectories from route type #15 are crowded in the downtown area, while the trajectories from route type #8 cover the highways outside of the central district. Moreover, the trajectories from route type #4 usually travel across the city and trajectories from route type #19 usually commute within the west side of Porto. Therefore, GM-VSAE is able to reveal route types with different semantic meanings, which is helpful for detecting anomalous trajectories.

4.6 Summary

In this chapter, we propose a novel deep generative model, namely GM-VSAE, to solve the problem of online anomalous trajectory detection. GM-VSAE can capture complex sequential information of trajectories and effectively model the probability distribution of their routes. To detect anomalous trajectories efficiently, we propose a novel scheme, i.e., anomaly detection via trajectory generation, which can work in an online manner. We also develop SD-VSAE that significantly improves the efficiency of GM-VSAE without losing the detection accuracy. Experiments on two large-scale datasets demonstrate the superiority of our methods over baseline methods in detecting anomalous trajectories.

Figure 4.8: Case study on Porto data (**the most anomalous trajectory**: orange dashed lines; **other trajectories**: blue solid lines).

Figure 4.9: Visualization of trajectories sampled from different route types (#15: trajectories in downtown area; #8: trajectories on highways; #19: trajectories on the west side of Porto; #4: trajectories across city).

Chapter 5

Similar Region Search

Various types of geo-spatial objects (e.g., POIs and tweets) can be considered as data sources for profiling, analyzing and retrieving *regions*. Here, a region is defined as a subarea that contains a set of geo-spatial objects on a large geographical space. In this chapter, we study a novel application, i.e., similar region search, which can be leveraged to support a variety of downstream applications. Given a query region, we aim at returning other regions that are the most similar to the query. However, this problem faces two major challenges: 1) How to model region similarity, which should consider the spatial information of the objects within regions; 2) How to efficiently search similar regions on large geographical space.

To support this application and address these challenges, we propose a novel solution for similar region search. To address the first challenge, we leverage a deep metric learning model, namely triplet network, to learn a ranking metric for comparing regions. It extracts features using Convolutional Neural Network (CNN), which can inherently capture local relationships between objects for similarity learning. We propose a method to generate training data with hard negative example mining, which makes the similarity metric robust to small noise and object shift. We also propose a ratio-based training loss that is robust to highly skewed spatial data.

To address the second challenge with the learned similarity metric, we first reduce the time cost of computing similarity by transforming the problem. We next propose a branch and bound algorithm, namely ExactSFRS, that greatly reduces the search space. In addition, we also propose an approximation method, namely ApproxSFRS, whose worst-case time complexity is adjustable. It can make trade-off between accuracy and efficiency to support different applications.

Overall, our contribution can be summarized as follows:

This chapter is published as *Yiding Liu, Kaiqi Zhao, Gao Cong. Efficient Similar Region Search with Deep Metric Learning. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (SIGKDD 2018) (Research Track – Oral), Pages 1850-1859, London, United Kingdom, August 19 - 23, 2018 [LZC18b].*

- We propose the first deep metric learning method to learn similarity between regions. The similarity not only considers the attributes of objects in a region, but also captures the relative locations between the objects.
- We propose an efficient search algorithm called ExactSFRS that solves the similar region search problem. We also propose an approximation method called ApproxSFRS that can strike a balance between accuracy and efficiency to support different applications.
- We apply the proposed methods to three large-scale datasets. The effectiveness experiments demonstrate our deep metric learning method is significantly better than the baselines for measuring region similarity. The efficiency experiments show that ExactSFRS is more than $10^5 \times$ faster than the exhaustive search method and over $45 \times$ faster than the best baseline method. Furthermore, we demonstrate ApproxSFRS can largely reduce the runtime by slightly sacrificing the accuracy.

This chapter is organized as follows. Section 5.1 defines the similar region search problem. Section 5.2 introduces the deep metric learning framework for region similarity learning. Section 5.3 presents two algorithms for efficient similar region search. Section 5.4 evaluates the effectiveness of the similarity metric and the efficiency of the searching algorithms. Section 5.5 summarizes this chapter.

5.1 Problem Definition

We consider a set of spatial objects $\mathcal{O}$ in a 2-dimensional rectangular geographical space P . Each object $o \in \mathcal{O}$ is associated with an attribute vector $\mathbf{a}_o \in \mathbb{R}^{1 \times d}$ and a geo-location $(o.x, o.y)$. The attributes of objects can be different in different applications, such as categories of POIs and vector representations of tweets. We define a region R as:

Definition 2 (Region) *A spatial rectangular subspace on P (i.e., $R \subset P$) bounded by (t, b, l, r) (i.e., top, bottom, left and right bound, respectively) that contains a set of objects $\mathcal{O}_R \subset \mathcal{O}$, where $\forall o \in \mathcal{O}_R, b < o.x < t, l < o.y < r$.*

Note that we extend $\subset$ to denote the inclusion relation between two regions, i.e., $R \subset P$ means R is contained in P .

For any two regions R_1 and R_2 , we define their similarity as $\text{sim}(R_1, R_2)$. The function $\text{sim}(\cdot)$ should consider both the attributes (i.e., $\mathbf{a}_o, \forall o \in \mathcal{O}_R$) and the relative locations of the objects in a region. Figure 5.1 shows an example of the definition of $\text{sim}(\cdot)$. Suppose we have a query region R_q and three candidate regions R_1, R_2 and R_3 . Different shapes represent different types of objects. In the three regions, R_3 is the most dissimilar

Figure 5.1: An example of defining $\text{sim}(\cdot)$ (R_q represents the query region, and R_1 is the most similar to R_q , as they have similar object attributes and relative locations; R_2 is less similar to R_q than R_1 , because the relative locations is dissimilar; R_3 is the most dissimilar to R_q).

Table 5.1: Notations for similarity learning.

Symbol	Description
R (or R_q)	a region (or a query region)
$\text{sim}(\cdot)$	the similarity metric
$\text{Net}(\cdot)$	the shared convolutional neural network
$g(\cdot)$	the feature aggregation function (max-pooling)
$\mathcal{X}_i$	the feature map of region R_i
$\mathbf{v}_i$	the aggregated feature vector or region R_i

region to R_q because it contains different types of objects to R_q . Both R_1 and R_2 contain the same number of objects from each type as R_q . However, the relative locations of the objects in R_1 are more similar to R_q , i.e., the three types of objects are co-located together, while the objects in R_2 are scattered. Thus, we have $\text{sim}(R_q, R_1) > \text{sim}(R_q, R_2) > \text{sim}(R_q, R_3)$.

Next, we formulate the similar region search (SRS) problem as:

Definition 3 (SRS problem) *Given a geographical space P , a query region R_q and a similarity metric $\text{sim}(\cdot)$, the **similar region search (SRS) problem** is to retrieve a set of N regions (denoted as $\mathcal{R}$), such that $\text{sim}(R_q, R_i) \geq \text{sim}(R_q, R_j), \forall R_i \in \mathcal{R}, \forall R_j \notin \mathcal{R}$.*

In this chapter, we take two steps to solve the *SRS* problem. We first propose a method to learn the similarity metric $\text{sim}(\cdot)$ in Section 5.2. Then, we develop efficient search algorithms based on the learned $\text{sim}(\cdot)$ in Section 5.3.

5.2 Region Similarity Learning

In this section, we prone to design $\text{sim}(\cdot)$ that (1) captures the attributes and the relative locations of objects, and (2) is robust to small noise and shift. Instead of designing a hand-crafted similarity metric, we present a deep metric learning method, namely triplet network, that directly learns $\text{sim}(\cdot)$ from data. Table 5.1 lists the notations for similarity learning.

Figure 5.2: The workflow of region similarity learning.

5.2.1 Ranking Metric Learning Model

In the *SRS* problem, it is hard to set a threshold value to distinguish similar and dissimilar. However, given a query region and two candidate regions, it is easier to tell which one is more similar to the query than the other one. Therefore, we propose to learn a ranking metric for regions.

Triplet network. Triplet network [HA15] is widely-used in Computer Vision to learn ranking metrics between images. A triplet network comprises three instances of a shared convolutional neural network (CNN) [KSH12] (denoted as $Net(\cdot)$). The input of a triplet network consists of: a query x_q , a positive example x_+ (i.e., similar) and a negative example x_- (i.e., dissimilar). When feeding the three examples, the triplet network generates two Euclidean distances $d_+ = ||Net(x_q) - Net(x_+)||_2$ and $d_- = ||Net(x_q) - Net(x_-)||_2$, where $Net(x)$ represents the feature map of x in the last layer of the CNN. Note that the fully-connected layer in the CNN is removed, as we are interested in a feature embedding only.

The goal of learning a triplet network is to train $Net(\cdot)$ such that the positive example has smaller distance than the negative example to the query [WG15]:

$$\mathcal{L} = \sum_{(x_q, x_+, x_-)} \max\{0, d_+ - d_- + \delta\} + \lambda ||Net(\cdot)||_2,$$

where δ is the gap parameter between two distances. $\lambda ||Net(\cdot)||_2$ is a L_2 regularization term for $Net(\cdot)$, which is parameterized by λ .

The CNN in triplet network is inherently appropriate for perceiving regions. In particular, convolutions can extract features of the local areas in a region, preserving the

information about object attributes and how they are locally distributed. Moreover, the feature extraction is performed in a hierarchical manner, which is similar to how objects are organized on geographical space. Thus, it is useful to capture the relative locations between the objects at different levels (i.e., from local to global) of a region. Therefore, we apply triplet network to region similarity learning.

To feed a region into $Net(\cdot)$, we partition it into grids with a predefined granularity (e.g., $10m \times 10m$). We associate each grid with a vector by summing the attribute vectors of the objects in the grid cell. We associate empty grids with $\mathbf{0}$. A region is therefore represented as a 3-d tensor, i.e., $R \in \mathbb{R}^{w \times h \times d}$, where the first two dimensions are the spatial dimensions of the grid cells and the last dimension represents the attributes. We denote the output feature map of $Net(R)$ as $\mathcal{X} \in \mathbb{R}^{w^* \times h^* \times K}$ (i.e., $\mathcal{X} = Net(R)$), where K is the number of dimensions of the output features. Each of the K dimensions is a latent feature that contains spatial information. To compare feature maps with different sizes, we further apply a feature aggregation layer (denoted as $g(\cdot)$) after $Net(\cdot)$ to aggregate each feature map to a K -dimensional feature vector, i.e., $\mathbf{v} = g(\mathcal{X})$, $\mathbf{v} \in \mathbb{R}^{1 \times K}$. In this chapter, we use a global max-pooling layer as $g(\cdot)$. Hence, the output distances are $d_+ = \|\mathbf{v}_q - \mathbf{v}_+\|_2$ and $d_- = \|\mathbf{v}_q - \mathbf{v}_-\|_2$. By using the learned metric, we define the similarity between two regions R_1 and R_2 as

$$sim(R_1, R_2) = \frac{1}{1 + \|\mathbf{v}_1 - \mathbf{v}_2\|_2}.$$

5.2.2 Training Robust Triplet Network

We next propose to train a triplet network on regions that is robust to small noise and shift. In solving visual recognition tasks, the robustness of a triplet network is usually achieved by learning from semantic labels of data. However, there is no labeled data for region similarity. To solve this problem, we propose a self-supervised learning scheme that directly mines robustness by learning self-similarity, where the “labels” are free to obtain.

Training data generation. The robustness of $sim(\cdot)$ can also be interpreted in another way: regions that differ in a small amount of noise and shift should be considered as similar. Taking a leaf from this intuition, we propose to generate the query region R_q , the negative input R_- and the positive input R_+ as follows.

We randomly sample regions with various size at arbitrary locations as R_q . For each R_q , we first generate R_+ by adding noise and shift to R_q , including (1) randomly removing objects, (2) adding random objects at random locations, and (3) randomly shifting the locations of the objects. We control the generation of noise and shift by a *noise rate* and a *shift rate*, respectively. In particular, setting the noise rate as 0.1 means randomly deleting 10% of the objects then adding the same number random objects at random

locations in the region. Setting the shift rate as 0.1 means moving each object along a random direction by a random distance, which is <10% of the region’s height and width. The specific values of noise rate and shift rate used in the experiments are introduced in the parameter setting of Section 5.4.

We next generate R_- for each R_q . Specifically, we randomly sample regions on the map that does not overlap with R_q as R_- . However, some of the sampled R_- may be too dissimilar to R_q . In this case, the trivial task of distinguishing R_+ and R_- leads the model to converge on a trivial solution. To solve this problem, we also generate R_- by applying slightly more noise to R_q than the noise applied for generating R_+ . By doing this, R_+ and R_- are both noisy versions of R_q , while R_+ contains less noise and thus more similar to R_q . We put a small portion of such hard examples in the training data. It pushes the model to learn more discriminative features to correctly distinguish R_+ and R_- and thus minimize the loss. Using the generated data, the model can learn features that are robust to small noise and random shift.

Ratio-based training loss. The numbers of objects in different regions are very different, which makes the triplet loss highly skewed. Particularly, some training tuples with very large loss values would overshadow other training data. A common way to solve this is to apply normalization to the input. However, this would loss the information about the number of objects in the regions, e.g., a region with 100 restaurants would be falsely considered similar to a region with 5 restaurants. Therefore, we resort to adjusting the loss computation. We propose a ratio-based training loss:

$$\mathcal{L} = \sum_{(R_q, R_+, R_-)} \max\{0, \frac{d_+}{d_+ + d_-} - \delta\} + \lambda \|Net(\cdot)\|_2.$$

The ratio-based loss for every training tuple is bounded in $[0, 1 - \delta]$, i.e., no training instance would excessively affect $\mathcal{L}$.

Figure 5.2 summarizes the process of training triplet network for regions. We propose to generate training tuples, feed them into a triplet network and train the model based on a ratio-based loss. The model can be optimized by commonly-used optimization methods, such as Stochastic Gradient Descent (SGD) and Adaptive Subgradient Methods (AdaGrad) [DHS11].

5.3 Region Search Algorithms

In this section, we propose efficient search algorithms to search similar region with the learned $sim(\cdot)$. For ease of presentation, Table 5.2 lists the notations for the search algorithms.

Table 5.2: Notations for the search algorithms.

Symbol	Description
R_c	a candidate region
$\mathcal{X}_P$	the feature map of the geographical space P
$\mathcal{S}$	a search space
(t^*, b^*, l^*, r^*)	the location of a feature region on $\mathcal{X}_P$
$\hat{f}(\cdot)$	the distance lower bound function

The geographical space P is a 2-dimensional continuous space, which contains an infinite number of regions and is infeasible to search. Therefore, we first divide P into grids with the same grid granularity used in the metric learning. A region is thus represented as a rectangular sub-area of grids in P .

Given a query R_q and the learned similarity metric $\text{sim}(\cdot)$, a naïve search method is to compare R_q with every candidate region $R_c \subset P$, i.e., computing $\text{sim}(R_q, R_c), \forall R_c \subset P$. However, this is very time-consuming due to two reasons:

- **Computing $\text{sim}(\cdot)$ is inefficient.** In particular, the main computational cost of $\text{sim}(R_q, R_c), \forall R_c \subset P$ lies in computing $\mathcal{X}_c = \text{Net}(R_c), \forall R_c \subset P$, which contains a lot of convolutions.
- **The number of candidate regions is numerous.** For a $30\text{km} \times 30\text{km}$ city with a grid size of $10\text{m} \times 10\text{m}$, the number of candidate regions is over 10^{13} . Even if we only search for regions whose height and width are within 500m to 3km, the number is still larger than 100 billion. Such number of data instances are impossible for exhaustive search and unmanageable for indexing methods like Locality Sensitive Hashing [DIIM04] and Kd-tree [Ben75].

Therefore, to answer *SRS* queries efficiently, we prone to solve the above-mentioned issues in this section. In Section 5.3.1, we first share the computation for $\text{Net}(R_c), \forall R_c \subset P$, transforming the *SRS* problem to a *Similar Feature Region Search (SFRS)* problem. Section 5.3.2 and Section 5.3.3 present our exact and approximate solutions for the *SFRS* problem, respectively.

5.3.1 Problem Transformation via Sharing Computation

Sharing computation. Computing $\mathcal{X}_c = \text{Net}(R_c), \forall R_c \subset P$ is slow because it performs forward pass through $\text{Net}(\cdot)$ for each region, without sharing computation for those who overlap with each other. Inspired by the idea of sharing computation in visual object recognition methods [HZRS14, Gir15, RHGS15], we precompute the feature map of the entire geographical space, i.e., $\mathcal{X}_P = \text{Net}(P)$, and share it for computing $\mathcal{X}_c$. Figure 5.3

 Figure 5.3: Sharing $\mathcal{X}_P$ for computing $\text{sim}(R_q, R_c)$.

shows the computation of $\text{sim}(R_q, R_c)$ via sharing $\mathcal{X}_P$. When a query comes, we first compute $\mathbf{v}_q = g(\text{Net}(R_q))$. Next, we project the location of R_c on P (i.e., (t, b, l, r)) to the location of $\mathcal{X}_c$ on $\mathcal{X}_P$ (denoted as (t^*, b^*, l^*, r^*)). We can get $\mathcal{X}_c$ directly from the projected location on $\mathcal{X}_P$. The time complexity of region projection is $O(1)$ and the time cost of computing $\text{Net}(R_c)$ is therefore eliminated.

Problem transformation. By sharing computation, the *SRS* problem is equivalent to searching feature regions (i.e., $\mathcal{X}_c$) on $\mathcal{X}_P$ that are similar to $\mathcal{X}_q$. We call $\mathcal{X}_P$ the feature space and $\mathcal{X}_c$ a feature region in the rest of this paper. We transform the *SRS* problem to a *Similar Feature Region Search (SFRS)* problem:

Definition 4 (SFRS problem) Given $\mathcal{X}_q$ and $\mathcal{X}_P$, searching a set of N feature regions (denoted as $\mathcal{F}$) on $\mathcal{X}_P$, such that

$$\|g(\mathcal{X}_q) - g(\mathcal{X}_i)\|_2 \leq \|g(\mathcal{X}_q) - g(\mathcal{X}_j)\|_2, \forall \mathcal{X}_i \in \mathcal{F}, \forall \mathcal{X}_j \notin \mathcal{F},$$

After finding the top- N similar feature regions, we can project their locations on $\mathcal{X}_P$ back to the locations of their corresponding regions on P . Hence, the top- N similar regions are found.

5.3.2 Exact Algorithm for *SFRS*

In the *SFRS* problem, the second efficiency challenge still remains, i.e., the number of candidate feature regions is numerous. In this subsection, we propose an efficient search algorithm, namely Exact Similar Feature Region Search (ExactSFRS), that exactly solves the *SFRS* problem.

Representing search space. We apply a method for representing image patches [LBH09] to represent a search space in *SFRS* problem. In particular, the feature space $\mathcal{X}_P \in$

Figure 5.4: An example of search space, which is represented by four intervals $[t_{min}^*, t_{max}^*]$, $[b_{min}^*, b_{max}^*]$, $[l_{min}^*, l_{max}^*]$ and $[r_{min}^*, r_{max}^*]$.

$\mathbb{R}^{W^* \times H^* \times K}$ is a 3-d tensor, where W^* and H^* are its width and height, respectively. A candidate feature region $\mathcal{X}_c \in \mathbb{R}^{w^* \times h^* \times K}$ is a sub-tensor on $\mathcal{X}_P$ that has the same number of feature dimensions (i.e., K) as $\mathcal{X}_P$. Its location is parameterized by a tuple (t^*, b^*, l^*, r^*) , where $1 \leq l^* < r^* \leq H^*$ and $1 \leq b^* < t^* \leq W^*$. By indicating the ranges of the four location parameters, a search space can be represented by four intervals:

Definition 5 (Search space) A set of feature region locations $\mathcal{S} = \bar{t} \times \bar{b} \times \bar{l} \times \bar{r}$, where $\bar{t} = [t_{min}^*, t_{max}^*]$, $\bar{b} = [b_{min}^*, b_{max}^*]$, $\bar{l} = [l_{min}^*, l_{max}^*]$ and $\bar{r} = [r_{min}^*, r_{max}^*]$, such that $\forall (t^*, b^*, l^*, r^*) \in \mathcal{S}$ satisfies $t^* \in \bar{t}$, $b^* \in \bar{b}$, $l^* \in \bar{l}$ and $r^* \in \bar{r}$.

Here, $\times$ denotes the Cartesian product of intervals. Figure 5.4 shows an intuitive example of a search space. In this case, the four intervals form an area (i.e. the shadowed area) between a large feature region $\mathcal{X}_U$ (i.e., $(t_{max}^*, b_{min}^*, l_{min}^*, r_{max}^*)$) and a small feature region $\mathcal{X}_n$ (i.e., $(t_{min}^*, b_{max}^*, l_{max}^*, r_{min}^*)$). This search space $\mathcal{S}$ therefore represents a set of all the feature regions that lie in this area, i.e., between $\mathcal{X}_n$ and $\mathcal{X}_U$ (i.e., $\mathcal{X}_n \subset \mathcal{X}_c \subset \mathcal{X}_U, \forall \mathcal{X}_c \in \mathcal{S}$). The dashed rectangles are two examples of the feature regions in $\mathcal{S}$. Using this representation, the entire search space can be denoted as $\mathcal{S}_v = [1, H^*] \times [1, H^*] \times [1, W^*] \times [1, W^*]$.

Branch and bound search. Instead of exhaustively search in $\mathcal{S}_v$, we use a branch and bound search scheme. The intuition is that the objects on a geographical space are not randomly distributed. Some search subspaces are more probably to contain the optimal results.

To perform a branch and bound search, we first define the distance lower bound of $\mathcal{S}$ given a query $\mathcal{X}_q$ as $\hat{f}(\mathcal{S}|\mathcal{X}_q) \leq \|g(\mathcal{X}_q) - g(\mathcal{X}_c)\|_2, \forall \mathcal{X}_c \in \mathcal{S}$. The function $\hat{f}(\mathcal{S}|\mathcal{X}_q)$ estimates the minimum distance that $\exists \mathcal{X}_c \in \mathcal{S}$ could possibly have compared to $\mathcal{X}_q$. Intuitively, lower $\hat{f}(\mathcal{S}|\mathcal{X}_q)$ means a higher chance that $\exists \mathcal{X}_c \in \mathcal{S}$ is a top- N similar feature region of $\mathcal{X}_q$. To construct $\hat{f}(\mathcal{S}|\mathcal{X}_q)$, we first reexamine the example of search space

in Figure 5.4, where we have $\mathcal{X}_\cap \subset \mathcal{X}_c \subset \mathcal{X}_\cup, \forall \mathcal{X}_c \in \mathcal{S}$. Moreover, because $g(\cdot)$ (i.e., max-pooling) is a monotonically increasing function, we can conclude that $g(\mathcal{X}_\cap) \leq g(\mathcal{X}_c) \leq g(\mathcal{X}_\cup)$, i.e., $\mathbf{v}_\cap[k] \leq \mathbf{v}_c[k] \leq \mathbf{v}_\cup[k], \forall k \in [1, K]$, where $\mathbf{v}[k]$ denotes the k -th dimension of $\mathbf{v}$. We reformulate the distance computation between $\mathcal{X}_c$ and $\mathcal{X}_q$ as $\sqrt{\sum_{k \in [1, K]} (\mathbf{v}_q[k] - \mathbf{v}_c[k])^2}$, where each of the K terms $(\mathbf{v}_q[k] - \mathbf{v}_c[k])^2$ can be bounded using $\mathbf{v}_\cap[k] \leq \mathbf{v}_c[k] \leq \mathbf{v}_\cup[k]$ as:

- (i) $(\mathbf{v}_q[k] - \mathbf{v}_c[k])^2 \geq (\mathbf{v}_q[k] - \mathbf{v}_\cap[k])^2$, if $\mathbf{v}_\cap[k] \geq \mathbf{v}_q[k]$.
- (ii) $(\mathbf{v}_q[k] - \mathbf{v}_c[k])^2 \geq (\mathbf{v}_q[k] - \mathbf{v}_\cup[k])^2$, if $\mathbf{v}_\cup[k] \leq \mathbf{v}_q[k]$.

We denote the k values where $\mathbf{v}_\cap[k] > \mathbf{v}_q[k]$ as k_1 and those satisfy $\mathbf{v}_\cup[k] < \mathbf{v}_q[k]$ as k_2 . Therefore, the distance lower bound is

$$\hat{f}(\mathcal{S}|\mathcal{X}_q) = \sqrt{\sum_{k_1} (\mathbf{v}_q[k_1] - \mathbf{v}_\cap[k_1])^2 + \sum_{k_2} (\mathbf{v}_q[k_2] - \mathbf{v}_\cup[k_2])^2}.$$

Note that when $t_{min}^* < b_{max}^*$ or $r_{min}^* < l_{max}^*$, the feature region $\mathcal{X}_\cap$ does not exist. In this case, we use $\hat{f}(\mathcal{S}|\mathcal{X}_q) = \sqrt{\sum_{k_2} (\mathbf{v}_q[k_2] - \mathbf{v}_\cup[k_2])^2}$ to bound the search space.

We next present the branch and bound search scheme of ExactSFERS in Algorithm 1, which contains the following steps:

- *Initialization* step (line 2-3): We define an empty result set $\mathcal{F}$ and a priority queue Q . We put the initial search space $\mathcal{S}_\forall$ in Q .
- *Branch* step (line 6-7): We retrieve from Q the top search space $\mathcal{S}'$, which has the smallest distance lower bound in the current Q , i.e., $\hat{f}(\mathcal{S}'|\mathcal{X}_q) \leq \hat{f}(\mathcal{S}|\mathcal{X}_q), \forall \mathcal{S} \in Q$. Next, we split $\mathcal{S}'$ into two disjoint subspace $\mathcal{S}_1$ and $\mathcal{S}_2$ by equally dividing the largest interval of its $\bar{t}$, $\bar{b}$, $\bar{l}$ and $\bar{r}$, while the other three keep unchanged.
- *Bound* step (line 8-9): We compute $\hat{f}(\mathcal{S}_1|\mathcal{X}_q)$ and $\hat{f}(\mathcal{S}_2|\mathcal{X}_q)$, and then put them into Q with $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively.

We repeat the branch and bound steps until $|\mathcal{S}'| = 1$, i.e., the top-retrieved space only contains 1 feature region (denoted as $\mathcal{X}'$). This only happens when the intervals of $\mathcal{S}'$ satisfy $t_{max}^* = t_{min}^*$, $b_{max}^* = b_{min}^*$, $l_{max}^* = l_{min}^*$ and $r_{max}^* = r_{min}^*$. Here, we can guarantee $\mathcal{X}'$ is in the top- N similar feature regions of $\mathcal{X}_q$. Because for $\mathcal{S}'$ in this case, we have $\mathcal{X}_\cap = \mathcal{X}_\cup = \mathcal{X}'$ and $k_1 \cup k_2 = [1, K]$, which indicates $\hat{f}(\mathcal{S}'|\mathcal{X}_q) = \|\mathbf{v}_q - \mathbf{v}'\|_2$. Therefore, we can conclude that

$$\|\mathbf{v}_q - \mathbf{v}'\|_2 = \hat{f}(\mathcal{S}'|\mathcal{X}_q) \leq \hat{f}(\mathcal{S}|\mathcal{X}_q) \leq \|\mathbf{v}_q - \mathbf{v}_c\|_2, \forall \mathcal{S} \in Q, \forall \mathcal{X}_c \in \mathcal{S},$$

Algorithm 1: ExactSFRS

Input: Initial search space $\mathcal{S}_V$, query feature region $\mathcal{X}_q$, distance lower bound $\hat{f}(\cdot)$

Output: top- N similar feature regions $\mathcal{F}$

```

1 begin
2    $\mathcal{F} \leftarrow \emptyset; Q \leftarrow \emptyset;$ 
3    $Q.Insert(\mathcal{S}_V);$ 
4   repeat
5     repeat
6        $\mathcal{S}' \leftarrow Q.RetrieveTop();$ 
7        $\text{split } \mathcal{S} \rightarrow \mathcal{S}_1 \cup \mathcal{S}_2;$ 
8        $Q.Insert((\hat{f}(\mathcal{S}_1|\mathcal{X}_q), \mathcal{S}_1));$ 
9        $Q.Insert((\hat{f}(\mathcal{S}_2|\mathcal{X}_q), \mathcal{S}_2));$ 
10      until  $|\mathcal{S}'| = 1;$ 
11       $\mathcal{F} \leftarrow \mathcal{F} \cup \mathcal{S}';$ 
12  until  $|\mathcal{F}| = N;$ 

```

which means $\mathcal{X}'$ has the smallest distance to $\mathcal{X}_q$ than all the other feature regions.

Next, we put $\mathcal{X}'$ into $\mathcal{F}$ and continue the search process until N results are found. The feature regions in $\mathcal{F}$ are the top- N similar feature regions, which can be project back to find the top- N similar regions on P . Note that the method of constructing distance lower bound is also applicable to other $g(\cdot)$ and L_p distance, as long as $g(\cdot)$ is monotonically increasing, such as max-pooling and sum-pooling.

Time complexity analysis. In the branch step, splitting a search space takes $O(1)$ time; In the bound step, we use integral histogram technique [TSJ15, Por05] to compute $\mathbf{v}_\cap$ and $\mathbf{v}_\cup$ and the total complexity of the bound step is $O(K)$. In the worst case, the search process needs to check totally $2|\mathcal{S}_V| - 1$ search spaces and takes $O(\log |\mathcal{S}_V|)$ time to maintain Q , where $|\mathcal{S}_V|$ denotes the total number of candidate feature regions. Therefore, the total complexity is $O(K|\mathcal{S}_V| \log |\mathcal{S}_V|)$. Although the worst-case complexity is very high, ExactSFRS is empirically efficient as shown in the experiments.

5.3.3 Approximate Algorithm for *SFRS*

In an exploratory search environment, the users usually need very fast response and can endure a slightly decrease of the accuracy. For example, social media websites need to efficiently response the queries launched by thousands of users at the same time. Moreover, the queries can be submitted interactively in an exploratory manner. To adapt our model to this kind of scenarios, we propose an approximation method, namely

Table 5.3: Spatial object datasets.

Dataset	SG	NYC	US
object type	POI	POI	Tweet
#objects	244,266	435,210	1,980,943
width (W)	47.7km	61.2km	6078.3km
height (H)	28.7km	48.1km	2608.4km
#attr. dimensions	10	10	50
attr. type	Category	Category	Topic

Approximate Similar Feature Region Search (ApproxSFRS), that can further improve the efficiency by slightly sacrificing the accuracy.

ApproxSFRS uses the same branch and bound search scheme as ExactSFRS, except that we limit the size of Q by a constant M , where $M \leq |\mathcal{S}_V|$. When $Q.size > M$ during the search process, the worst search space (i.e., has the largest $\hat{f}(\mathcal{S}|\mathcal{X}_q)$) in Q will be discarded. To support popping the worst search space out of Q , we re-introduce Q as a min-max heap, whose time complexity of retrieving min/max value and insertion are $O(\log M)$ in our case.

The intuition behind ApproxSFRS is to reduce the unnecessary branch and bound for the bad search spaces. The reason is that branching a bad search space may generate two worse search spaces, who are also unlikely to contain the optimal results. Therefore, we choose to discard those bad search space to prevent wasting computation on its subspaces. **Time complexity analysis.** The time complexity of ApproxSFRS is $O(KM \log M(1 + \log |\mathcal{S}_V| - \log M))$ (see Appendix for the proof¹). M can be varied to adjust the time complexity for different applications. Particularly, when $M = |\mathcal{S}_V|$, the complexity becomes the same as ExactSFRS, i.e., $O(K|\mathcal{S}_V| \log |\mathcal{S}_V|)$; When $M = 1$, the complexity becomes $O(K \log |\mathcal{S}_V|)$, which is equivalent to a greedy binary search in $\mathcal{S}_V$. Therefore, ApproxSFRS is essentially a trade-off between ExactSFRS and a binary search method.

5.4 Experiments

5.4.1 Experimental Setup

Datasets. Our experiments are conducted on three spatial object datasets. Table 6.2 presents the statistics of the datasets. In particular, we collect two POI datasets from Singapore (SG) and New York area (NYC), respectively. The attribute of a POI is its category (e.g., hotel), which is represented as an one-hot vector. As the two datasets are collected from two different countries, the spatial distributions of their POIs are quite

¹http://www.ntu.edu.sg/home/kqzhao/srs_appendix.pdf

different, and thus can show the effectiveness of tested models in different scenarios. To evaluate the efficiency of our search algorithms, we further introduce a large dataset US, which contains around 2 million tweets from United States (except Alaska and Hawaii). We train a Latent Dirichlet Allocation (LDA) [BNJ03] model using these tweets, each of which is associated with a topic distribution vector as its attributes. We partition the regions and the geographical space of SG and NYC into $10\text{m} \times 10\text{m}$ grids. We use $500\text{m} \times 500\text{m}$ grids for US.

Test data generation. To evaluate the effectiveness of the methods, we use a similar method as described in Section 5.2.2 to generate test data. In particular, we randomly sample 2000 regions that contains more than 50 objects as the test queries, denoted as D_q . The height and width of a region vary from 640m to 3km. For an $R_q \in D$, we create a similar region R_+ as the ground truth by applying a small amount of noise and shift to R_q . We create a candidate region database as $D_c = (D_q/R_q) \cup R_+$ (also excluding those have overlap with R_q), considering other regions in D_q as dissimilar regions to R_q . We use R_q to retrieve its similar regions from D_c . Ideally R_+ is ranked at the top. We vary the noise rate and shift distance of generating R_+ in the effectiveness experiments.

To evaluate the efficiency, we reuse the 2000 random sampled R_q to search on P and record the average runtime for each method. To prevent finding too large or too small regions that are usually uninformative to users, we only consider the candidate regions R_c whose width and height are constrained by $0.5w_q \leq w_c \leq 2w_q$ and $0.5h_q \leq h_c \leq 2h_q$.

Baselines. To evaluate the effectiveness of our deep metric learning method (denoted as Triplet), we compare it with the only existing baseline similarity metric, namely SVSM. We also include two heuristic metrics, i.e., SPM and Grid. Details of the baselines are listed as below.

- **SPM** [LSP06]. This method is originally proposed for image categorization. SPM recursively partitions a region into four fine-grained grids and forms a pyramid structure. The l -th pyramid level has $2^{l-1} \times 2^{l-1}$ grids, where $l = 1, 2, 3 \dots$. SPM computes the average grid-wise similarity for each level and sum up all levels with different weights. We use SPM3, SPM4 and SPM5 to represent partitioning a region into a pyramid with 3, 4 and 5 levels. We compare our method with SPM3, SPM4 and SPM5.
- **Grid**. This method is equivalent to comparing a specific level of two region pyramids in SPM. We denote a level with 4×4 grids as Grid4. We use Grid4, Grid8 and Grid16 as baselines.
- **SVSM** [SZH⁺10]. This method defines the four corners and the center of a region as its “reference points”, and compute the average distance of each category of objects to each reference point. To compare two regions, it computes the cosine similarity between their distance vectors.

For the efficiency evaluation, we compare ExactSFRS with the following search methods:

- **Quad** [SZH⁺10]. This is a quadtree based search method developed for SVSM. It builds a quadtree for the geographical space and searches the tree nodes that are similar to the query in a top-down manner. The optimal tree nodes are further expanded greedily as the final similar regions.
- **BruteSFRS**. This is a brute-force method that solves the *SFRS* problem by comparing $\mathcal{X}_q$ with every candidate feature region, i.e., $\forall \mathcal{X}_c \in \mathcal{S}_v$.

Parameter setting. For all the experiments, we use a 3-layer CNN as $Net(\cdot)$. Note that all the layers are convolutional layers with 2×2 stride (instead of using pooling layers). A ReLU non-linearity is applied between two consecutive layers. The filter sizes and feature map dimensions (ordered from input to output) are $\{11, 9, 7\}$ and $\{128, 64, 32\}$, respectively. Note that we use large filters in the convolutional layers, which is helpful for capturing the spatial relations between sparsely distributed objects. For the regularization, we set $\lambda = 0.00005$ and further apply dropout [SHK⁺14] to the output of $Net(\cdot)$. The dropout rate is set to 0.1. The gap parameter for training loss is $\delta = 0.3$. We train our model on 10k generated training tuples. For generating each R_+ , we randomly set the noise rate and the shift rate between 0.1 to 0.3.

Evaluation metrics. We use two metrics to evaluate the effectiveness. (1) *HR@10*. Given the top-10 returned regions $\mathcal{R}$ for a query, *HR@10* is defined as $HR@10 = 1$ if $R_+ \in \mathcal{R}$, $HR@10 = 0$ otherwise. We report the average *HR@10* over all queries in the experiments. (2) *MRR*. *MRR* is a rank-based metric defined as $MRR = \frac{1}{rank}$, where *rank* denotes the rank of R_+ in D_c for a query R_q . We report the average *MRR* over all queries in the experiments.

Experimental environment. All the methods are implemented in Python 3.5. The implementation of Triplet is based on Keras [C⁺15] (v2.0.6) with tensorflow [ABC⁺16] backend. The training can be done within several hours on a single NVIDIA Tesla P100 GPU (16GB memory). All the experiments are conducted on a server with 20-Cores Intel(R) Xeon(R) CPU E5-2680 v2 @2.8GHz and 64GB RAM.

5.4.2 Effectiveness of Triplet

Overall effectiveness. We first study the effectiveness of Triplet. We fix the shift distance as 50m (i.e., shifting the objects along a random direction by 50m) and vary the noise rate from 0.05 to 0.2. Figure 5.5 shows the results of different methods on both SG and NYC datasets with respect to the two metrics. We can see that Triplet is significantly better than the baselines on both datasets in terms of both *HR@10* and

Figure 5.5: Varying noise rate.

Figure 5.6: Varying shift distance.

MRR. For example, when applying only 5% noise, Triplet outperforms SVSM by 180% and 200% on SG and NYC, respectively, in terms of MRR. Grid and SPM perform worst in most of the cases, because they are developed for object categorization in images and cannot adapt to spatial object data very well. Moreover, Grid and SPM perform worse on NYC than on SG. This is because NYC has larger area and more spatial objects. Grid and SPM may easily fail on such massive and noisy data. In contrast, Triplet is very robust to noise. When increasing the noise rate from 0.05 to 0.2 on NYC data, the MRR and HR@10 of Triplet only decrease by 3%–12%, while those values of SVSM decline by 64%–70%.

We next fix the noise rate as 0.05 and vary the shift distance from 50m to 200m. Figure 5.6 shows the results. Triplet still performs the best on different datasets and in terms of both metrics. For example, it outperforms SVSM by more than 55% on NYC in terms of HR@10. We also note that the performance of Triplet decreases when applying larger shift distance. This is because largely shifting objects may change the form of their relative locations, which affects the similarity. The method that does not consider their relative locations do not show this property.

Ablation study. We also study our improvements on triplet network for spatial data, i.e., using ratio-based training loss (denoted as RL) and applying hard negative example

Figure 5.7: Comparing different triplet networks

mining in training data generation (denoted as HM). We also include using large filters in CNN (denoted as LF) as an improvement. Because small filters are usually applied to image data [SZ14]. We denote our triplet network without RL, HM and LF as Triplet/RL, Triplet/HM and Triplet/LF, respectively. Figure 5.7 compares the different versions of Triplet with SVSM on NYC in terms of HR@10. The results on SG and for MRR are similar and thus omitted. We can see that LF and HM are the most important for training triplet network on regions. For example, in Figure 5.7(b), the performance of Triplet/LF and Triplet/HM are even worse than SVSM by more than 10% and 40%, respectively. In addition, the results also show that RL can further improve the effectiveness. On NYC data, the improvement of RL on HR@10 is around 25%–50%.

Case study. To further show the effectiveness of Triplet, we study two typical cases in Singapore and compare the results with SVSM. In particular, we show the query regions (black rectangles) and the corresponding top-5 similar regions (red rectangles) on a map. We further point out the correct similar regions with their names. In the first case, we use a recreation area, i.e., Bukit Timah Nature Reserve, as the query region. Figure 5.8(a) and 5.8(b) show the search results of Triplet and SVSM, respectively. We can see that all the regions found by Triplet are recreation areas, such as Chinese Garden, Sungei Buloh Wetland Reserve and East Coast Park. However, SVSM can only find two regions near Sembawang Park and Pasir Ris Town Park, which is not very accurate. In the second case, we use the most popular shopping & food area in Singapore, i.e., Orchard Road, as the query region. As shown in Figure 5.8(c) and 5.8(d), all the regions returned by both methods contain shopping malls and restaurants. However, Triplet can find two famous shopping & food regions in the downtown area, i.e., Chinatown and Bugis, which are the

(a) Triplet- query: Bukit Timah Nature Reserve (recreation area)

(b) SVSM- query: Bukit Timah Nature Reserve (recreation area)

(c) Triplet- query: Orchard Road (shopping & food area)

(d) SVSM- query: Orchard Road (shopping & food area)

Figure 5.8: Case study – finding top-5 similar regions (black rectangle: query region; red rectangle: found similar region).

most similar to the query but not found by SVSM. These cases show that Triplet can find similar regions more accurately than SVSM.

5.4.3 Efficiency of ExactSFRS and ApproxSFRS

Overall efficiency. Figure 5.9 presents the average runtime of the methods for top-1 similar region search on three datasets. The runtime of BruteSFRS is too long (more than one day per query). Thus, we report its estimated runtime, which is computed by the time cost for checking one candidate region times the total number of candidate regions. Quad runs out of memory on US data and thus we do not show the result. We can see from the figure that ExactSFRS outperforms the two baselines by a significant margin. Particularly, ExactSFRS is over $10^5 \times$ faster than BruteSFRS on all datasets, and more than $130 \times$ and $45 \times$ faster than Quad on SG and NYC, respectively. Moreover, even for a very large dataset like US, the runtime of ExactSFRS is still less than 100

Figure 5.9: Runtime on different datasets.

 Figure 5.10: Runtime of ExactSFRS by varying N .

 Figure 5.11: Trade-off of ApproxSFRS by varying M .

seconds. These show us that ExactSFRS can greatly reduce the search space and is very efficient for similar region search.

Scalability. We demonstrate the scalability of ExactSFRS with respect to N (i.e., top- N similar region search). Figure 5.10 shows the runtime of ExactSFRS when varying N from 1 to 100. We can see that ExactSFRS scales well to N . For example, when N increases from 1 to 100, the runtime on US only increases by 5%.

We further investigate the scalability of ExactSFRS with respect to regions with different numbers of objects. We divide the query regions into five groups based on their numbers of objects: “<100”, “100–500”, “500–1k”, “1k–2k” and “2k+”, each of which indicates the range for the numbers of objects in a group. Table 5.4 compares the average runtime of ExactSFRS and Quad for different query groups on SG. The results show that Quad takes longer to answer for queries that contain more objects. Particularly, it costs

Table 5.4: Average runtime (sec.) for query regions with different numbers of objects on SG.

#obj.	<100	100–500	500–1k	1k–2k	2k<
Quad	1077.5	1281.31	1326.13	1549.77	1606.60
ExactSFRS	20.23	12.50	8.01	7.13	6.85

Table 5.5: Average runtime (sec.) for query regions with different areas (km²) on SG.

Area (km ²)	<2	2–3	3–4	4–5	5<
Quad	1103.89	1098.22	1303.07	1421.82	1893.99
ExactSFRS	8.28	9.36	11.14	11.43	14.32

around 50% more time for the last group of queries (2k<) than for the first group of queries (<100). In contrast, ExactSFRS is more efficient to answer queries that contain more objects. In particular, its runtime for the last group (2k<) drops by over 66% compared with its runtime for the first group (<100). The reason is that more objects can provide more information to characterize the spatial features in a region, which can be leveraged by ExactSFRS to reduce the search space even more effectively.

In addition, we also illustrate the scalability of ExactSFRS w.r.t. different query region sizes. We divide the query regions into five groups based on their area (km²): “<2”, “2–3”, “3–4”, “4–5” and “5<”. Table 5.5 shows the runtime of Quad and ExactSFRS for the five query groups on SG. We observe that both Quad and ExactSFRS scale linearly with respect to the area of query region.

Trade-off between efficiency and accuracy. We vary the maximum queue size M in ApproxSFRS from 1 to $|\mathcal{S}_V|$. Figure 5.11(a) and 5.11(b) present the average runtime and similarity error of ApproxSFRS, respectively, for top-1 similar region search. The similarity error is defined as the difference between the similarity values produced by ExactSFRS and ApproxSFRS. The results show that when M is small, ApproxSFRS is extremely efficient but the error is large. For example, when $M \leq 10$, the similarity difference is larger than 0.2 on SG and NYC. In contrast, when M becomes larger, ApproxSFRS tends to find the accurate results, while the runtime also increases largely. We find that when $M = 10^3$, ApproxSFRS can achieve a good trade-off, where the runtime is reduced by 40.31%, 86.23% and 82.72% on SG, NYC and US, respectively, compared to $M = \mathcal{S}_V$ (i.e., ExactSFRS). Meanwhile, the similarity error is no larger than 0.05. Thus, ApproxSFRS can further improve the search efficiency by slightly sacrificing the accuracy.

5.5 Conclusion

In this chapter, we study a similar region search problem on geographical space. We leverage a deep metric learning method to learn a similarity metric for comparing regions.

The method can capture both the attributes and the relative locations between objects in regions for measuring similarity, and is robust to small noise and shift. To search similar regions efficiently, we propose a branch and bound search algorithm that can greatly reduce the search space. We also develop an approximation method to further improve efficiency by slightly trading off the accuracy. Experiments on three datasets demonstrate our proposed methods outperform baseline methods for similar region search.

Acknowledgements. This research was carried out at the Rapid-Rich Object Search (ROSE) Lab at the Nanyang Technological University, Singapore. The ROSE Lab is supported by the National Research Foundation, Singapore, and the Infocomm Media Development Authority, Singapore. This work is also supported by the MOE Tier-2 grant (MOE2016-T2-1-137) awarded by Ministry of Education Singapore. We thank Prof. Guosheng Lin (Nanyang Technological University) for valuable discussion.

Chapter 6

Modeling Nonstationary Spatiotemporal Activity

Spatiotemporal activity modeling is a fundamental problem of mining geo-tagged post data. It aims at characterizing spatiotemporal dynamics of urban residents via modeling the multimodal attributes (i.e., *spatial*, *temporal* and *textual* attributes) of spatiotemporal activities (i.e., geo-tagged posts). However, most of the existing studies [Siz10, WWXM07, HAG⁺12, KKSS14, YCH⁺11, YCM⁺13b, YCZ⁺15] mainly focus on mining spatial and textual attributes (e.g., geographical topic models) from the activities, missing to consider their *temporal dynamics*. In recent years, several studies [ZZY⁺17a, ZLL⁺18, ZZY⁺17b, ZLL⁺17] utilize multimodal embedding to capture the correlations among the multimodal attributes (i.e., spatial, temporal and textual attributes). Nevertheless, they neglect an important fact that spatiotemporal activities revealed by geo-tagged posts are *temporally non-stationary*. For example, Twitter users are usually more interested in tweeting about breaking news and interesting events, which may vary dramatically across time. Therefore, their methods would easily fuse nonstationary information from different time periods, which result in suboptimal performances.

Motivated by these, we propose a novel solution for modeling nonstationary spatiotemporal activities, which can effectively address the above-mentioned limitations. The basic intuition is that we can consider the nonstationary information behind spatiotemporal activities as a fusion of *static information* and *dynamic information*. The dynamic information represents the information of spatiotemporal activities that varies dramatically across different time periods, while the static information represents those inherent attributes that keep steady across a long timespan. For example, we can consider the nonstationary information of a shopping mall during Christmas as the fusion of its static information (i.e., “shopping”) and dynamic information (“Christmas”), which together reveal the topics like “Christmas sales” or “buying Christmas gifts”. Moreover, we consider that the static information is *transferable* across different periods. Given a

specific time period, it should be aggregated with the dynamic information provided by a limited number of observed activities in the period.

Based on the above intuitions, we formulate the spatiotemporal activity modeling as a *temporal transfer learning* problem and propose a novel end-to-end model, namely Meta Multimodal Embedding (MME) for solving it. Similar to conventional multimodal embedding methods, our method learns rich semantics of regions, hours and keywords. In addition, our MME method explicitly learns *transferable priors* for different geographical regions, which reveal their static information (e.g., region functions and geographical semantics). To fuse both static and dynamic information, we update the transferable priors containing static information with only few activities in a specific time period, which is preferable under data scarcity. Furthermore, we notice that the spatiotemporal activities of a region is usually influenced by nearby regions. To consider such spatial influence, we propose a spatial-aware multi-representational (SAMR) prior to integrate the prior knowledge of a region and its nearby regions, extending MME to a spatial-aware MME (SA-MME) method. Based on the SAMR prior, we propose a spatial-aware Stein Variational Gradient Decent (SVGD) method to optimize SA-MME in an end-to-end manner.

Overall, our contributions can be summarized as follows:

- To the best of our knowledge, we are the first to formulate spatiotemporal activity modeling as a *temporal transfer learning* problem, which allows us to extract static region semantics from nonstationary data and use them for modeling activities for different time periods under data scarcity.
- We propose a novel method based on multimodal embedding, namely Meta Multimodal Embedding (MME), which can learn rich semantics of regions, texts and time units. More importantly, the method is able to extract static information of regions as transferable priors, and solve the scarcity problem by effectively transferring the priors to adapt the dynamic information provided by few observations.
- We further extend MME to a Spatial-Aware MME (SA-MME) model, which considers the spatial influences between nearby regions. We model the spatial-aware prior knowledge of regions using spatial-aware multi-representational prior, and propose a novel spatial-aware Stein Variational Gradient Decent method for an end-to-end training.
- We conduct extensive experiments on two large-scale real-world datasets. The experimental results show that our proposed methods are more effective in spatiotemporal activity modeling with nonstationary geo-textual posts.

This chapter is organized as follows. Section 6.1 defines the problem of spatiotemporal activity modeling. Section 6.2 introduces conventional multimodal embedding technique, and its limitations for modeling nonstationary activities. Section 6.3 presents our transfer learning paradigm, and how to accommodate this paradigm to multimodal embedding. Section 6.4 further extends the method considering spatial influence. Section 6.5 shows the experiments. Section 6.6 summarizes this chapter.

6.1 Problem Description

6.1.1 Spatiotemporal Activity Modeling.

Suppose we have a stream of geo-textual posts $\mathcal{D}$. Each post $e \in \mathcal{D}$ is represented by a tuple (l_e, t_e, W_e) , where l_e is the GPS coordinates of the record, t_e is the timestamp and W_e represents the textual content. The goal of spatiotemporal activity modeling is to learn representations of *spatial*, *temporal* and *textual* attributes (denoted as $\mathbf{r}$, $\mathbf{t}$ and $\mathbf{w}$, respectively) via exploiting geo-textual data (denoted as $\mathcal{D}$), such that the representations reveal the *semantic information* about spatiotemporal activities of those attributes. For example, the representation of a university area (i.e., spatial attributes) should be similar to the representations of keywords (i.e., textual attributes) “exam”, “study”, “education”, etc. More importantly, the representations should be informative for being applied to downstream applications, such as answering the following queries [ZZY⁺17b]: 1) Spatial query: What are the activities that are most likely to appear in a given query region? and 2) Activity query: Given a query activity (represented by keywords), where is it most likely to appear?

6.1.2 Period-specific Spatiotemporal Activity Modeling.

As the spatiotemporal activity is temporally nonstationary, we build *period-specific* spatiotemporal activity models, such that we can more comprehensively understand different information in different time periods. For example, if we consider each day as a period, we can more discriminately model the activities in different workdays and the weekend, which provides fine-grained information for downstream applications.

To be more specific, we first denote the entire timespan of the geo-textual data as $[t_{min}, t_{max}]$, where t_{min} and t_{max} represent the earliest and the latest timestamps of the posts in the data, respectively, i.e., $\forall e \in \mathcal{D}, t_e \in [t_{min}, t_{max}]$. A *time period* τ can therefore be defined as a small time interval (denoted as $\tau = [t_{min}^\tau, t_{max}^\tau]$) of the entire timespan, i.e., $\tau \subset [t_{min}, t_{max}]$. We denote the data instances recorded during τ as $\mathcal{D}^\tau$, where $\forall e \in \mathcal{D}^\tau, t_e \in [t_{min}^\tau, t_{max}^\tau]$. Moreover, we introduce *period-specific representation* $\mathbf{r}^\tau$ for the spatial attributes appear in a specific time period τ . Note that in our study, we only

consider the nonstationarity for spatial attributes, i.e., temporal and textual attributes are still denoted using global representations $\mathbf{t}$ and $\mathbf{w}$ (i.e., shared across different time periods). Modeling nonstationary temporal and textual attributes is left as future work. Finally, we formulate the problem of modeling period-specific spatiotemporal activities as

Definition 6 (Period-specific Spatiotemporal Activity Modeling) *Given a specific time period τ associated with the geo-textual data $\mathcal{D}_\tau$, learning period-specific representations of spatial attributes $\mathbf{r}_\tau$, such that $\mathbf{r}_\tau$ reveals the semantic information about spatiotemporal activities during a specific time periods τ .*

6.2 Preliminary

6.2.1 Multimodal Embedding

Multimodal embedding (ME) has become a popular technique and achieves the state-of-the-art performance for spatiotemporal activity modeling [ZZY⁺17a, ZLL⁺18, ZZY⁺17b, ZLL⁺17]. Multimodal embedding aims at learning representations for *regions*, *hours of a day* and *keywords*, which represents the spatial, temporal and textual attributes, respectively. The learned representations should capture the semantic correlations between these entities (i.e., regions, hours and keywords), e.g., university regions should be highly correlated to “exam” and “study”.

In particular, each entity is associated with a latent embedding vector as its representation, which encodes its semantic meanings. For a given geo-textual post $e = (l_e, t_e, W_e)$, we can represent its attributes as a set of entity embeddings as $\mathcal{V}_e = \{\mathbf{r}_e\} \cup \{\mathbf{t}_e\} \cup \mathcal{W}_e$, where $\mathbf{r}_e \in \mathbb{R}^d$, $\mathbf{t}_e \in \mathbb{R}^d$ and $\mathcal{W}_e = \{\mathbf{w}_{e1}, \mathbf{w}_{e2}, \dots, \mathbf{w}_{en} | \mathbf{w}_{ei} \in \mathbb{R}^d, i = 1, 2, \dots, n\}$ stands for the region representation, hour representation and the set of keyword representations of e , respectively. To learn the representations, it follows the intuition of word2vec model [MSC⁺13] that utilizes the co-occurrence relationships between entities. The objective of training the entity embeddings is to maximize a reconstruction probability of given each entity (i.e., region, hour or keyword) in the geo-textual post as

$$p(\mathbf{v} | \mathcal{V}_e^{-\mathbf{v}}) = \frac{\exp(s(\mathbf{v}, \mathcal{V}_e))}{\sum_{\mathbf{u}} \exp(s(\mathbf{u}, \mathcal{V}_e))}, \quad (6.1)$$

where $\mathbf{v} \in \mathcal{V}_e$ is the target entity embedding; $\mathcal{V}_e^{-\mathbf{v}}$ is the set of entity embeddings of p except $\mathbf{v}$, and $\mathbf{u}$ represents all entity embeddings that have the same type of $\mathbf{v}$. Here, $s(\cdot)$ denotes to a similarity metric, which is commonly implemented as dot-product, i.e., $s(\mathbf{v}, \mathcal{V}_e^{-\mathbf{v}}) = \mathbf{v}^\top \sum_{\mathbf{v}' \in \mathcal{V}_e^{-\mathbf{v}}} \mathbf{v}' / |\mathcal{V}_e^{-\mathbf{v}}|$, where $|\mathcal{V}_e^{-\mathbf{v}}|$ is the number of embeddings in the set.

To train the model more efficiently, negative sampling is often used. The loss function of ME is formulated as

$$\mathcal{L}_e = -\log \sigma(s(\mathbf{v}, \mathcal{V}_e^{-\mathbf{v}})) - \sum_{k=1}^K \log \sigma(-s(\mathbf{u}, \mathcal{V}_e^{-\mathbf{v}})), \quad (6.2)$$

where $\mathbf{u}$ is sampled from the same type of entities as $\mathbf{v}$, but does not appear together with $\mathcal{V}_e^{-\mathbf{v}}$; The number of negative samples is denoted by K . The total loss is therefore given by $\mathcal{L}_{\mathcal{D}} = \sum_{e \in \mathcal{D}} \mathcal{L}_e$.

6.2.2 Limitation and Challenge of Multimodal Embedding

Conventional multimodal embedding [ZZY⁺17a] can hardly be applied to period-specific spatiotemporal activity modeling. It learns *global representations* of the entities using the mixture of data from different time periods. It can hardly be applied to a specific time period, as it might bring noisy information from other time periods. This would largely limit the capacity of spatiotemporal activity modeling for real-time applications. Thus, it is more desirable to learn *period-specific representations*, which can help us understand the activities during a specific time period.

The main challenge of learning period-specific representations is the lack of data collected during a specific period. The solution to alleviate such data scarcity is to leverage other information and transfer to the specific period. Existing studies try to 1) heuristically subsample the data in recent time periods to augment the training data for the target period [ZZY⁺17b], and 2) utilize external information from other data sources [ZLL⁺18]. However, they have two main drawbacks — 1) the heuristic method introduces extra parameters that are hard to be determined and may not work well on various situations; 2) the external information from other data sources requires extra efforts on data collection and processing, which limit its application for real-world scenarios.

6.3 Period-Specific Modeling via Temporal Transfer Learning

In this section, we propose a novel solution for period-specific spatiotemporal activity modeling, which is based on multimodal embedding. Our method overcomes the limitations of conventional multimodal embedding on modeling nonstationary spatiotemporal activities (as indicated in Section 6.2.2). In particular, we formulate this problem as a *temporal transfer leaning* problem, and propose a Meta Multimodal Embedding (MME) method that directly learns how to transfer knowledge for period-specific activity modeling. Table 6.1 lists the frequently-used notations in the proposed method.

Table 6.1: Description of frequently-used notations.

Symbol	Description
τ	a time period
$\mathbf{r}_0$	the region prior embedding
$\mathbf{r}_\tau$	the region embedding for time period τ
$\mathbf{t}$	the embedding of an hour
$\mathbf{w}$	the embedding of a keyword
$\mathcal{D}_\tau$	the data of period τ
$\mathcal{D}_\tau^s$	the support data of period τ
$\mathcal{D}_\tau^q$	the query data of period τ
$f(\cdot)$	the updating function of region representation

6.3.1 Transferable Priors of Regions

We first introduce a *transferable prior* for each region (denoted as $\mathbf{r}_0$). The transferable prior represents the static information (e.g., region function, geographical semantics), which keeps invariant across different time periods. Therefore, we can define period-specific region representation $\mathbf{r}_\tau$ as a fusion of the static information provided by $\mathbf{r}_0$ and the period-specific dynamic information provided by $\mathcal{D}_\tau$, i.e.,

$$\mathbf{r}_\tau = f(\mathbf{r}_0, \mathcal{D}_\tau), \quad (6.3)$$

where f denotes the updating function that aggregates these two types of information. From the view of knowledge transfer, the updating function transfers the prior representation to the period-specific representation by aggregating the activities from the period.

However, the transferable prior $\mathbf{r}_0$ is difficult to be determined. A straightforward method is to use the region representations trained on the whole dataset $\mathcal{D}$ as $\mathbf{r}_0$. Nevertheless, such method may mix dynamic information from multiple time periods in the prior, making it difficult to be used for modeling period-specific activities. Also, it is also unknown whether such representations can be transferred to τ , where the data is very scarce.

To solve these issues, we propose to directly learn $\mathbf{r}_0$ from data. Particularly, we first assume some downstream applications may query for the activities underlying a set of data, which is named as the *query set* (denoted as $\mathcal{D}_\tau^q$). To avoid confusion, we re-denote the training dataset for modeling $\mathbf{r}_\tau$ as *support set* $\mathcal{D}_\tau^s$, i.e., $\mathbf{r}_\tau = f(\mathbf{r}_0, \mathcal{D}_\tau^s)$. Note that $\mathcal{D}_\tau^q \not\subseteq \mathcal{D}_\tau^s$, since the queries may not appear in $\mathcal{D}_\tau^s$. Given both the support set $\mathcal{D}_\tau^s$ and the query set $\mathcal{D}_\tau^q$, our goal of learning period-specific region representations can be interpreted as: Given a region whose transferable prior is $\mathbf{r}_0$, deriving $\mathbf{r}_\tau$ with the support data as $\mathbf{r}_\tau = f(\mathbf{r}_0, \mathcal{D}_\tau^s)$, such that the period-specific region representation $\mathbf{r}_\tau$ also

Figure 6.1: Meta Multimodal Embedding for period-specific spatiotemporal activity modeling.

contain the information of $\mathcal{D}_\tau^q$. Formally, the objective is to maximize the probability of capturing the information in query set, i.e.,

$$\arg \max_{\mathbf{r}_0 \in \mathcal{R}, \mathbf{t} \in \mathcal{H}, \mathbf{w} \in \mathcal{W}} \mathbb{E}_{\tau \sim p(\tau)} \left[p(\mathcal{D}_\tau^q | \mathbf{r}_\tau, \mathbf{t}, \mathbf{w}) \right] \quad \text{where} \quad \mathbf{r}_\tau = f(\mathbf{r}_0, \mathcal{D}_\tau^s), \quad (6.4)$$

where $\mathcal{R}$, $\mathcal{H}$ and $\mathcal{W}$ represent the embeddings of all the regions, hours and keywords in the data, respectively. Here, $p(\tau)$ represents the distribution of sampling period, e.g., uniform distribution across the entire timespan.

6.3.2 Meta Multimodal Embedding

We propose a meta-learning based multimodal embedding method, namely Meta Multimodal Embedding (MME), which directly optimizes Eq. (6.4) for period-specific activity modeling. The method is inspired by Model-Agnostic Meta Learning (MAML) [FAL17].

To optimize Eq. (6.4), we need to carefully design

- the construction of $\mathcal{D}_\tau^s$ and $\mathcal{D}_\tau^q$,
- the updating function $f(\cdot)$, and
- the probability $p(\mathcal{D}_\tau^q | \mathbf{r}_\tau, \mathbf{t}, \mathbf{w})$.

First, we consider how to construct the pair of support and query datasets $(\mathcal{D}_\tau^s, \mathcal{D}_\tau^q)$ as the input of the optimization for each time period τ . As the real $\mathcal{D}_\tau^q$ issued by the downstream applications are unknown during training, we leverage a simple sampling strategy to construct a pseudo query set for a particular time period τ . For a particular period τ , we randomly sample two subsets of geo-textual posts from the data collected during τ (denoted as $\mathcal{D}_\tau$) as $\mathcal{D}_\tau^s$ and $\mathcal{D}_\tau^q$, i.e., $\mathcal{D}_\tau^s \subset \mathcal{D}_\tau$ and $\mathcal{D}_\tau^q \subset \mathcal{D}_\tau$. More importantly, to handle the data scarcity problem, we explicitly limit the size of $\mathcal{D}_\tau^s$ at a very small value, i.e., $|\mathcal{D}_\tau^s| \ll |\mathcal{D}_\tau|$, such that optimizing Eq. 6.4 can also learn how to generalize $\mathbf{r}_\tau$ to $\mathcal{D}_\tau^q$ even when $\mathcal{D}_\tau^s$ is scarce.

Next, we implement the updating function $f(\cdot)$ that transfers the region prior $\mathbf{r}_0$ to period-specific embedding $\mathbf{r}_\tau$ by incorporating $\mathcal{D}_\tau^s$. A straightforward choice is to use a single-step stochastic gradient decent w.r.t. multimodal embedding loss as

$$\mathbf{r}_\tau = f(\mathbf{r}_0, \mathcal{D}_\tau^s) = \mathbf{r}_0 - \epsilon \nabla_{\mathbf{r}_0} \mathcal{L}_{\mathcal{D}_\tau^s}(\mathbf{r}_0, \mathbf{t}, \mathbf{w}) \quad (6.5)$$

where

$$\mathcal{L}_{\mathcal{D}_\tau^s}(\mathbf{r}_0, \mathbf{t}, \mathbf{w}) = \sum_{e_\tau \in \mathcal{D}_\tau^s} \mathcal{L}_{e_\tau}(\mathbf{r}_0, \mathbf{t}, \mathbf{w}). \quad (6.6)$$

In Eq (6.5), ϵ is the updating step-size and $\mathcal{L}_{e_\tau}(\mathbf{r}_0, \mathbf{t}, \mathbf{w})$ is the loss function of conventional multimodal embedding as shown in Eq. 6.2. Thus, $\mathbf{r}_\tau$ aggregates the semantic information provided by $\mathcal{D}_\tau^s$ with $\mathbf{r}_0$. Note that other updating rules such as multi-step stochastic gradient are also applicable.

We now move on to implement $p(\mathcal{D}_\tau^q | \mathbf{r}_\tau, \mathbf{t}, \mathbf{w})$, which evaluates $\mathbf{r}_\tau$ and provides gradients that backpropagate to optimize $\mathbf{r}_0$, $\mathbf{t}$ and $\mathbf{w}$. Intuitively, $\mathbf{r}_\tau$ should capture the semantic information of $\mathcal{D}_\tau^q$. Therefore, we employ the reconstruction objective of multimodal embedding that follows this intuition, i.e.,

$$p(\mathcal{D}_\tau^q | \mathbf{r}_\tau, \mathbf{t}, \mathbf{w}) = \prod_{e \in \mathcal{D}_\tau^q} p(\mathbf{v} | \mathcal{V}_e^{-\mathbf{v}}), \quad (6.7)$$

where $p(\mathbf{v} | \mathcal{V}_e^{-\mathbf{v}})$ is as defined in Eq. (6.1). Empirically, this probability can be maximized via minimizing the loss $\mathcal{L}_{\mathcal{D}_\tau^q}(\mathbf{r}_\tau, \mathbf{t}, \mathbf{w})$, where

$$\mathcal{L}_{\mathcal{D}_\tau^q}(\mathbf{r}_\tau, \mathbf{t}, \mathbf{w}) = \sum_{e_\tau \in \mathcal{D}_\tau^q} \mathcal{L}_{e_\tau}(\mathbf{r}_\tau, \mathbf{t}, \mathbf{w}). \quad (6.8)$$

After training the MME model, we derive the period-specific region representation $\mathbf{r}_\tau$ for time period τ using Eq. (6.5).

6.4 Spatial-aware Prior Estimator

MME explicitly learns static information as a prior for each region, which are transferable for modeling across different periods. However, both the prior and period-specific modeling only contain the information *within* the target region, while missing to consider *spatial influence* from nearby areas. According to the Tobler’s law of geography [Mil04], nearby things should be more related to distant things. That is, the activities from nearby regions should have strong impact to the activities within the target region. For example, two shopping areas may have similar regional function (i.e., shopping), while area A is surrounded by universities and area B locates close to a sight-seeing landmark. When conducting promotions, area A may have discount on stationary items, while area B may sell souvenirs. In addition, the spatial influence can also provide extra information for a region to handle the data scarcity problem.

To this end, we propose to model region priors with consideration of spatial influence between regions. In particular, we extend the *single-vector region priors* to *prior estimators* that constructed by multiple prior vectors, which can effectively model the uncertainties brought by the influence of nearby regions. As a result, we can better handle the spatial influence when modeling period-specific activities, which improves the effectiveness of MME model.

Overview. To incorporate the information from nearby regions into the priors, we leverage Kernel Density Estimator (KDE), which allows us to leverage the multiple prior representations of nearby regions to estimate the prior of the target region. Moreover, we propose a Spatial-Aware Stein Variational Gradient Descent (SA-SVGD) that enables an end-to-end optimization of the model.

6.4.1 Incorporating Spatial-aware Prior Estimator for MME

Extending $\mathbf{r}_0$ to $p(\mathbf{r}|\Theta_0)$. We first define prior estimator of a region as $p(\mathbf{r}|\Theta_0)$, where Θ_0 represents the parameters of the estimator. Note that similar to $\mathbf{r}_0$, different regions would have different $p(\mathbf{r}|\Theta_0)$ (i.e., different Θ_0). Subsequently, we can assume that the period-specific region representation $\mathbf{r}_\tau$ is sampled from the posterior estimator as

$$\mathbf{r}_\tau \sim p(\mathbf{r}|\Theta_\tau) := p(\mathbf{r}|\Theta_0, \mathcal{D}_\tau^s). \quad (6.9)$$

where Θ_τ represents the parameters of the posterior estimator, which is updated from Θ_0 when observing support dataset $\mathcal{D}_\tau^s$. By applying Monte Carlo approximation, we can extend the objective shown in Eq. (6.4) as

$$\arg \max_{\Theta_0, \mathbf{t} \in \mathcal{H}, \mathbf{w} \in \mathcal{W}} \mathbb{E}_{\tau \sim p(\tau)} \left[\frac{1}{M} \sum_{m=1}^M p(\mathcal{D}_\tau^q | \mathbf{r}_{\tau,m}, \mathbf{t}, \mathbf{w}) \right] \quad (6.10)$$

where $\mathbf{r}_{\tau,m} \sim p(\mathbf{r}|\Theta_0, \mathcal{D}_\tau^s)$.

Figure 6.2: Construction of prior estimator using nearby regions (red square: target region; blue squares: nearby regions weight by distance).

Here, $\mathbf{r}_{\tau,m}$ represents the m -th sampled vector from the posterior estimator $p(\mathbf{r}|\mathbf{r}_0, \mathcal{D}_\tau^s)$ and M denotes to the total number of Monte Carlo samples.

Implementing $p(\mathbf{r}|\Theta_0)$ using kernel density estimator. Next, we implement $p(\mathbf{r}|\Theta_0)$, which should 1) be trained with MME in an end-to-end manner and 2) incorporate the spatial influence of nearby regions.

To achieve the first goal, we employ a *kernel density estimator (KDE)* for modeling $p(\mathbf{r}|\Theta_0)$, which is inspired by previous studies [YKD⁺18, LW16]. The motivation of using KDE to construct $p(\mathbf{r}|\Theta_0)$ is because it is more flexible for modeling complicated uncertainties.

In particular, we consider Θ_0 as a set of *particle vectors* i.e., $\Theta_0 = \{\theta_0^{(1)}, \theta_0^{(2)}, \dots, \theta_0^{(C)}\}$, where we use $\theta_0^{(c)}$ to represent each of the particle vector and the total particle number is C . The prior estimator can formally be defined as

$$p(\mathbf{r}|\Theta_0) = \frac{1}{C} \sum_{c=1}^C \mathcal{K}(\mathbf{r}, \theta_0^{(c)}), \quad \theta_0^{(c)} \in \Theta_0, \quad (6.11)$$

where $\mathcal{K}_h(\cdot, \cdot)$ is a positive-definite kernel.

To achieve the second goal, i.e., incorporating spatial influence, we propose to reuse the prior vectors $\mathbf{r}_0$ to construct a *spatial-aware kernel density estimator*. Specifically, we explicitly define Θ_0 of a target region (denoted as r^i) as a set of embeddings that contain 1) the prior embedding of the target r^i (denoted as $\mathbf{r}_0^i$) and 2) the embeddings of nearby

regions (denoted as $\mathbf{r}_0^j \in \mathcal{N}(i)$), i.e., $\Theta_0 = \{\mathbf{r}_0^i\} \cup \{\mathbf{r}_0^j | \forall r^j \in \mathcal{N}(i)\}$. The nearby regions r^j are sampled from a local neighborhood of r^i on the geographical space. Furthermore, we weight the spatial influence of each nearby region based on its distance to the target region. Therefore, we can rewrite Eq. (6.11) (i.e., the prior estimator of r^i) as

$$p(\mathbf{r}|\Theta_0) = \frac{1}{C} \left[\mathcal{K}(\mathbf{r}, \mathbf{r}_0^i) + \sum_{\mathbf{r}_0^j \in \mathcal{N}(i)} w(i, j) \times \mathcal{K}(\mathbf{r}, \mathbf{r}_0^j) \right], \quad (6.12)$$

where $w(i, j) = \frac{1}{1+dis(i, j)}$ and $dis(i, j)$ represents the geographical distance between region r^i and r^j . Figure 6.2 illustrates the construction of the prior estimator. As shown in Eq. (6.12), the spatial-aware prior estimator consists of two parts. The first term represents the static information of r_i itself, and the second term represents the aggregated spatial influence from the nearby regions. To this end, we are able to model the complex spatial influence of nearby region in the prior knowledge.

6.4.2 Posterior Inference via Spatial-Aware Stein Variational Gradient Decent

Similar to using $f(\cdot)$ to update $\mathbf{r}_0$ in MME, we also need to update the prior estimator $p(\mathbf{r}|\Theta_0)$ to a posterior estimator $p(\mathbf{r}|\Theta_\tau)$ (i.e., with period-specific knowledge) when giving the support data $\mathcal{D}_\tau^s$. To achieve this, we resort to leverage Stein Variational Gradient Decent (SVGD) method [LW16]. In the following, we first introduce conventional SVGD method and then describe how we use it to optimize the spatial-aware kernel density estimator as shown in Eq. (6.12).

Stein Variational Gradient Decent. For the prior KDE shown in Eq. (6.11), SVGD considers its posterior inference as updating each of the particle vectors (i.e., $\Theta_0 = \{\theta_0^{(1)}, \theta_0^{(2)}, \dots, \theta_0^{(C)}\}$) as

$$\theta_\tau^{(c)} = \theta_0^{(c)} - \epsilon^c \phi(\theta_0^{(c)}), \quad c = 1, 2, \dots, C, \quad (6.13)$$

where ϵ^c is the updating step-size for $\theta_0^{(c)}$. Here, we use $\phi(\theta_0^{(c)})$ to represent the functional gradients w.r.t. the c -th particle $\theta_0^{(c)}$, which is defined as

$$\phi(\theta_0^{(c)}) = \frac{1}{C} \sum_{c'=1}^C [\mathcal{K}(\theta_0^{(c')}, \theta_0^{(c)}) \times \eta(\theta_0^{(c')}) + \nabla_{\theta_0^{(c')}} \mathcal{K}(\theta_0^{(c')}, \theta_0^{(c)})], \quad (6.14)$$

where $\eta(\theta_0^{(c')})$ represents the gradients of the particle $\theta_0^{(c')}$ w.r.t. to a loss function $\mathcal{L}$, i.e.,

$$\eta(\theta_0^{(c')}) = \nabla_{\theta_0^{(c')}} \mathcal{L}(\theta_0^{(c')}). \quad (6.15)$$

Spatial-Aware SVGD. By integrating the spatial-aware kernel density estimator into SVGD method, we develop a spatial-aware SVGD method, namely SA-SVGd that performs posterior inference for Eq. (6.12), i.e., updating $\Theta_0 = \{\mathbf{r}_0^i\} \cup \{\mathbf{r}_0^j | \forall r^j \in \mathcal{N}(i)\}$.

Next, we elaborate the posterior inference procedure using $\mathbf{r}_0^i$ as an example, and $\mathbf{r}_0^j$ can be updated in the same way. In particular, we update $\mathbf{r}_0^i$ as

$$\mathbf{r}_\tau^i = \mathbf{r}_0^i - \epsilon^i \phi(\mathbf{r}_0^i), \quad (6.16)$$

where ϵ^i is the updating step-size for $\mathbf{r}_0^i$. The functional gradients $\phi(\mathbf{r}_0^i)$ is defined as

$$\phi(\mathbf{r}_0^i) = \frac{1}{C} \left[\alpha \phi_1(\mathbf{r}_0^i) + \beta \phi_2(\mathbf{r}_0^i) + \gamma \phi_3(\mathbf{r}_0^i) \right] \quad (6.17)$$

where α , β and γ are hyperparameters to weight the three functional gradient terms, which are defined as

$$\phi_1(\mathbf{r}_0^i) = \mathcal{K}(\mathbf{r}_0^i, \mathbf{r}_0^i) \times \eta(\mathbf{r}_0^i) \quad (6.18)$$

$$\phi_2(\mathbf{r}_0^i) = \sum_{\mathbf{r}_0^j \in \mathcal{N}(i)} w(i, j) \times \mathcal{K}(\mathbf{r}_0^i, \mathbf{r}_0^j) \times \eta(\mathbf{r}_0^j) \quad (6.19)$$

$$\phi_3(\mathbf{r}_0^i) = \sum_{\mathbf{r}_0^j \in \mathcal{N}(i)} w(i, j) \times \nabla_{\mathbf{r}_0^j} \mathcal{K}(\mathbf{r}_0^i, \mathbf{r}_0^j). \quad (6.20)$$

The gradient $\eta(\mathbf{r}_0^i)$ w.r.t. to the loss function on $\mathcal{D}_\tau^s$ is defined as,

$$\eta(\mathbf{r}_0^i) = \nabla_{\mathbf{r}_0^i} \mathcal{L}_{\mathcal{D}_\tau^s}(\mathbf{r}_0^i, \mathbf{t}, \mathbf{w}). \quad (6.21)$$

Interpretation of SA-SVGd. Intuitively, each of the functional gradient terms in Eq. (6.17) can be understood as follows:

- $\phi_1(\mathbf{r}_0^i)$ represents the gradients of r^i itself w.r.t. the support data,
- $\phi_2(\mathbf{r}_0^i)$ represents the gradients consulted from the nearby regions r^j , which can be considered as the spatial influence in period-specific region modeling.
- $\phi_3(\mathbf{r}_0^i)$ represents the repulsive force between r^i and r^j , which forces r^j to have different semantics to r^i , such that the spatial influence would be more complementary to r_i for its modeling.

By using SA-SVGd, we are able to infer the posterior estimator $p(\mathbf{r}|\Theta_\tau)$ considering spatial influence between nearby regions. After that, we can sample $\mathbf{r}_\tau$ from $p(\mathbf{r}|\Theta_\tau)$ and follow the same paradigm as MME to optimize Eq. (6.10).

6.5 Experiments

6.5.1 Experimental Setup

Datasets & Preprocessing. Our experiments are conducted on two real-world tweet datasets collected from New York City (NYC) and Los Angeles (LA). The geo-textual posts in both datasets are recorded during 2014-06-20 to 2015-12-31. We partition the geographical space into $300\text{m} \times 300\text{m}$ grids and consider each grid cell as a region. We filtered those tweet words that appear less than 500 times in each of the datasets. We finalize the two datasets NYC and LA with totally 9,048,630 and 6,228,474 tweets, respectively. Table II presents the statistics of the preprocessed datasets. As the spatiotemporal activities are very different between NYC and LA, evaluation on the two datasets can inspect the applicability of different modeling methods in different cities.

For the experimental evaluation, we partition the entire timespan into different days as different time periods (totally 559 days). We use the earliest 80% days as training periods and the rest as the test periods for each dataset, i.e., we evaluate our methods on modeling the activities in test time periods.

Baselines. We compare the following state-of-the-art methods, including topic model, factorization method and multi-modal embedding methods, in our experiments:

- **LGTA** [YCH⁺11] is a topic model that models region representation as a multinomial distribution over latent topics learned from the data.
- **SVD** [H⁺70] builds a 3-D tensor to represent the co-occurrence among regions, hours and keywords, whose representations are learned via Singular Value Decomposition.
- **CROSSMAP** [ZZY⁺17a] is a conventional multimodal embedding method that learns low-dimensional embeddings of regions, hours and keywords.
- **REACT** [ZZY⁺17b] is the state-of-the-art method for spatiotemporal activity modeling. It enhances conventional multimodal embedding with a recency-aware learning scheme for period-specific activity modeling.
- **MME** is our proposed method in Section 6.3, which learn transferable priors of regions for period-specific activity modeling.
- **SA-MME** extends the transferable prior vectors of regions in MME to prior estimators that considers underlying spatial influence between nearby regions.

Note that for all the methods, we use the same scheme for partitioning the regions (i.e., as grid cells) and use hours and keywords to represent the temporal and textual attributes.

Implementation details. For MME method, we set the latent dimensionality d of all the embedding vectors (i.e., $\mathbf{r}_0$, $\mathbf{r}_\tau$, $\mathbf{t}$ and $\mathbf{w}$) as 100. We set the updating step size as 0.1. For training MME, we first sample a period from all training periods, and then draw 8 pairs of support-query datasets from the sampled period. Each sampled support/query dataset contains 8 tweets. We set the learning rate of training all the parameters as 0.005 and the number of negative samples as 5. For the construction of region prior estimators of SA-MME, we leverage a cosine function, i.e., $\mathcal{K}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}^\top \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$, as the kernel function. For each target region, we sample 5 nearby regions (within a range of 5×5 cells) to construct the KDE shown in Eq. (6.12). For the parameters of SA-SVGD, we set $\alpha = 0.8$, $\beta = 0.1$ and $\gamma = 0.1$.

Evaluation paradigm. We evaluate our proposed methods on a region prediction task, which aims at predicting the region where a given query activity (i.e., keywords) is recorded. Note that a timestamp is also given as a part of the query for this task.

To evaluate period-specific activity modeling, we separately perform the region prediction task in different time periods. Taking MME as an example, for each test time period τ , we first sample 8 tweets from $\mathcal{D}_\tau$ as the support data for each region to get the period-specific region representation $\mathbf{r}_\tau$. For those regions that do not have any data during τ , we simply use their priors $\mathbf{r}_0$ as their period-specific representations. After that, we use the remaining tweets in τ as queries by masking their locations. Similarly for SA-MME, we sample 8 tweets for each region as support data for period-specific activity modeling, i.e., computing $p(\mathbf{r}|\Theta_\tau)$. For other methods that do not consider period-specific activity modeling, we finetune their static region representations with the sampled support data as their period-specific representations.

For each query, we compute cosine similarity between the embedding vector of a candidate region (i.e., $\mathbf{r}_\tau$) and each of the query keywords $\mathbf{w}_q \in q$, where q represents the query (i.e., the set of query keywords). Therefore, the prediction score of q belonging to the candidate region is defined as $\frac{1}{|q|} \sum_{\mathbf{w} \in q} \frac{\mathbf{w}^\top \mathbf{r}_\tau}{\|\mathbf{w}\| \|\mathbf{r}_\tau\|}$, where $|q|$ represents the number of keywords contained in the query q .

Evaluation metric. We use Mean Reciprocal Rank (MRR) as the metric to evaluate the prediction results in each time period. In particular, we formulate the region prediction task as a ranking problem. Specifically for each query q , we use the masked location of its original tweet as the ground-truth, and randomly sample N regions as negative candidates. We expect the ground-truth region to be ranked higher than the negative candidates.

We define the MRR value of each query q as $MRR_q = \frac{1}{1 + rank_+}$, where $rank_+$ denotes the rank of the ground-truth in the ranking list. The MRR value of a specific period is averaged on all the queries of the period, i.e., $MRR_\tau = \frac{1}{|Q_\tau|} \sum_{q \in Q_\tau} MRR_q$, where Q_τ denotes the set of queries in τ . We report the MRR value averaged on all the periods in our experiments, i.e., $MRR_\tau = \frac{1}{|\mathcal{T}_{test}|} \sum_{\tau \in \mathcal{T}_{test}} MRR_\tau$, where $\mathcal{T}_{test}$ represents the set of all the test periods.

Table 6.2: Tweet datasets.

Dataset	#tweets	#keywords	#periods
NYC	9,048,630	7,527	559 days
LA	6,228,474	4,449	559 days

Table 6.3: Performance of different methods for region prediction (metric: MRR).

Method	N=10		N=100	
	NYC	LA	NYC	LA
LGTA	0.3558	0.2242	0.0963	0.0653
SVD	0.3577	0.2103	0.0958	0.0548
CROSSMAP	0.4982	0.3917	0.1565	0.1083
REACT	0.5330	0.4296	0.1836	0.1195
MME	0.5893	0.4513	0.2172	0.1371
SA-MME	0.6349	0.5120	0.2612	0.2094

6.5.2 Quantitative Experiments

Figure 6.3 shows the performances of different methods for region prediction on the two datasets. Here, we set the number of negative samples for each query as $N = 10$ and $N = 100$. We can see from the results that 1) LGTA and SVD do not have very good performances, as both of them do not explicitly capture the correlations among spatial, temporal and textual attributes of spatiotemporal activities, where LGTA misses to consider temporal attributes and SVD is originally designed for filling missing values in the co-occurrence tensor; 2) CROSSMAP has better results than LGTA and SVD and thus is more effective in modeling spatiotemporal activities. For example, the relative improvements of CROSSMAP over LGTA and SVD are more than 40% and 70% on NYC and LA data, respectively, when setting $N = 10$; 3) MME outperforms conventional multimodal embedding methods (i.e., CROSSMAP and REACT) by a large margin. For example, when $N = 10$, MME is better than CROSSMAP and REACT by 18.3% and 10.6% on NYC data, respectively. This shows us that our transfer learning paradigm can better alleviate data scarcity problem than heuristic method (i.e., REACT) for period-specific activity modeling; 4) SA-MME achieves the best performance among all the methods. For example, on NYC data, it outperforms MME and REACT by more than 7% and 20% when $N = 10$ and $N = 100$, respectively. This tells us that incorporating spatial influence can better model spatiotemporal activities in different time-periods.

6.5.3 Parameter Sensitivity

We also study the effects of some critical parameters on the performances of our proposed methods. Firstly, we study two parameters of SA-MME, i.e., the choice of kernel function

Figure 6.3: Parameter studies for SA-MME.

and the number of sampled nearby regions for constructing $p(\mathbf{r}|\Theta_0)$. Figure 6.3(a) shows the MRR values of SA-MME for region prediction when using the default cosine kernel (denoted as SA-MME-Cosine) and a Euclidean kernel (denoted as SA-MME-Euclidean), which is defined as $\mathcal{K}(\mathbf{u}, \mathbf{v}) = \exp(-\|\mathbf{u} - \mathbf{v}\|)$. We can see from the figure that using cosine kernel is consistently better than using Euclidean kernel on both datasets. For example, when setting $N = 10$, SA-MME-Cosine outperforms SA-MME-Euclidean by around 3.2% on NYC data. The main reason might be that the cosine function is also used in the objective of multimodal embedding to capture the cross-modal correlation. Thus, using cosine kernel in modeling spatial influence is more consistent with the objective of multimodal embedding, which results in better performances. Figure 6.3(b) shows the MRR values of SA-MME for region prediction when varying the number of nearby regions (i.e., $|\mathcal{N}(i)|$) for constructing $p(\mathbf{r}|\Theta_0)$. We can see from the figure that setting $|\mathcal{N}(i)| = 3$ gives us the best results on both datasets. This may indicate that 1) if $|\mathcal{N}(i)|$ is too small, the integrated spatial influence might be too noisy for the modeling and 2) if $|\mathcal{N}(i)|$ is too large, the spatial influence among the sampled regions might be too complicated to be learned.

In addition, we study the effects of the latent dimensionality on our methods (i.e., MME, SA-MME). We also include the conventional multimodal embedding method, i.e., CROSSMAP, as a comparison. In particular, we vary the latent dimensionality d from 30 to 200 (i.e., $d = \{30, 50, 100, 200\}$). Figure 6.4 shows their results (i.e., MRR values) for region prediction on NYC and LA data. We can see from the figure that SA-MME consistently performs better than MME and CROSSMAP for different d . We also notice that all the methods perform better when using larger embedding size. However, the improvements of $d = 200$ over $d = 100$ are relatively small. For example, the MRR

 Figure 6.4: Varying d on NYC and LA data.

values of SA-MME with $d = 200$ are only 1.2% and 2.2% better than $d = 100$ on NYC and LA data, respectively. Therefore, setting $100 \leq d \leq 200$ might be an optimal choice for building ME-based spatiotemporal activity models on our datasets.

6.6 Summary

In this chapter, we study the problem of spatiotemporal activity modeling. Considering the temporal nonstationarity of spatiotemporal activities, we propose to build separate activity models for different time periods. To achieve this under data scarcity, we formulate the problem as a transfer learning problem, where each region has a transferable prior that contains its static information (e.g., region function, geographical semantics), which is temporally stationary. The transferable prior can be leveraged to a specific time period to model the period-specific dynamic information (e.g., local event). Based on this intuition, we extend a multimodal embedding method to a Meta Multimodal Embedding (MME) model, which explicitly learns the transferable priors of regions from data. Moreover, we further incorporate spatial influence between nearby regions into MME and extend it into a Spatial-Aware MME (SA-MME) model. In particular, we propose to use kernel density estimators (KDE) to integrate the spatial influence as a prior estimator of a region. Subsequently, we propose an optimization method, namely Spatial-Aware Stein Variational Gradient Decent (SA-SVGD), to train SA-MME in an end-to-end manner. Experiments on two real-world datasets show that our proposed methods are more effective than several baselines for modeling spatiotemporal activities.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this dissertation, we study and develop latent representation models for mining different types of geo-spatial data, aiming at supporting various kinds of location-based applications.

In Chapter 3, we conduct an all-around evaluation for 12 existing methods for POI recommendation. The methods cover different approaches of representing POIs and users, including various latent representation models (e.g., Weighted Matrix Factorization, ranking-based Matrix Factorization). They also utilize different context information, such as geographical, social and temporal information. We carefully design an evaluation procedure with three large-scale real-world datasets to experiment the models from different perspectives. In particular, we experiment these models on datasets of different sources, different scales, and different sparsities. Moreover, we explore the recommendation accuracy of using different representation models and context information for different types of users. Many new insights are drawn from the experimental results, indicating how, why, when or what kinds of representation models are effective or efficient in POI recommendations. This work provides us an overall picture about deploying and designing POI recommendation methods in real-world scenarios.

In Chapter 4, we propose a novel solution for online anomalous trajectory detection, which consists of a latent representation model and an efficient detection framework. In particular, we first develop a deep generative model GM-VSAE, which is powerful at capturing the sequential information of trajectories and representing their routes in a continuous latent space. We jointly use a Gaussian mixture distribution to model the route representations, which allows us to discover different types of normal routes to support effective anomalous trajectory detection. Next, we propose a novel paradigm based on GM-VSAE, namely *detection-via-generation* framework, for anomalous trajectory detection, where the detection score for a given trajectory can be computed online

and updated in $O(1)$ time. Furthermore, an approximate posterior inference method is proposed that largely reduces the time cost of the detection framework. Extensive experiments on two large-scale real-world trajectory datasets show that our solution is more effective than the state-of-the-art baselines and is efficient for online detection.

In Chapter 5, we tackle the problem of similar region search. We are the first to employ Convolutional Neural Network (CNN) to learn latent representations of regions, and design a deep metric learning framework to learn similarity between regions. The similarity is aware of both the attributes of objects and the relative locations between the objects in a region. Based on the latent representation model that defines the region similarity, we further propose an efficient search algorithm called **ExactSFRS** that solves the similar region search problem. An approximation method called **ApproxSFRS** is also proposed, which can strike a balance between accuracy and efficiency to support different applications. The effectiveness experiments on two large-scale datasets demonstrate our deep metric learning method is significantly better than the baselines for measuring region similarity. The efficiency experiments on three datasets show that **ExactSFRS** is more than $10^5 \times$ faster than the exhaustive search method and over $45 \times$ faster than the best baseline method. Besides, we also illustrate that **ApproxSFRS** can largely reduce the runtime by slightly sacrificing the accuracy.

In Chapter 6, we tackle the problem of spatiotemporal activity modeling, which aims at learning multimodal representations for the spatial, temporal and textual attributes of spatiotemporal activities. Observing that the geo-tagged posts are temporally nonstationary, we first formulate the problem as a *temporal transfer learning* problem, which allows us to extract static region semantics from nonstationary data and use them for modeling activities for different time periods under data scarcity. We propose a novel method based on multimodal embedding, namely Meta Multimodal Embedding (MME), which can learn rich semantics of regions, hours and keywords. Notably, MME explicitly captures the static information of regions as transferable priors that can be effectively adopted to the activity modeling of different time periods. By further considering the spatial influences between nearby regions, we extend MME to a Spatial-Aware MME (SA-MME) model, where the spatial-aware prior knowledge of regions are captured via introducing spatial-aware prior estimators. We also propose a novel spatial-aware Stein Variational Gradient Decent method for training SA-MME in an end-to-end manner. Extensive experiments on two large-scale real-world datasets show that the learned multimodal representations encode rich cross-modal semantics of spatial, temporal and textual attributes of nonstationary activity data, which can be effectively applied for spatial query processing.

7.2 Future Directions

My research opens several potential directions for future work. Here, we present three directions:

- **Anomalous subtrajectory detection and repairing.** For detection anomalous trajectory, we can further design latent representation models for representing different *route segments*. By doing this, we anticipate the detection method to reveal the anomalousness of the subtrajectory for a given trajectory. This would be helpful for taking prompt and accurate actions against the anomalies in real-world scenarios. Moreover, it is even more promising to design models to *repair anomalous trajectories*, which not only detects anomalous trajectories, but also automatically provides solutions and suggestions for fixing the detected anomalies.
- **Relation inference for geo-spatial entities.** Our current researches only learn latent representations for specific geo-spatial entities. However, the relationships associated between those geo-spatial entities are still unclear to us. For example, different POIs usually have different kinds of relations, such as two nearby restaurants are usually *competitors* to each other, and shopping malls and the food courts within are usually *complimentary*. These relations may provide us more detailed and valuable knowledge about urban space and human dynamics. Also, it is also critical to analyze the relations between different types of geo-spatial entities, such as revealing the role of a particular POI in a given region.
- **Reinforcement learning for decision making and scheduling on geographical space.** As the mobility of pedestrians and vehicles becoming increasingly frequent and complicated, there is a urgent need to harness geo-spatial data for optimizing city transportations, such as building intelligent traffic light control system and improving taxi dispatching efficiency. However, these applications can hardly be solved by using conventional latent representation models, since they usually involve many decision making and scheduling processes. As a result, it calls a employment of *reinforcement learning* technique for solving such decision making and scheduling problem, where the main objective of the models is to effectively representing different traffic conditions as *states* and scheduling process as *actions*.

Appendix A

List of Publications

- **Yiding Liu**, Kaiqi Zhao, Gao Cong, Zhifeng Bao. Online Anomalous Trajectory Detection with Deep Generative Sequence Modeling. In Proceedings of the 36rd IEEE International Conference on Data Engineering **ICDE 2020** (Accepted), Dallas, Texas, USA, April 2020.
- Lucas Vinh Tran, Tuan-Anh Nguyen Pham, Yi Tay, **Yiding Liu**, Gao Cong and Xiaoli Li. Interact and Decide: Medley of Sub-Attention Networks for Effective Group Recommendation. In Proceedings of the 42th International ACM SIGIR Conference on Research and Development in Information Retrieval (**SIGIR 2019**), Pages 255-264, Paris, France, July 21-25, 2019.
- Huaxiu Yao, **Yiding Liu**, Ying Wei, Xianfeng Tang, Zhenhui Li. MetaST: Deep Spatial-Temporal Prediction via Knowledge Transfer from Multiple Cities. In The World Wide Web Conference (**WWW 2019**), Pages 2181-2191, San Francisco, CA, USA, May 13 - 17, 2019.
- **Yiding Liu**, Kaiqi Zhao, Gao Cong. Efficient Similar Region Search with Deep Metric Learning. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (**SIGKDD 2018**) (Research Track – Oral), Pages 1850-1859, London, United Kingdom, August 19 - 23, 2018.
- **Yiding Liu**, Tuan-Anh Nguyen Pham, Gao Cong, Quan Yuan. An Experimental Evaluation of Point-of-interest Recommendation in Location-based Social Networks. In Proceedings of the VLDB Endowment (**VLDB 2017**), Volume 10, Issue 10, Pages 1010-1021, June 2017.

- Kaiqi Zhao, **Yiding Liu**, Quan Yuan, Lisi Chen, Zhida Chen, Gao Cong. Towards Personalized Maps: Mining User Preferences from Geo-textual Data. In Proceedings of the VLDB Endowment (**VLDB 2016**), Volume 9, Issue 13, Pages 1545-1548, September 2016.
- Kaiyu Feng, Kaiqi Zhao, **Yiding Liu**, Gao Cong. A System for Region Search and Exploration. In Proceedings of the VLDB Endowment (**VLDB 2016**), Volume 9, Issue 13, Pages 1549-1552, September 2016.

References

- [ABC⁺16] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.
- [AEA⁺18] Mohammed Eunus Ali, Shadman Saqib Eusuf, Kaysar Abdullah, Farhana M Choudhury, J Shane Culpepper, and Timos Sellis. The maximum trajectory coverage query in spatial databases. *Proceedings of the VLDB Endowment*, 12(3):197–209, 2018.
- [AKK18] Gowtham Atluri, Anuj Karpatne, and Vipin Kumar. Spatio-temporal data mining: A survey of problems and methods. *ACM Computing Surveys (CSUR)*, 51(4):83, 2018.
- [ASG13] Hamed Abdelhaq, Christian Sengstock, and Michael Gertz. Eventtweet: Online localized event detection from twitter. *Proceedings of the VLDB Endowment*, 6(12):1326–1329, 2013.
- [AT05] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE TKDE*, 17(6):734–749, 2005.
- [BCTH12] Adel Bolbol, Tao Cheng, Ioannis Tsapakis, and James Haworth. Inferring hybrid transportation modes from sparse gps data using a moving window svm classification. *Computers, Environment and Urban Systems*, 36(6):526–537, 2012.
- [BCV13] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

REFERENCES

- [Ben75] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.
- [Ber06] Pavel Berkhin. Bookmark-coloring algorithm for personalized pagerank computing. *Internet Mathematics*, 3(1):41–62, 2006.
- [BGL⁺94] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a” siamese” time delay neural network. In *Advances in neural information processing systems*, pages 737–744, 1994.
- [BHKT05] Petko Bakalov, Marios Hadjieleftheriou, Eamonn Keogh, and Vassilis J Tsotras. Efficient trajectory joins using symbolic representations. In *Proceedings of the 6th international conference on Mobile data management*, pages 86–93. ACM, 2005.
- [BK07] Robert M Bell and Yehuda Koren. Lessons from the netflix prize challenge. *ACM SIGKDD Explorations Newsletter*, 9(2):75–79, 2007.
- [BMNS08] Viorica Botea, Daniel Mallett, Mario A Nascimento, and Jörg Sander. Pist: an efficient and practical indexing technique for historical spatio-temporal point data. *GeoInformatica*, 12(2):143–168, 2008.
- [BNJ03] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *JMLR*, 3(Jan):993–1022, 2003.
- [BPSW05] Sotiris Brakatsoulas, Dieter Pfoser, Randall Salas, and Carola Wenk. On map-matching vehicle tracking data. In *Proceedings of the 31st international conference on Very large data bases*, pages 853–864. VLDB Endowment, 2005.
- [BPZL15] Preeti Bhargava, Thomas Phan, Jiayu Zhou, and Juhan Lee. Who, what, when, and where: Multi-dimensional collaborative recommendations using tensor factorization on sparse user-generated data. In *WWW*, pages 130–140. IW3C2, 2015.
- [BT06] Petko Bakalov and Vassilis J Tsotras. Continuous spatiotemporal trajectory joins. In *International conference on GeoSensor Networks*, pages 109–128. Springer, 2006.

REFERENCES

- [BYR16] Prithu Banerjee, Pranali Yawalkar, and Sayan Ranu. Mantra: a scalable approach to mining temporally anomalous sub-trajectories. In *SIGKDD*, pages 1415–1424. ACM, 2016.
- [C⁺15] François Chollet et al. Keras, 2015.
- [CAC12] Rui Chen, Gergely Acs, and Claude Castelluccia. Differentially private sequential data publication via variable-length n-grams. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 638–649. ACM, 2012.
- [CFDS12] Rui Chen, Benjamin Fung, Bipin C Desai, and Néria M Sossou. Differentially private transit data publication: a case study on the montreal transportation system. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 213–221. ACM, 2012.
- [Cho17] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.
- [CMWM10] Philippe Cudre-Mauroux, Eugene Wu, and Samuel Madden. Trajstore: An adaptive storage system for very large trajectory data sets. In *2010 IEEE 26th International Conference on Data Engineering (ICDE 2010)*, pages 109–120. IEEE, 2010.
- [CÖO05] Lei Chen, M Tamer Özsu, and Vincent Oria. Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 491–502. ACM, 2005.
- [CP09] Yun Chen and Jignesh M Patel. Design and evaluation of trajectory join algorithms. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 266–275. ACM, 2009.
- [CPP⁺18] Buru Chang, Yonggyu Park, Donghyeon Park, Seongsoon Kim, and Jaewoo Kang. Content-aware hierarchical point-of-interest embedding model for successive poi recommendation. In *IJCAI*, pages 3301–3307, 2018.

REFERENCES

- [CSAT17] Jinghui Chen, Saket Sathe, Charu Aggarwal, and Deepak Turaga. Outlier detection with autoencoder ensembles. In *SDM*, pages 90–98. SIAM, 2017.
- [CVMG⁺14] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- [CWPZ18] Peng Cui, Xiao Wang, Jian Pei, and Wenwu Zhu. A survey on network embedding. *IEEE Transactions on Knowledge and Data Engineering*, 31(5):833–852, 2018.
- [CYKL12] Chen Cheng, Haiqin Yang, Irwin King, and Michael R Lyu. Fused matrix factorization with geographical and social influence in location-based social networks. In *AAAI*, 2012.
- [CYLK13] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. Where you like to go next: Successive point-of-interest recommendation. In *IJCAI*, volume 13, pages 2605–2611, 2013.
- [CZC⁺13] Chao Chen, Daqing Zhang, Pablo Samuel Castro, Nan Li, Lin Sun, Shijian Li, and Zonghui Wang. iboat: Isolation-based online anomalous trajectory detection. *IEEE TITS*, 14(2):806–818, 2013.
- [CZC⁺15] Xuefeng Chen, Yifeng Zeng, Gao Cong, Shengchao Qin, Yanping Xiang, and Yuanshun Dai. On information coverage for location category based point-of-interest recommendation. In *AAAI*, pages 37–43, 2015.
- [DAM⁺15] Romain Deveaud, M-Dyaa Albakour, Craig Macdonald, Iadh Ounis, et al. Experiments with a venue-centric model for personalised and time-aware venue suggestion. In *CIKM*, pages 53–62. ACM, 2015.
- [DCG⁺18] Xin Ding, Lu Chen, Yunjun Gao, Christian S Jensen, and Hujun Bao. Ultra-man: a unified platform for big trajectory data management and analytics. *Proceedings of the VLDB Endowment*, 11(7):787–799, 2018.
- [DHS11] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12(Jul):2121–2159, 2011.

REFERENCES

- [DIIM04] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *SoCG*, pages 253–262. ACM, 2004.
- [DMMD18] Steven Derby, Paul Miller, Brian Murphy, and Barry Devereux. Using sparse semantic embeddings learned from multimodal text and image data to model human conceptual knowledge. *arXiv preprint arXiv:1809.02534*, 2018.
- [DTS08] Hui Ding, Goce Trajcevski, and Peter Scheuermann. Efficient similarity join of large sets of moving object trajectories. In *2008 15th International Symposium on Temporal Representation and Reasoning*, pages 79–87. IEEE, 2008.
- [EKHT14] M Erlebach, P Klapka, M Halás, and P Tonev. Inner structure of functional region: theoretical aspects. In *17th International Colloquium on Regional Science, Conference Proceedings*, pages 722–727. Masaryk University Brno, 2014.
- [EKS⁺96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *SIGKDD*, volume 96, pages 226–231, 1996.
- [FAL17] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.
- [FCAC17] Shanshan Feng, Gao Cong, Bo An, and Yeow Meng Chee. Poi2vec: Geographical latent representation for predicting future visitors. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- [FLZ⁺15] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. Personalized ranking metric embedding for next new poi recommendation. In *IJCAI*, 2015.

REFERENCES

- [FLZ⁺18] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 World Wide Web Conference*, pages 1459–1468. International World Wide Web Conferences Steering Committee, 2018.
- [FWD⁺19] Yanjie Fu, Pengyang Wang, Jiadi Du, Le Wu, and Xiaolin Li. Efficient region embedding with multi-view spatial networks: A perspective of locality-constrained spatial autocorrelations. 2019.
- [FYL13] Gregory Ferenec, Mao Ye, and Wang-Chien Lee. Location recommendation for out-of-town users in location-based social networks. In *CIKM*, pages 721–726. ACM, 2013.
- [FZZ⁺15] Wei Feng, Chao Zhang, Wei Zhang, Jiawei Han, Jianyong Wang, Charu Aggarwal, and Jianbin Huang. Streamcube: hierarchical spatio-temporal hashtag clustering for event exploration over the twitter stream. In *2015 IEEE 31st International Conference on Data Engineering*, pages 1561–1572. IEEE, 2015.
- [GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [GH12] Michael U Gutmann and Aapo Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13(Feb):307–361, 2012.
- [Gir15] Ross Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015.
- [GL16] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.
- [GMH13] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.

REFERENCES

- [GTHL13] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. Exploring temporal effects for location recommendation on location-based social networks. In *RecSys*, pages 93–100. ACM, 2013.
- [GTHL15] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. Content-aware point of interest recommendation on location-based social networks. In *AAAI*, pages 1721–1727, 2015.
- [GXLZ11] Yong Ge, Hui Xiong, Chuanren Liu, and Zhi-Hua Zhou. A taxi driving fraud detection system. In *ICDM*, pages 181–190. IEEE, 2011.
- [GXZ⁺10] Yong Ge, Hui Xiong, Zhi-hua Zhou, Hasan Ozdemir, Jannite Yu, and Kuo Chu Lee. Top-eye: Top-k evolving trajectory outlier detection. In *CIKM*, pages 1733–1736. ACM, 2010.
- [H⁺70] Richard A Harshman et al. Foundations of the parafac procedure: Models and conditions for an” explanatory” multimodal factor analysis. 1970.
- [HA15] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *ICLR Workshop*, pages 84–92. Springer, 2015.
- [HAG⁺12] Liangjie Hong, Amr Ahmed, Siva Gurumurthy, Alexander J Smola, and Kostas Tsioutsoulis. Discovering geographical topics in the twitter stream. In *Proceedings of the 21st international conference on World Wide Web*, pages 769–778. ACM, 2012.
- [HBL17] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.
- [HCM⁺15] Xi He, Graham Cormode, Ashwin Machanavajjhala, Cecilia M Procopiuc, and Divesh Srivastava. Dpt: differentially private trajectory synthesis using hierarchical reference systems. *Proceedings of the VLDB Endowment*, 8(11):1154–1165, 2015.
- [HE13] Bo Hu and Martin Ester. Spatial topic modeling in online social media for location recommendation. In *RecSys*, pages 25–32. ACM, 2013.

REFERENCES

- [HGDG17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.
- [HKV08] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, pages 263–272. IEEE, 2008.
- [HLH17] Xiao Huang, Jundong Li, and Xia Hu. Label informed attributed network embedding. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 731–739. ACM, 2017.
- [HLL⁺16] Jing He, Xin Li, Lejian Liao, Dandan Song, and William K Cheung. Inferring a personalized next point-of-interest recommendation model with latent behavior patterns. In *AAAI*, 2016.
- [HQY⁺13] Jian Huang, Shaoqing Qiao, Haitao Yu, Jinhui Qie, and Chunwei Liu. Parallel map matching on massive vehicle gps data using mapreduce. In *2013 IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing*, pages 1498–1503. IEEE, 2013.
- [HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [HSL14] Longke Hu, Aixin Sun, and Yong Liu. Your neighbors affect your ratings: on geographical neighborhood influence to rating prediction. In *SIGIR*, pages 345–354. ACM, 2014.
- [HYL17] William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017.
- [HZC⁺17] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

REFERENCES

- [HZRS14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, pages 346–361. Springer, 2014.
- [HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [HZZC18] Chao Huang, Junbo Zhang, Yu Zheng, and Nitesh V Chawla. Deepcrime: attentive hierarchical recurrent networks for crime prediction. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1423–1432. ACM, 2018.
- [IS11] Tsuyoshi Idé and Masashi Sugiyama. Trajectory regression on road networks. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.
- [JSZ08] Hoyoung Jeung, Heng Tao Shen, and Xiaofang Zhou. Convoy queries in spatio-temporal databases. In *2008 IEEE 24th International Conference on Data Engineering*, pages 1457–1459. IEEE, 2008.
- [JYZ⁺08] Hoyoung Jeung, Man Lung Yiu, Xiaofang Zhou, Christian S Jensen, and Heng Tao Shen. Discovery of convoys in trajectory databases. *Proceedings of the VLDB Endowment*, 1(1):1068–1080, 2008.
- [JZT⁺17] Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: an unsupervised and generative approach to clustering. In *IJCAI*, pages 1965–1972. AAAI Press, 2017.
- [K⁺13] Brian Kulis et al. Metric learning: A survey. *Foundations and Trends® in Machine Learning*, 5(4):287–364, 2013.
- [KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [KBV09] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

REFERENCES

- [KCTS15] Mohamed Kafsi, Henriette Cramer, Bart Thomee, and David A Shamma. Describing and understanding neighborhood characteristics through online social media. In *WWW*, pages 549–559. International World Wide Web Conferences Steering Committee, 2015.
- [KH15] John Krumm and Eric Horvitz. Eyewitness: Identifying local events via space-time signals in twitter feeds. In *Proceedings of the 23rd sigspatial international conference on advances in geographic information systems*, page 20. ACM, 2015.
- [KHT13] P Klapka, M Halás, and P Tonev. Functional regions: concept and types. In *16th International Colloquium on Regional Sciences, Conference Proceedings. Brno, Masaryk University*, pages 94–101, 2013.
- [KHY15] Younghoon Kim, Jiawei Han, and Cangzhou Yuan. Toptrac: Topical trajectory pattern mining. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 587–596. ACM, 2015.
- [KIH⁺13] Takeshi Kurashima, Tomoharu Iwata, Takahide Hoshide, Noriko Takaya, and Ko Fujimura. Geo topic model: joint modeling of user’s activity area and interests for location recommendation. In *WSDM*, pages 375–384. ACM, 2013.
- [KKSS14] Christoph Carl Kling, Jérôme Kunegis, Sergej Sizov, and Steffen Staab. Detecting non-gaussian geographical topics in tagged photo collections. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 603–612. ACM, 2014.
- [KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.
- [KW13] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [KXM⁺19] Xiangjie Kong, Feng Xia, Kai Ma, Jianxin Li, and Qiuyuan Yang. Discovering transit-oriented development regions of megacities using heterogeneous urban data. *IEEE Transactions on Computational Social Systems*, 2019.

REFERENCES

- [LBH09] Christoph H Lampert, Matthew B Blaschko, and Thomas Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *IEEE TPAMI*, 31(12):2129–2142, 2009.
- [LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.
- [LCI09] Anthony JT Lee, Yi-An Chen, and Weng-Chong Ip. Mining frequent trajectory patterns in spatial-temporal databases. *Information Sciences*, 179(13):2218–2231, 2009.
- [LCJB⁺89] Yann Le Cun, Lionel D Jackel, Brian Boser, John S Denker, Henry P Graf, Isabelle Guyon, Don Henderson, Richard E Howard, and William Hubbard. Handwritten digit recognition: Applications of neural network chips and automatic learning. *IEEE Communications Magazine*, 27(11):41–46, 1989.
- [LCL⁺15] Xutao Li, Gao Cong, Xiao-Li Li, Tuan-Anh Nguyen Pham, and Shonali Krishnaswamy. Rank-geofm: A ranking based geographical factorization method for point of interest recommendation. In *SIGIR*, pages 433–442. ACM, 2015.
- [LCSC19] Xiucheng Li, Gao Cong, Aixin Sun, and Yun Cheng. Learning travel time distributions with deep generative model. In *The World Wide Web Conference*, pages 1017–1027. ACM, 2019.
- [LFGM15] Géraud Le Falher, Aristides Gionis, and Michael Mathioudakis. Where is the soho of rome? measures and algorithms for finding similar neighborhoods in cities. In *ICWSM*, pages 228–237, 2015.
- [LFYX13] Bin Liu, Yanjie Fu, Zijun Yao, and Hui Xiong. Learning geographical preferences for point-of-interest recommendation. In *KDD*, pages 1043–1051. ACM, 2013.
- [LGRZ16] Huayu Li, Yong Ge, Hong Richang, and Hengshu Zhu. Point-of-interest recommendations: Learning potential check-ins from friends. In *KDD*, pages 975–984. ACM, 2016.

REFERENCES

- [LGZ⁺15] Defu Lian, Yong Ge, Fuzheng Zhang, Nicholas Jing Yuan, Xing Xie, Tao Zhou, and Yong Rui. Content-aware collaborative filtering for location recommendation based on human mobility data. In *ICDM*, pages 261–270. IEEE, 2015.
- [LHL08] Jae-Gil Lee, Jiawei Han, and Xiaolei Li. Trajectory outlier detection: A partition-and-detect framework. In *ICDE*, pages 140–149. IEEE, 2008.
- [LHLG08] Jae-Gil Lee, Jiawei Han, Xiaolei Li, and Hector Gonzalez. Traiclass: trajectory classification using hierarchical region-based and trajectory-based clustering. *Proceedings of the VLDB Endowment*, 1(1):1081–1094, 2008.
- [LHW07] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. Trajectory clustering: a partition-and-group framework. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 593–604. ACM, 2007.
- [LHZG15] Huayu Li, Richang Hong, Shiai Zhu, and Yong Ge. Point-of-interest recommender systems: A separate-space perspective. In *ICDM*, pages 231–240. IEEE, 2015.
- [LLAM13] Xin Liu, Yong Liu, Karl Aberer, and Chunyan Miao. Personalized point-of-interest recommendation by mining users’ preference transition. In *CIKM*, pages 733–738. ACM, 2013.
- [LLL⁺12] Siyuan Liu, Ce Liu, Qiong Luo, Lionel M Ni, and Ramayya Krishnan. Calibrating large scale vehicle trajectory data. In *2012 IEEE 13th International Conference on Mobile Data Management*, pages 222–231. IEEE, 2012.
- [LLL⁺16] Yanchi Liu, Chuanren Liu, Bin Liu, Meng Qu, and Hui Xiong. Unified point-of-interest recommendation with temporal interval assessment. In *KDD*, pages 1015–1024. ACM, 2016.
- [LLLH10] Zhenhui Li, Jae-Gil Lee, Xiaolei Li, and Jiawei Han. Incremental clustering for trajectories. In *International Conference on Database Systems for Advanced Applications*, pages 32–46. Springer, 2010.

REFERENCES

- [LPCY17] Yiding Liu, Tuan-Anh Nguyen Pham, Gao Cong, and Quan Yuan. An experimental evaluation of point-of-interest recommendation in location-based social networks (full version). In <http://spatialkeyword.sce.ntu.edu.sg/eval-vldb17/>, 2017.
- [LSEM12] Justin J Levandoski, Mohamed Sarwat, Ahmed Eldawy, and Mohamed F Mokbel. Lars: A location-aware recommender system. In *ICDE*, pages 450–461. IEEE, 2012.
- [LSP06] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, volume 2, pages 2169–2178. IEEE, 2006.
- [LW16] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances In Neural Information Processing Systems*, pages 2378–2386, 2016.
- [LWS11] Ryong Lee, Shoko Wakamiya, and Kazutoshi Sumiya. Discovery of unusual regional social activities using geo-tagged microblogs. *World Wide Web*, 14(4):321–349, 2011.
- [LWSM14] Yong Liu, Wei Wei, Aixin Sun, and Chunyan Miao. Exploiting geographical neighborhood characteristics for location recommendation. In *CIKM*, pages 739–748. ACM, 2014.
- [LXP⁺15] Bin Liu, Hui Xiong, Spiros Papadimitriou, Yanjie Fu, and Zijun Yao. A general geographical probabilistic factor model for point of interest recommendation. *IEEE TKDE*, 27(5):1167–1179, 2015.
- [LXZ⁺15] Defu Lian, Xing Xie, Vincent W Zheng, Nicholas Jing Yuan, Fuzheng Zhang, and Enhong Chen. Cepr: A collaborative exploration and periodically returning model for location prediction. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(1):8, 2015.
- [LXZ⁺17] Zhongjian Lv, Jiajie Xu, Pengpeng Zhao, Guanfeng Liu, Lei Zhao, and Xiaofang Zhou. Outlier trajectory detection: A trajectory analytics based approach. In *DASFAA*, pages 231–246. Springer, 2017.

REFERENCES

- [LZC⁺11] Yunhao Liu, Yiyang Zhao, Lei Chen, Jian Pei, and Jinsong Han. Mining frequent trajectory patterns for activity monitoring using radio frequency tag arrays. *IEEE Transactions on Parallel and Distributed Systems*, 23(11):2138–2149, 2011.
- [LZC⁺18a] Xiucheng Li, Kaiqi Zhao, Gao Cong, Christian S Jensen, and Wei Wei. Deep representation learning for trajectory similarity computation. In *ICDE*, pages 617–628. IEEE, 2018.
- [LZC18b] Yiding Liu, Kaiqi Zhao, and Gao Cong. Efficient similar region search with deep metric learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1850–1859. ACM, 2018.
- [LZX13] Defu Lian, Vincent W Zheng, and Xing Xie. Collaborative filtering meets next check-in location prediction. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 231–232. ACM, 2013.
- [LZX⁺14] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. Geomf: Joint geographical modeling and matrix factorization for point-of-interest recommendation. In *KDD*, pages 831–840. ACM, 2014.
- [LZZ⁺09] Yin Lou, Chengyang Zhang, Yu Zheng, Xing Xie, Wei Wang, and Yan Huang. Map-matching for low-sampling-rate gps trajectories. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 352–361. ACM, 2009.
- [Man93] RH Manukian. Trajectory correction with sequential filtering. Technical report, Deutsches Elektronen-Synchrotron DESY, 1993.
- [MB05] Frederic Morin and Yoshua Bengio. Hierarchical probabilistic neural network language model. In *Aistats*, volume 5, pages 246–252. Citeseer, 2005.
- [Mil04] Harvey J Miller. Tobler’s first law and spatial analysis. *Annals of the Association of American Geographers*, 94(2):284–289, 2004.
- [MLKL11] Hao Ma, Chao Liu, Irwin King, and Michael R Lyu. Probabilistic factor models for web site recommendation. In *SIGIR*, pages 265–274. ACM, 2011.

REFERENCES

- [MNSB15] Cristina Ioana Muntean, Franco Maria Nardini, Fabrizio Silvestri, and Ranieri Baraglia. On learning prediction models for tourists paths. *ACM TIST*, 7(1):8, 2015.
- [MPTG09] Anna Monreale, Fabio Pinelli, Roberto Trasarti, and Fosca Giannotti. Wherenext: a location predictor on trajectory pattern mining. In *KDD*, pages 637–646. ACM, 2009.
- [MRA⁺16] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Lstm-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*, 2016.
- [MSC⁺13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
- [MXJY16] Junhua Mao, Jiajing Xu, Kevin Jing, and Alan L Yuille. Training and evaluating multimodal word embeddings with large-scale web annotated images. In *Advances in Neural Information Processing Systems*, pages 442–450, 2016.
- [MYQZ09] Qiang Ma, Bin Yang, Weining Qian, and Aoying Zhou. Query processing of massive trajectory data based on mapreduce. In *Proceedings of the first international workshop on Cloud data management*, pages 9–16. ACM, 2009.
- [MZWL18] Chen Ma, Yingxue Zhang, Qinglong Wang, and Xue Liu. Point-of-interest recommendation: Exploiting self-attentive autoencoders with neighbor-aware influence. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 697–706. ACM, 2018.
- [MZZ⁺18] Yuexin Ma, Xinge Zhu, Sibozhang, Ruigang Yang, Wenping Wang, and Dinesh Manocha. Trafficpredict: Trajectory prediction for heterogeneous traffic-agents. *arXiv preprint arXiv:1811.02146*, 2018.
- [NK09] Paul Newson and John Krumm. Hidden markov map matching through noise and sparseness. In *SIGSPATIAL*, pages 336–343. ACM, 2009.

REFERENCES

- [NSLM12] Anastasios Noulas, Salvatore Scellato, Neal Lathia, and Cecilia Mascolo. A random walk around the city: New venue recommendation in location-based social networks. In *SOCIALCOM-PASSAT*, pages 144–153. IEEE, 2012.
- [PARS14] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.
- [PCS18] Pouya Pezeshkpour, Liyan Chen, and Sameer Singh. Embedding multi-modal relational data for knowledge base completion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3208–3218, 2018.
- [PLC17] Tuan-Anh Nguyen Pham, Xutao Li, and Gao Cong. A general model for out-of-town region recommendation. In *WWW*, pages 401–410. IW3C2, 2017.
- [Por05] Fatih Porikli. Integral histogram: A fast way to extract histograms in cartesian spaces. In *CVPR*, volume 1, pages 829–836. IEEE, 2005.
- [PPCY13] Daniel Preotiuc-Pietro, Justin Cranshaw, and Tae Yano. Exploring venue-based city-to-city similarity measures. In *KDD Workshop on Urban Computing*, page 16. ACM, 2013.
- [PZC⁺08] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. One-class collaborative filtering. In *ICDM*, pages 502–511. IEEE, 2008.
- [RFGST09] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461. AUAI Press, 2009.
- [RHGS15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.

REFERENCES

- [Ric18] Keven Richly. A survey on trajectory data management for hybrid transactional and analytical workloads. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 562–569. IEEE, 2018.
- [RSCM14] Ali Sharif Razavian, Josephine Sullivan, Stefan Carlsson, and Atsuto Maki. Visual instance retrieval with deep convolutional networks. *arXiv preprint arXiv:1412.6574*, 2014.
- [RSF17] Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. struc2vec: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 385–394. ACM, 2017.
- [SFHS07] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer, 2007.
- [SHK⁺14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958, 2014.
- [Siz10] Sergej Sizov. Geofolk: latent spatial semantics in web 2.0 social media. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 281–290. ACM, 2010.
- [SJK⁺13] Dongxu Shao, Kaifeng Jiang, Thomas Kister, Stephane Bressan, and Kian-Lee Tan. Publishing trajectory with differential privacy: A priori vs. a posteriori sampling mechanisms. In *International Conference on Database and Expert Systems Applications*, pages 357–365. Springer, 2013.
- [SLEM14] Mohamed Sarwat, Justin J Levandoski, Ahmed Eldawy, and Mohamed F Mokbel. Lars*: An efficient and scalable location-aware recommender system. *IEEE TKDE*, 26(6):1384–1399, 2014.
- [SLJ⁺15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

REFERENCES

- [SMM⁺11] Salvatore Scellato, Mirco Musolesi, Cecilia Mascolo, Vito Latora, and Andrew T Campbell. Nextplace: a spatio-temporal prediction framework for pervasive systems. In *International Conference on Pervasive Computing*, pages 152–169. Springer, 2011.
- [SOM10] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM, 2010.
- [SWX⁺18] Li Song, Ruijia Wang, Ding Xiao, Xiaotian Han, Yanan Cai, and Chuan Shi. Anomalous trajectory detection using recurrent neural network. In *ADMA*, pages 263–277. Springer, 2018.
- [SZ14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [SZH⁺10] Chang Sheng, Yu Zheng, Wynne Hsu, Mong Li Lee, and Xing Xie. Answering top-k similar region queries. In *DSAA*, pages 186–201. Springer, 2010.
- [SZH⁺15] Han Su, Kai Zheng, Jiamin Huang, Haozhou Wang, and Xiaofang Zhou. Calibrating trajectory data for spatio-temporal similarity analysis. *The VLDB Journal—The International Journal on Very Large Data Bases*, 24(1):93–116, 2015.
- [SZW⁺13] Han Su, Kai Zheng, Haozhou Wang, Jiamin Huang, and Xiaofang Zhou. Calibrating trajectory data for similarity-based analysis. In *Proceedings of the 2013 ACM SIGMOD international conference on management of data*, pages 833–844. ACM, 2013.
- [SZZ⁺18] Ying Sun, Hengshu Zhu, Fuzhen Zhuang, Jingjing Gu, and Qing He. Exploring the urban region-of-interest through the analysis of online map search queries. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2269–2278. ACM, 2018.

REFERENCES

- [TIT⁺18] Yusuke Tanaka, Tomoharu Iwata, Toshiyuki Tanaka, Takeshi Kurashima, Maya Okawa, and Hiroyuki Toda. Refining coarse-grained spatial data using auxiliary spatial data sets with various granularities. *arXiv preprint arXiv:1809.07952*, 2018.
- [TM08] Manolis Terrovitis and Nikos Mamoulis. Privacy preservation in the publication of trajectories. In *MDM*, volume 8, pages 65–72, 2008.
- [Tob70] Waldo R Tobler. A computer movie simulating urban growth in the detroit region. *Economic geography*, 46(sup1):234–240, 1970.
- [TQW⁺15] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077. International World Wide Web Conferences Steering Committee, 2015.
- [TSJ15] Giorgos Tolias, Ronan Sifre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. *arXiv preprint arXiv:1511.05879*, 2015.
- [TY10] Kim-Chuan Toh and Sangwoon Yun. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of optimization*, 6(615-640):15, 2010.
- [Usa09] OV Usatenko. *Random finite-valued dynamical systems: additive Markov chain approach*. Cambridge Scientific Publishers, 2009.
- [WFW⁺17] Hao Wang, Yanmei Fu, Qinyong Wang, Hongzhi Yin, Changying Du, and Hui Xiong. A location-sentiment-aware recommender system for both home-town and out-of-town users. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1143. ACM, 2017.
- [WFZ⁺18] Pengyang Wang, Yanjie Fu, Jiawei Zhang, Xiaolin Li, and Dan Lin. Learning urban community structures: A collective embedding perspective with periodic spatial-temporal mobility graphs. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 9(6):63, 2018.

REFERENCES

- [WG15] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, pages 2794–2802, 2015.
- [WHG17] Xiaolong Wang, Kaiming He, and Abhinav Gupta. Transitive invariance for self-supervised visual representation learning. *arXiv preprint arXiv:1708.02901*, 2017.
- [WHKD12] Jürgen Wiest, Matthias Höffken, Ulrich Kreßel, and Klaus Dietmayer. Probabilistic trajectory prediction with gaussian mixture models. In *2012 IEEE Intelligent Vehicles Symposium*, pages 141–146. IEEE, 2012.
- [WJW⁺19] Hongjian Wang, Porter Jenkins, Hua Wei, Fei Wu, and Zhenhui Li. Learning task-specific city region partition. In *The World Wide Web Conference*, pages 3300–3306. ACM, 2019.
- [WK13] Maximilian Walther and Michael Kaisser. Geo-spatial event detection in the twitter stream. In *European conference on information retrieval*, pages 356–367. Springer, 2013.
- [WKGL16] Hongjian Wang, Daniel Kifer, Corina Graif, and Zhenhui Li. Crime rate inference with big data. In *KDD*, pages 635–644. ACM, 2016.
- [WL17] Hongjian Wang and Zhenhui Li. Region representation learning via mobility flow. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 237–246. ACM, 2017.
- [WSOC18] Hao Wang, Huawei Shen, Wentao Ouyang, and Xueqi Cheng. Exploiting poi-specific geographical influence for point-of-interest recommendation. In *IJCAI*, pages 3877–3883, 2018.
- [WSZ17] Hao Wu, Weiwei Sun, and Baihua Zheng. A fast trajectory outlier detection approach via driving behavior modeling. In *CIKM*, pages 837–846. ACM, 2017.
- [WTM13] Hao Wang, Manolis Terrovitis, and Nikos Mamoulis. Location recommendation in location-based social networks using user check-in data. In *SIGSPATIAL*, pages 374–383. ACM, 2013.

REFERENCES

- [WWXM07] Chong Wang, Jinggang Wang, Xing Xie, and Wei-Ying Ma. Mining geographic knowledge using location aware topic model. In *Proceedings of the 4th ACM workshop on Geographical information retrieval*, pages 65–70. ACM, 2007.
- [WYC⁺15] Weiqing Wang, Hongzhi Yin, Ling Chen, Yizhou Sun, Shazia Sadiq, and Xiaofang Zhou. Geo-sage: a geographical sparse additive generative model for spatial item recommendation. In *KDD*, pages 1255–1264. ACM, 2015.
- [WYL⁺15] Yingzi Wang, Nicholas Jing Yuan, Defu Lian, Linli Xu, Xing Xie, Enhong Chen, and Yong Rui. Regularity and conformity: Location prediction using heterogeneous mobility data. In *KDD*, pages 1275–1284. ACM, 2015.
- [WZL17] Chong Wang, Xue Zhang, and Xipeng Lan. How to train triplet networks with 100k identities? *arXiv preprint arXiv:1709.02940*, 2017.
- [WZX⁺14] Haozhou Wang, Kai Zheng, Jiajie Xu, Bolong Zheng, Xiaofang Zhou, and Shazia Sadiq. Sharkdb: An in-memory column-oriented trajectory storage. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pages 1409–1418. ACM, 2014.
- [WZY16] Ying Wei, Yu Zheng, and Qiang Yang. Transfer knowledge between cities. In *KDD*, pages 1905–1914, 2016.
- [XLP17] Dong Xie, Feifei Li, and Jeff M Phillips. Distributed trajectory similarity search. *Proceedings of the VLDB Endowment*, 10(11):1478–1489, 2017.
- [XMC⁺16] Xike Xie, Benjin Mei, Jinchuan Chen, Xiaoyong Du, and Christian S Jensen. Elite: an elastic infrastructure for big spatiotemporal trajectories. *The VLDB Journal—The International Journal on Very Large Data Bases*, 25(4):473–493, 2016.
- [YBZ⁺17] Carl Yang, Lanxiao Bai, Chao Zhang, Quan Yuan, and Jiawei Han. Bridging collaborative filtering and semi-supervised learning: a neural approach for poi recommendation. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1245–1254. ACM, 2017.

REFERENCES

- [YCH⁺11] Zhijun Yin, Liangliang Cao, Jiawei Han, Chengxiang Zhai, and Thomas Huang. Geographical topic discovery and comparison. In *Proceedings of the 20th international conference on World wide web*, pages 247–256. ACM, 2011.
- [YCM⁺13a] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. Time-aware point-of-interest recommendation. In *SIGIR*, pages 363–372. ACM, 2013.
- [YCM⁺13b] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. Who, where, when and what: discover spatio-temporal topics for twitter users. In *KDD*, pages 605–613. ACM, 2013.
- [YCRW14] Yanwei Yu, Lei Cao, Elke A Rundensteiner, and Qin Wang. Detecting moving object outliers in massive-scale trajectory streams. In *SIGKDD*, pages 422–431. ACM, 2014.
- [YCS14] Quan Yuan, Gao Cong, and Aixin Sun. Graph-based point-of-interest recommendation with geographical and temporal influences. In *CIKM*, pages 659–668. ACM, 2014.
- [YCZ⁺15] Quan Yuan, Gao Cong, Kaiqi Zhao, Zongyang Ma, and Aixin Sun. Who, where, when, and what: A nonparametric bayesian approach to context-aware recommendation and search for twitter users. *ACM TOIS*, 33(1):2, 2015.
- [YCZ⁺16] Hongzhi Yin, Bin Cui, Xiaofang Zhou, Weiqing Wang, Zi Huang, and Shazia Sadiq. Joint modeling of user check-in behaviors for real-time point-of-interest recommendation. *ACM Transactions on Information Systems (TOIS)*, 35(2):11, 2016.
- [YCZB19] Di Yao, Gao Cong, Chao Zhang, and Jingping Bi. Computing trajectory similarity in linear time: A generic seed-guided neural metric learning approach. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 1358–1369. IEEE, 2019.
- [YFL⁺18] Zijun Yao, Yanjie Fu, Bin Liu, Wangsu Hu, and Hui Xiong. Representing urban functions through zone embedding with human mobility patterns. In *IJCAI*, pages 3919–3925, 2018.

REFERENCES

- [YHPC18] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing. *ieee Computational intelligence magazine*, 13(3):55–75, 2018.
- [YKD⁺18] Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pages 7332–7342, 2018.
- [YKTL14] Josh Jia-Ching Ying, Wen-Ning Kuo, Vincent S Tseng, and Eric Hsueh-Chan Lu. Mining user check-in behavior with a random walk for urban point-of-interest recommendations. *ACM TIST*, 5(3):40, 2014.
- [YLWT11] Josh Jia-Ching Ying, Wang-Chien Lee, Tz-Chiao Weng, and Vincent S Tseng. Semantic trajectory mining for location prediction. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 34–43. ACM, 2011.
- [YSC⁺13] Hongzhi Yin, Yizhou Sun, Bin Cui, Zhiting Hu, and Ling Chen. Lcars: a location-content-aware recommender system. In *KDD*, pages 221–229. ACM, 2013.
- [YSZ⁺17] Guan Yuan, Penghui Sun, Jie Zhao, Daxing Li, and Canwei Wang. A review of moving object trajectory clustering algorithms. *Artificial Intelligence Review*, 47(1):123–144, 2017.
- [YTW⁺18] Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, Yanwei Yu, and Zhenhui Li. Modeling spatial-temporal dynamics for traffic prediction. *arXiv preprint arXiv:1803.01254*, 2018.
- [YWK⁺18] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. Deep multi-view spatial-temporal network for taxi demand prediction. *AAAI Conference on Artificial Intelligence*, 2018.
- [YYL10] Mao Ye, Peifeng Yin, and Wang-Chien Lee. Location recommendation for location-based social networks. In *SIGSPATIAL*, pages 458–461. ACM, 2010.

REFERENCES

- [YYLL11] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *SIGIR*, pages 325–334. ACM, 2011.
- [YZQ16] Dingqi Yang, Daqing Zhang, and Bingqing Qu. Participatory cultural mapping based on collective behavior data in location-based social networks. *ACM TIST*, 7(3):30, 2016.
- [YZS⁺15] Hongzhi Yin, Xiaofang Zhou, Yingxia Shao, Hao Wang, and Shazia Sadiq. Joint modeling of user check-in behaviors for point-of-interest recommendation. In *CIKM*, pages 1631–1640. ACM, 2015.
- [YZX12] Jing Yuan, Yu Zheng, and Xing Xie. Discovering regions of different functions in a city using human mobility and pois. In *KDD*, pages 186–194. ACM, 2012.
- [YZXS11] Jing Yuan, Yu Zheng, Xing Xie, and Guangzhong Sun. Driving with knowledge from the physical world. In *SIGKDD*, pages 316–324. ACM, 2011.
- [YZZ⁺10a] Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, and Yan Huang. T-drive: driving directions based on taxi trajectories. In *SIGSPATIAL*, pages 99–108. ACM, 2010.
- [YZZ⁺10b] Jing Yuan, Yu Zheng, Chengyang Zhang, Xing Xie, and Guang-Zhong Sun. An interactive-voting based map matching algorithm. In *Proceedings of the 2010 Eleventh International Conference on Mobile Data Management*, pages 43–52. IEEE Computer Society, 2010.
- [YZZ⁺17] Quan Yuan, Wei Zhang, Chao Zhang, Xinhe Geng, Gao Cong, and Jiawei Han. Pred: periodic region detection for mobility modeling of social media users. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 263–272. ACM, 2017.
- [ZBLL17] Guanjie Zheng, Susan L Brantley, Thomas Lauvaux, and Zhenhui Li. Contextual spatial outlier detection with metric learning. In *SIGKDD*, pages 2161–2170. ACM, 2017.

REFERENCES

- [ZC13] Jia-Dong Zhang and Chi-Yin Chow. igsir: personalized geo-social location recommendation: a kernel density estimation approach. In *SIGSPATIAL*, pages 334–343. ACM, 2013.
- [ZC15a] Jia-Dong Zhang and Chi-Yin Chow. Geosoca: Exploiting geographical, social and categorical correlations for point-of-interest recommendations. In *SIGIR*, pages 443–452. ACM, 2015.
- [ZC15b] Jia-Dong Zhang and Chi-Yin Chow. Spatiotemporal sequential influence modeling for location recommendations: A gravity-based approach. *ACM TIST*, 7(1):11, 2015.
- [ZCL⁺10] Yu Zheng, Yukun Chen, Quannan Li, Xing Xie, and Wei-Ying Ma. Understanding transportation modes based on gps data for web applications. *ACM Transactions on the Web (TWEB)*, 4(1):1, 2010.
- [ZCL14] Jia-Dong Zhang, Chi-Yin Chow, and Yanhua Li. Lore: Exploiting sequential influence for location recommendations. In *SIGSPATIAL*, pages 103–112. ACM, 2014.
- [ZCLR15] Liang Zhao, Feng Chen, Chang-Tien Lu, and Naren Ramakrishnan. Spatiotemporal event forecasting in social media. In *Proceedings of the 2015 SIAM international conference on data mining*, pages 963–971. SIAM, 2015.
- [ZCLR16] Liang Zhao, Feng Chen, Chang-Tien Lu, and Naren Ramakrishnan. On-line spatial event forecasting in microblogs. *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, 2(4):15, 2016.
- [ZCWX14] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(3):38, 2014.
- [ZCY19] Liang Zhao, Feng Chen, and Yanfang Ye. Efficient learning with exponentially-many conjunctive precursors to forecast spatial events. *IEEE Transactions on Knowledge and Data Engineering*, 2019.
- [ZCYZ15] Kaiqi Zhao, Gao Cong, Quan Yuan, and Kenny Q Zhu. Sar: A sentiment-aspect-region model for user preference analysis in geo-tagged reviews. In *ICDE*, pages 675–686. IEEE, 2015.

REFERENCES

- [ZCZ15] Jia-Dong Zhang, Chi-Yin Chow, and Yu Zheng. Orec: An opinion-based point-of-interest recommendation framework. In *CIKM*, pages 1641–1650. ACM, 2015.
- [ZGT⁺18] Fan Zhou, Qiang Gao, Goce Trajcevski, Kunpeng Zhang, Ting Zhong, and Fengli Zhang. Trajectory-user linking via variational autoencoder. In *IJ-CAI*, pages 3212–3218, 2018.
- [ZJL⁺15] Jie Zhu, Wei Jiang, An Liu, Guanfeng Liu, and Lei Zhao. Time-dependent popular routes based trajectory outlier detection. In *WISE*, pages 16–30. Springer, 2015.
- [ZJL⁺17] Jie Zhu, Wei Jiang, An Liu, Guanfeng Liu, and Lei Zhao. Effective and efficient trajectory outlier detection based on time-dependent popular route. *WWWJ*, 20(1):111–134, 2017.
- [ZLL⁺17] Chao Zhang, Liyuan Liu, Dongming Lei, Quan Yuan, Honglei Zhuang, Tim Hanratty, and Jiawei Han. Triovecevent: Embedding-based online local event detection in geo-tagged tweet streams. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 595–604. ACM, 2017.
- [ZLL⁺18] Chao Zhang, Mengxiong Liu, Zhengchao Liu, Carl Yang, Luming Zhang, and Jiawei Han. Spatiotemporal activity modeling under data scarcity: A graph-regularized cross-modal embedding approach. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [ZLY⁺18] Chao Zhang, Dongming Lei, Quan Yuan, Honglei Zhuang, Lance Kaplan, Shaowen Wang, and Jiawei Han. Geoburst+: effective and real-time local event detection in geo-tagged tweet streams. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 9(3):34, 2018.
- [ZLZ⁺11] Daqing Zhang, Nan Li, Zhi-Hua Zhou, Chao Chen, Lin Sun, and Shijian Li. ibat: detecting anomalous taxi trajectories from gps traces. In *UbiComp*, pages 99–108. ACM, 2011.
- [ZN13] Jiangchuan Zheng and Lionel M Ni. Time-dependent trajectory regression on road networks via multi-task learning. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

REFERENCES

- [ZP17] Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *SIGKDD*, pages 665–674. ACM, 2017.
- [ZSM⁺18] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *ICLR*, 2018.
- [ZSY⁺15] Liang Zhao, Qian Sun, Jieping Ye, Feng Chen, Chang-Tien Lu, and Naren Ramakrishnan. Multi-task learning for spatio-temporal event forecasting. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1503–1512. ACM, 2015.
- [ZT17] Xiangyu Zhao and Jiliang Tang. Modeling temporal-spatial correlations for crime prediction. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 497–506. ACM, 2017.
- [ZXL⁺17] Pengpeng Zhao, Xiefeng Xu, Yanchi Liu, Ziting Zhou, Kai Zheng, Victor S Sheng, and Hui Xiong. Exploiting hierarchical structures for poi recommendation. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 655–664. IEEE, 2017.
- [ZYC⁺16] Liang Zhao, Jieping Ye, Feng Chen, Chang-Tien Lu, and Naren Ramakrishnan. Hierarchical incomplete multi-source feature learning for spatiotemporal event forecasting. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2085–2094. ACM, 2016.
- [ZYST19] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1):5, 2019.
- [ZZB⁺17] Xuchao Zhang, Liang Zhao, Arnold P Boedihardjo, Chang-Tien Lu, and Naren Ramakrishnan. Spatiotemporal event forecasting from incomplete hyper-local price data. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 507–516. ACM, 2017.

REFERENCES

- [ZZG⁺18] Bang Zhang, Lelin Zhang, Ting Guo, Yang Wang, and Fang Chen. Simultaneous urban region function discovery and popularity estimation via an infinite urbanization process model. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2692–2700. ACM, 2018.
- [ZZKL17] Shenglin Zhao, Tong Zhao, Irwin King, and Michael R Lyu. Geo-teaser: Geo-temporal sequential embedding rank for point-of-interest recommendation. In *Proceedings of the 26th international conference on world wide web companion*, pages 153–162. International World Wide Web Conferences Steering Committee, 2017.
- [ZZY⁺13] Kai Zheng, Yu Zheng, Nicholas J Yuan, Shuo Shang, and Xiaofang Zhou. Online discovery of gathering patterns over trajectories. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1974–1988, 2013.
- [ZZY⁺16a] Chao Zhang, Guangyu Zhou, Quan Yuan, Honglei Zhuang, Yu Zheng, Lance Kaplan, Shaowen Wang, and Jiawei Han. Geoburst: Real-time local event detection in geo-tagged tweet streams. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 513–522. ACM, 2016.
- [ZZY⁺16b] Shenglin Zhao, Tong Zhao, Haiqin Yang, Michael R Lyu, and Irwin King. Stellar: Spatial-temporal latent ranking for successive point-of-interest recommendation. In *AAAI*, 2016.
- [ZZY⁺17a] Chao Zhang, Keyang Zhang, Quan Yuan, Haoruo Peng, Yu Zheng, Tim Hanratty, Shaowen Wang, and Jiawei Han. Regions, periods, activities: Uncovering urban dynamics via cross-modal representation learning. In *Proceedings of the 26th International Conference on World Wide Web*, pages 361–370. International World Wide Web Conferences Steering Committee, 2017.
- [ZZY⁺17b] Chao Zhang, Keyang Zhang, Quan Yuan, Fangbo Tao, Luming Zhang, Tim Hanratty, and Jiawei Han. React: Online multimodal embedding for recency-aware spatiotemporal activity modeling. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 245–254. ACM, 2017.