ENHANCING GEO-SOCIAL SYSTEMS: PROFILING, RANKING AND RECOMMENDATION

A Dissertation

by

WEI NIU

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	James Caverlee
Committee Members,	Frank Shipman
	Krishna Narayanan
	Yoonsuck Choe
Head of Department,	Dilma Da Silva

December 2017

Major Subject: Computer Engineering

ABSTRACT

The global sharing of fine-grained geo-spatial footprints – via smart mobile devices and social media services (e.g., Facebook, Twitter, Flickr) – is leading to the creation of a new class of *geo-social systems*. These systems promise new insights into the dynamics of human behavior and new intelligent location-aware applications, enabled by both geographic and social characteristics of their users. Yet, we are just beginning to understand these nascent systems. Indeed, there is a significant research gap in understanding, modeling, and leveraging geo-tagged user activities and in identifying the key factors that influence the success of new geo-social systems.

Hence, this dissertation research seeks to enhance existing and future geo-social systems through a systematic study of the intrinsic mutual reinforcement relationship between geography (geo) and user behaviors (social). In particular, we focus on three important scenarios: geo-social profiling, ranking, and recommendation. In summary, this dissertation makes three unique contributions:

- The first contribution of this dissertation research lies in frameworks for profiling both users and locations in geo-social systems. We propose a framework to identify conceptual communities and a smoothing-based approach that collectively balances the information from physical neighbors and conceptual community for estimating the hashtag distribution at a particular location. We further propose a location-sensitive folksonomy construction framework and build high-quality tag profiles for users by identifying candidate tags from these location-sensitive folksonomies, and then employ a learning-to-rank approach for ordering these tags.

- The second contribution of this dissertation research is a framework for location-sensitive topical expert identification and ranking in social media. Three of the key features

of the proposed approach are: (i) a learning-based framework for integrating multiple user-based, content-based, list-based, and crowd-based factors impacting local expertise that leverages the fine-grained GPS coordinates of millions of social media users; (ii) a location-sensitive random walk that propagates crowd knowledge of a candidate's expertise; and (iii) a comprehensive controlled study over AMT-labeled local experts on eight topics and in four cities.

- In the third contribution, we create a new geo-social framework for effective personalized image recommendation as part of the effort toward enhancing geo-social systems. We propose Neural Personalized Ranking (NPR) – a personalized pairwise ranking model over implicit feedback datasets – that is inspired by Bayesian Personalized Ranking (BPR) and recent advances in neural networks. We further build an enhanced model which significantly boosts performance by augmenting the basic NPR model with multiple contextual preference clues derived from geographic features, user tags and visual factors.

DEDICATION

To my mother, my father, and Tian.

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my advisor Dr. James Caverlee, for his exceptional guidance, caring, patience, and providing me with an excellent resources and atmosphere for doing research. Back in 2012, Dr. Caverlee inspired me in the beginning with his personality and teaching and I strongly felt he is the professor to work with. During these years, he continuously fuels me with impetus, confidence and passion for research in everyday life. I'm always able to draw strength from his creative insights and wise guidance, making me less confused in envisioning the research problem and helping me avoid detours on my PhD pathway. I have also been continuously learning from his technical knowledge, critical thinking and professionalism through numerous weekly meetings, which significantly help developing my research attitude and capability. Dr. Caverlee is not only a preeminent advisor in research, but also an important role model and life mentor for me. I'm thankful for his encouragement and optimism which helped me went through my hard times and influenced me to positively deal with difficulties.

I would like to acknowledge with gratitude the love and caring from my family. Especially my parents, for their unconditional support and selfless devotion to me. Thank you for providing me the great education and environment for me to grow up. I'm grateful for my father's advice whenever I face some dilemma and for my mother's loving care in all these years. Meanwhile, I would like to thank my girlfriend Tian. Words cannot describe how fortunate and joyful I am to have her in my life. She is always there cheer me up. Very grateful for having her accompany in all these years in a foreign land.

I would like to thank my lab members – Haokai Lu, Cheng Cao, Hancheng Ge, Majid Alfifi, Zhijiao Liu, Haiping Xue, Henry Qiu, Xing Zhao, Parisa Kaghazgaran, Yin Zhang

and others who worked in the lab. Their dedication and perseverance kept motivating me in this 5-year journey. Thank you all and the best of luck in your future endeavors. Additionally I would like to thank former lab members Krishna Kymath, Zhiyuan Cheng, Kyumin Lee, Elham Kabiri and Yuan Liang. They helped me get familiar with the research areas, and showed me how to be a successful PhD student.

I'm also grateful to Dr. Yoonsuck Choe, Dr. Krishna Narayanan and Dr. Frank Shipman for generously being my committee members. Thank you for the constructive comments and feedback that help make my research work more polished and valuable. Thank for the time you devoted to me. Special thanks to Dr. Xia Hu, for attending my defense.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Professor James Caverlee and Frank Shipman, Yoonsuck Choe of the Department of Computer Science and Engineering and Professor Krishna Narayanan of the Department of Electrical and Computer Engineering.

Funding Sources

Graduate study was supported by a research assistantship from Texas A&M University.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

1.1 Motivation

In the past decade, we have witnessed tremendous growth of various social media and their ubiquity in almost every aspect of modern life. For example, Facebook has over 2 billion monthly active users measured in June 2017 [1], followed by online video sharing system Youtube which has 1.5 billion monthly active users [2]. Such popularity ensures and is accompanied by the rapid growth of the sheer volume of data generated by social media users. For example, 510,000 comments are posted, 293,000 statuses are updated, and 136,000 photos are uploaded every minute on Facebook and more than 500 million tweets are posted on Twitter per day. Increased interaction among user and content through these platforms marks a shift in the way people connect and share information.

The success of social media together with the proliferation of location-aware devices, such as smart phones and tablets, fosters a new form of media – *geo-social systems* – that allow a large portion of user-generated content contributed through social media services to be geolocated. This potentially bridges the gap between the physical world and online social media. We have seen the rise of location sharing services (e.g., Foursquare, Gowalla) that allow users to share their location with friends. For example, Foursquare has 10 billion check-ins in total as of late 2016 [3]. Moreover, geo-social systems are not confined to these location sharing services and exist more broadly, as sharing of location data coupled with various content types is increasingly popularized in existing social media. For example, geo-tagged Twitter posts, geo-tagged images on Flickr, geo-tagged status updates on Facebook, and so on.

Through the sharing of fine-grained geo-spatial footprints via smart mobile devices and social media services (e.g., Facebook, Twitter, Flickr), we are able to observe human activ-

ities in scales and resolutions that were so far unavailable. This promises new insights into the dynamics of human behavior and new intelligent location-aware applications. Massive amounts of geo-tagged user activities – including user checkins, user-contributed multimedia (like text, images, and videos), and user tagging of other users and resources – are beginning to provide new insights and new capabilities for *geo-social systems*.

These new geo-social systems target a myriad of critical problem domains by leveraging these geo-spatial footprints: examples include personalized geo-aware recommendation systems [4, 5], local authority discovery [6, 7, 8], intelligent disaster response [9, 10], urban computing [11], targeted advertising [12], and traffic prediction [13, 14]. All of which offer the potential for improvements in decision-making and quality-of-life for stakeholders of these systems. For example, in the emergency response scenario, there is hope to understand and evaluate the damage of disasters like earthquakes and hurricanes from what people have posted on social media, leading to improved response and recovery.

1.2 Challenges

However, we are just beginning to understand these nascent geo-social systems. Indeed, there is a significant research gap in understanding, modeling, and leveraging geo-tagged user activities and in identifying the key factors that influence the success of new geo-social systems. A geo-social system does not only mean adding a location to an existing social network so that people in the social structure can share location-embedded information, but also consists of new social structures made up of individuals connected by the interdependency derived from their locations in the physical world. These interdependencies, for example, include interest and behavior, as well as their location-tagged media content, such as photos and texts [15]. Indeed, we face a number of key challenges for extracting valuable knowledge from these systems and in facilitating new applications built over these systems. In this section, we highlight three specific challenges: (i) under-

standing locations via user activity and mobility; (ii) understanding users based upon their locations; and (iii) understanding locations and users jointly, so as to facilitate application such as event discovery, local expert identification and personalized recommendation.

- Understand locations via user activity and mobility. There is a lack of study for understanding and characterizing locations from conceptual perspective according to user-generated content or user mobility. Yet, there exists inherent sparsity, uncertainty in social media data, which is more severe in the collected data samples. It is challenging for us to accurately characterize location while coping with the sparsity issues.

- Understand users based upon their locations. Few existing works aim to capture spatial variation in user behavior or content they generated and how would such variations inform us with more robust user modeling under sparsity.

- Understand locations and users jointly. Despite some recent work jointly uncover geographical characteristics and user preference, for example in point-of-interest recommendation, the rigorous way to model and leverage geo-tagging metadata in a wide range of user-related application scenarios or tasks remains unclear and the mutual reinforcement relation between location and user is open to exploration. Furthermore, the potential for incorporating such geo info in impacting these applications is understudied.

1.3 Contributions

With these research challenges in mind, this dissertation research seeks to enhance existing and future geo-social systems through a systematic study of these questions on three important scenarios: geo-social profiling, ranking, and recommendation. Through the development of new frameworks that naturally integrate location signals and social

networks, we aim to enhance (i) user and location profiling; (ii) local topic expert discovery and ranking; and (iii) personalized geo-social image recommendation. Concretely, toward enhancing geo-social systems, this dissertation makes three contributions:

- The first contribution of this dissertation research lies in frameworks for profiling both users and locations in geo-social systems. Estimating these profiles is critical for location-based search and mining applications, but faces challenges in sparsity, uncertainty, and diversity across locations and times. For *location profiling*, we investigate a large geo-tagged tweets dataset with more than 2 billion tweets and 300 million hashtags. We propose a framework to identify conceptual communities – where locations share similar interest and topics are grouped together rather then directly considering the geographical proximity – through an investigation of location representation, distance measurement between locations, and a clustering-based approach for finding the conceptual communities. We propose a smoothing based approach that collectively considers the information from physical neighbors and conceptual community for estimating the hashtag distribution at a particular location, in order overcome the sparsity of geo-tagged hashtag and missing information from sampling. For *user profiling*, we explore the impact of spatial variation in social media on the construction of user profiles. This variation – e.g., that `energy` in San Francisco is more associated with the `green movement`, whereas in Houston it is more associated with `oil and gas` – is a valuable, but often understudied aspect of user profiling. Concretely, we propose a location-sensitive folksonomy construction framework that naturally integrates the geo-spatial scope of social media tags via an optimization method that generalizes from previous folksonomy induction approaches. We further build high-quality tag profiles for users by identifying candidate tags from these location-sensitive folksonomies, and then employ a learning-to-rank approach for ordering these tags.

- The second contribution of this dissertation research is a framework for location-sensitive topical expert identification and ranking in social media. Unlike general experts, local experts have specialized knowledge focused around a particular location. Identifying local experts can improve location-sensitive search, recommendation and create new possibilities for people to connect with knowledgeable locals. The framework builds on top of the state-of-the art learning to rank approach and an investigation of multiple classes of features that impact local expertise including: (i) user-based features (e.g., the number of users a candidate is following, the number of posts this candidate has made); (ii) tweet content features (e.g., tweet-based entropy of a candidate, the TFIDF score of a topic keyword in a candidate's tweets); (iii) list-based features (e.g., the number of lists the candidate is a member of, the number of lists the candidate has created); (iv) local authority features (e.g., the distance between candidate and the query location, the average distance from a candidate's labelers to the candidate); and (v) features based on a location-sensitive random walk that propagates crowd knowledge of a candidate's expertise. Through evaluation we find promising results and see that a compact set of influential features can increase the efficiency of the model. We make an encouraging discovery through experiments that a generalized model yields competitive performance with uniquely trained model of each location and topic pair. This largely increases the feasibility of developing an online local expert system based on our framework. Analytically, we provide a through discussion about the system design issues.

- Finally, in the third contribution, we create new geo-social models for effective personalized image recommendation as part of the effort toward enhancing geo-social systems. The prevalence of online image sharing services like Flickr, Pinterest, and Instagram demonstrates the increasing preference for image-based social media. As images have become more indispensable, a key challenge is in connecting users to the right images

from a vast sea of candidate images. The key challenges we are facing include uncovering the underlying dimensions, in particular *geo-social* factors, that describe the properties of the item as well as users' preferences toward them, extreme sparsity (and its corollary cold-start problem) in user feedback and implicit nature where recommendations rely on like "favorites" or "likes" instead of explicit ratings. We propose a new model toward improving the quality of image recommendations in social sharing communities. Concretely, we propose Neural Personalized Ranking (NPR) – a personalized pairwise ranking model over implicit feedback datasets – that is inspired by Bayesian Personalized Ranking (BPR) and recent advances in neural networks. We further build an enhanced model by augmenting the basic NPR model with multiple contextual preference clues derived from metadata including user tags, geographic features, and visual factors. In our experiments over the Flickr YFCC100M dataset, we demonstrate the proposed NPR model is more effective than multiple baselines. Moreover, the contextual enhanced NPR model significantly outperforms the base model by 16.6% and a contextual enhanced BPR model by 4.5% in precision and recall. We also discuss the different characteristics of the NPR and BPR based approaches with respect to varying sizes of training data.

1.4 Dissertation Overview

The remainder of this dissertation is organized as follows:

- **Chapter 2: Related Work.** We begin with a comprehensive summarization of related works corresponding to our three main contributions for enhancing geo-social systems – location and user profiling, local topic-expert discovery and personalized image recommendation.

- **Chapter 3: Profiling: Community-Based Geospatial Tag Estimation.** In this chapter, we tackle the geospatial tag estimation problem, which is of critical importance for

location-based search, retrieval, and mining applications. However, tag estimation is challenging due to massive sparsity, uncertainty in the tags actually used, as well as diversity across locations and times. Toward overcoming these challenges, we propose a community-based smoothing approach that seeks to uncover hidden conceptual communities which link multiple related locations by their common interests in addition to their proximity. Through extensive experiments over a sample of millions of geo-tagged Twitter posts, we demonstrate the effectiveness of the smoothing approach and validate the intuition that geo-locations have the tendency to share similar "ideas" in the formation of conceptual communities.

- **Chapter 4: Profiling: Location-sensitive User Profiling.** User profiles typically provide insight into the interests and expertise of each user, and can lead to improved personalization of search and recommender systems. And yet, the vast majority of users have only partial user profiles; their interests are essentially hidden from important applications. Hence in this chapter, we investigate the impact of spatial variation on the construction of location-sensitive user profiles. Our proposed approach has three unique contributions: first, it constructs a crowdsourced label similarity graph induced from crowdsourced labels, where each labeler and labelee are annotated with a geographic coordinate; second, it transforms this similarity graph into a directed weighted tree that imposes a hierarchical structure over these labels; third, it embeds this location-sensitive folksonomy into a user profile ranking algorithm that outputs a ranked list of candidate labels for a partially observed user profile. Through extensive experiments over a Twitter list dataset, we demonstrate the effectiveness of this location-sensitive user profiling.

- **Chapter 5: Ranking: Local Expert Discovery.** Local experts – who have specialized knowledge focused around a particular location – are critical for many location-sensitive

information needs. While existing search engines and question-answer systems may provide partial coverage of some local information needs, local experts can provide ongoing help for evolving and ill-specified needs, as well as personalized access to knowledge and experience that only experts possess. In this chapter, we explore a geo-spatial learning-to-rank framework for identifying local experts that leverages the fine-grained GPS coordinates of millions of Twitter users. Two of the key features of the proposed framework are: (i) a location-sensitive random walk that propagates crowd knowledge of a candidate's expertise; and (ii) a comprehensive study of the user-based, list-based, and crowd-based factors impacting location expertise. Through a controlled study over AMT, we find significant improvements of local expert finding versus two state-of-the-art alternatives.

- **Chapter 6: Recommendation: Personalized Image Recommendation.** In this chapter, we tackle the problem of personalized image recommendation in geo-social image sharing communities like Pinterest, Flickr, and Instagram. Recommendations in these communities often rely on implicit signals like "favorites" or "likes" instead of explicit ratings. And yet, these implicit signals are often sparse, with many images receiving few if any "likes" at all. We propose a new model toward improving the quality of image recommendations in social sharing communities like Pinterest, Flickr, and Instagram. Concretely, we propose *Neural Personalized Ranking (NPR)* – a new neural network based personalized pairwise ranking model for implicit feedback, which incorporates the idea of generalized matrix factorization. We further develop an enhanced model which is generalizable for incorporating with multiple contextual preference clues including user tags, geographic features, and visual factors to further boost performance.

- **Chapter 7**: We conclude with a summary of our thesis contributions and a discussion of future research extensions to the results presented here.

2. RELATED WORK

In this chapter, we highlight the related work for each of our thrusts to enhance geo-social systems, including location and user profiling, local topic expert detection and geo-social personalized image recommendation.

2.1 Location and User Profiling

Location Profiling. Recently, there has been a rise of research aimed at location-revealing social networks like Facebook and Twitter [16]. Some works have represented locations as a bag-of-tags of geo-tagged photos [17] or checkins and venue features [18]. Cranshaw et al. present several methods for comparing cities as vectors of venue categories and then using hierarchical clustering to find similar cities [19]. Spectral clustering is used to detect geospatial neighborhood-like communities or similar locations based on user activity patterns or venue features [20, 21]. In contrast, tour work uses tag distribution information to represent locations for uncovering hidden conceptual communities [22]. That is, we do not require users to necessarily check-in in one place and then another; instead, we aim to uncover locations that are tied by common topic interests at multiple granularities and potentially across physical proximity boundaries.

User and Resource Profiling. User profiling is critical for enabling effective information search and recommendation. Many efforts have been devoted to profiling a user's topic interest for applications in personalized search [23], targeted advertisement [24], and social media [25]. In a separate direction, studies have focused on expertise profiling in an enterprise domain from evidence like CV and publications [26]. Many other research efforts aim to reveal users' demographic information like age and gender [27, 28, 29]. For example, Li et al. [29] propose a co-profiling approach to profile attributes of a user like employer, college, and circles of friends. Different from the previous research, we aim to

discover the location-aware interests and expertise profile of users based on crowdsourced labels [30].

Another research thread is aimed at studying resource profiling which aims to provide rich context for the tagged object. Example tags include Flickr-like tags on images and Twitter-style tags on text posts. Much effort has been devoted to recommending or ranking tags to resources (like images or URLs), targeting the sparsity of collaborative tagging in order to construct comprehensive and complete tag profiles [31, 32, 33, 34, 35]. Some proposed approaches are based on topic models and contextual information [34, 36]. Heymann et al. [37] use relations between tags discovered through association rule mining to predict social tags. Later works construct hierarchical folksonomies to assist resource tag prediction [38, 39]. Both of these approaches rank candidate tags by measuring how each candidate tag relates to the whole set of visible tags, which often leads to the prediction of general tags instead of specific tags. Moreover, they consider tags as a binary symbol and neglect the frequencies of tags. In our work, we cast the profiling problem as a ranking task, we leverage location-sensitive folksonomy for identifying candidate tags and a learning to rank approach that automatically weighs a group of factors to minimize the personalized ranking error.

In a similar direction, hashtags can be recommended to users to encourage additional appropriate tagging; much of this work has focused on content similarity between user's tweets and an existing hashtag topic [40, 41]. In another direction, researchers have studied the overall temporal and geospatial distribution of hashtags as they diffuse [42]. For example, there has been research aimed at predicting hashtag popularity from temporal and geospatial perspectives [43]. The work presented here – where geographic patterns of hashtag use uncover conceptual communities – could inform efforts on tag recommendation and geographic tag diffusion.

Geographic Influence. The recent proliferation of location-based social networks has driven increasing attention to geographic influence, e.g., [16, 44]. Additionally, there have been studies of spatial variation over query logs and social media. For example, Backstrom et al. model how spatial variation manifests in search engine query logs [45], while Zhang et al. discovered tags with similar geographic and temporal patterns of use beyond the contextual co-occurrence [46]. The geospatial variations and influence have been incorporated into many location-based applications, for example, user location prediction [47], modeling video watching preferences [48] and Yelp ratings prediction [49]. Our perception is that due to the geographic, cultural, and structural differences among locations, there could be corresponding difference in how people organize information in these locations as revealed through crowdsourced labels.

Targeting Sparsity. There is also recent work on estimation to overcome sparse data in social networks, such as social network user home location prediction [47, 50, 51]. To alleviate data sparsity [52], for example, researchers have proposed to use transfer learning in collaborative filtering [53]. Another widely used approach in language modeling for adjusting the maximum likelihood estimator to compensate for data sparseness is smoothing [54, 55]. While in our study, we use the community distribution to smooth over the target location, where the hashtag information is unknown, to deal with data sparsity and to focus on the overall correctness of estimating the hashtag distribution.

2.2 Local Expert Detection

Expertise retrieval. Expertise retrieval has long been recognized as a key research challenge [56]. For example, for many years TREC's Enterprise Search Track has included a track for researchers to empirically assess methods for expert finding [57]. Methods proposed can generally be grouped into two categories according to the source of expertise indicators used. First, content-based methods utilize textual content and related

documents which contain terms semantically relevant to the candidates' expertise areas. Several works adopt content-based approaches to identify the most appropriate community members for answering a given question in question-answering systems, e.g., [58, 59, 60]. Al-Kouz and et al. leverage user profile and post to match topic expertise on Facebook [61]. Balog et al. proposed a candidate generative model which represent a candidate directly by terms and a document model which first finds documents that are relevant to the topic and then locates the experts associated with these documents [62]. Second, graph-based methods rely on social link analysis so as to consider each expert candidate's importance or social influence, e.g., [63, 64, 65]. For example, Campbell and et al. utilize the link between authors and receivers of emails to improve expert finding in an email-based social network [66]. Moreover, there exist hybrid models considering both textual content and social relationships in expert finding, e.g., [67, 68, 69].

Expertise retrieval in Twitter-like systems. Recently, effort has focused on expert finding in Twitter-like systems. Weng et al. consider both tweet content and link structures among users to find topic experts on Twitter [69]. Based on the list meta-data in Twitter, Ghosh and et al. built the Cognos systems to help find experts on a specific topic [70]. They rank experts by taking into account the overall popularity of a candidate and topic similarity. In the past year, a few efforts have begun to examine local aspects of expertise finding [6, 71]. Li and et al. investigate expertise in terms of a user's knowledge about a place or a class of places [71]. Cheng and et al. identified several factors, including local authority and topical authority, for assessing local expertise [6]. Note that the LocalRank method in [6] – which considers topical authority and local authority – is similar in spirit to [72], in which a rank scoring function is defined over a combination of document relevance and physical distance. Experimentally we observe that LExL outperforms LocalRank, indicating the importance of these additional models that build on what was proposed [6] and

[72]. Compared to these works, we introduce the first learning-based method for ranking local experts, introduce a new distance-biased random walk class of feature that models distance and relevance jointly (rather than independently), and conduct the first comprehensive study of factors impacting local expert ranking. Alternatively, many commercial systems provide search capabilities over regional content – e.g., Google+ Local, YellowPages.com, Craigslist – but not direct access to local experts, nor transparent models of how (or even whether) user expertise is assessed.

Learning to rank. The method in our work builds on learning-to-rank [73], which has been an active area of ranking that builds on results in machine learning. Generally, learning-to-rank can be classified into three main types: pointwise methods, in which a single score is predicted for each query document through solving a regression problem; pairwise methods, in which the relative quality of each document pair is judged in a binary classification problem; and listwise methods, in which the evaluation metric is optimized as the direct goal [74]. In the area of expert finding, Yang et al. [75] apply Ranking SVM to rank candidate experts for ArnetMiner, an academic web resource. Moreira et al. explore the use of learning-to-rank algorithms for expert search on DBLP [76].

2.3 Personalized Geo-social Image Recommendation

Research attention on recommendation has shifted towards the common scenario where only implicit feedback is available, as is common in social imaging sharing communities. One pioneer work terms such scenario as one-class collaborative filtering [77], where the authors proposed to weight positive and unobserved feedback differently in fitting the objective function. This idea was further improved to introduce varying confidence levels [78]. These approaches are mainly variations of pointwise approaches suitable for explicit feedback. In contrast to pointwise methods that consider unobserved feedback as negative to some degree, pairwise methods assume positive feedback is more preferable than

unobserved feedback. The idea is similar with pairwise learning to rank which tries to minimize the probability of incorrect pairwise orders. Pairwise learning for implicit feedback, specifically Bayesian personalized ranking with matrix factorization (BPR-MF), outperforms pointwise learning counterparts under sparse conditions [79].

Image Recommendation. Many works have tackled the problem of image recommendation, e.g., [80, 81, 82]. For example, Jing et al. use a weighted matrix factorization model that combines image importance and local community user rating [81]. Sang et al. measure the distance of an image and a personalized query language through a graph-based topic-sensitive probabilistic model [83]. On a similar track, cross domain image collection recommendation learns a query-specific distance metric for identifying similar Pinterest boards according to users' click logs [84]. Later works begin incorporating a variety of visual features, including high-level features from deep convolutional neural networks. For example, Liu et al. introduce social embedding image distance learning that learns image similarity based on social constraints and leverages Borda Count for recommendation [85]. Liu et al. augment a weighted matrix factorization model with a focus on modeling sparse latent visual topics [86]. Lei et al. propose a comparative deep learning model that learns image and user representation jointly and identifies the nearest neighbor images of each user for recommendation [87].

Location-aware and Context-aware Recommendation. To overcome ratings sparsity, many recommenders have proposed to incorporate additional contextual information [88], including but not limited to social connections [89, 90], content [91, 92], and so on. Visual features have received much attention in recent work, with some methods using metrics for visual similarity according to social behavior or activity pattern to identify compatible items [85, 93]. In [94], the authors incorporate visual features to improve product recommendation for clothing, where appearance is an important determining factor. With the

rapid growth of location-based social networks and smart mobile devices, many applications take advantage of geographical information in modeling video watching preferences [48], Yelp ratings prediction [49], and most commonly in point of interest recommendation, where representative work includes [95, 96, 97, 98, 99]. In our work, we derive and integrate multiple categories of contextual features for image recommendation. We show that our proposed method to model user's preference is effective and adaptable to different frameworks.

Deep recommendation with implicit feedback. Recently, we have seen increasing efforts devoted to recommendation models based on deep learning [87, 100, 101, 102, 103, 104, 105, 106]; note that we neglect discussion of works that leverage deep learning for deriving features then can be integrated into traditional recommendation models. Several of these target the common scenario of implicit feedback [87, 100, 101]. For example, He and et al. introduce a pointwise neural collaborative filtering framework which includes an ensemble of multi-layer perceptron and generalized matrix factorization components that jointly contribute to better performance [100]. The work that is most relevant to ours is [101], where the authors propose a multi-layer feed forward neural network based pairwise ranking model which can be applied to personalized recommendation. Distinct from previous works, we propose a pairwise ranking based recommendation model that incorporate the idea of generalized matrix factorization for implicit feedback [107]. We also provide a framework for explicitly modeling user's contextual preference for alleviating sparsity issues.

3. PROFILING: COMMUNITY-BASED GEOSPATIAL TAG ESTIMATION[1]

To accurately estimate the tag distribution for a particular place of interest from the massive geo-tagged content is a critical challenge for search, retrieval, and mining applications. In this chapter we tackle the problem of geospatial tag distribution estimation for locations.

3.1 Introduction

We have argued that the sharing of fine-grained geospatial footprints through smartphones and social media services (e.g., Facebook, Twitter) promises new insights into the dynamics of human behavior and new intelligent location-aware applications. Already, we have seen many research efforts aimed at enhancing web and social media search by integrating location signals [108, 109] and identifying "living neighborhoods" [20, 18]. Many of these scenarios are driven by social media posts that include latitude-longitude coordinates and a timestamp, often including a user-generated tag that provides additional context (e.g,. #sunset, #beach, #happy).

A critical challenge for search, retrieval, and mining applications built on these geo-tagged posts is accurate identification of popular tags and estimation of the tag distribution for a particular place of interest. For example, a real-time local search system may want to reflect what is currently "hot" in the area of interest (e.g., reaction to a political debate). Similarly, a geo-enabled advertising system may place targeted social media ads based on local interests (e.g., advertising jerseys during a football match). Unfortunately many locations may have no (or little) evidence of tags due to the sparsity of geo-enabled social media. Indeed, recent estimates suggest that only 1.5% of tweets have geo-coordinates

[1] ©2016 IEEE. Reprinted, with permission, from W. Niu and et al., "Community-based geospatial tag estimation", in Advances in Social Networks Analysis and Mining(ASONAM), 2016.

[110]. And even for locations that are well-represented, there is inherent uncertainty around using social media to capture the overall distribution of interests in a particular location. Moreover, tags are diverse from location to location and can temporally fluctuate in popularity.

Toward overcoming these challenges, this chapter tackles the *geospatial tag distribution estimation problem* for accurately estimating the tags in a particular location. The main intuition of the proposed approach is to "smooth" a location's tag distribution from a larger "conceptual" community in which a particular location belongs to, as well as geographically contiguous neighbor locations. Evidence of *homophily* in social networks [111] – wherein individuals tend to associate and bond with similar others – motivates our hypothesis that geo-locations (as comprised by individuals) may also share a similar tendency in the "ideas" (or hashtags) that are shared. Thus, it may be possible to rely on the denser (and richer) information from a larger community to shed light on individual locations, potentially alleviating the challenges of tag estimation. Concretely, we propose a community discovery framework and an approach for estimating the tags in a particular location. Through this framework, we investigate:

- The representation of locations for community discovery – both through the distribution of hashtags adopted and by how rapidly they are adopted;
- Methods for measuring the conceptual distance between locations – by two content-based methods and an adoption time metric, plus variations integrating physical distance;
- Methods for estimating the unknown hashtag distribution – through a novel community and neighbor-based smoothing method.

Through extensive experiments based on 100s of millions of geo-tagged Twitter posts,

we demonstrate the effectiveness of conceptual community-based smoothing. We investigate the impact on precision and recall of multiple estimators, examine the impact of the number of communities on tag distribution estimation, and find an improvement of the proposed approach versus a neighbor-based baseline. These findings are encouraging for improving the quality and coverage of geospatial tag estimation, which is an important step toward providing intelligent location-aware search and recommendation systems.

3.2 Geospatial Tag Distribution Estimation

We begin by assuming that we have a collection of social media posts that include latitude-longitude coordinates, a timestamp and one or more *tags* (or *hashtags*; in the following we shall use hashtags). We view a tag on a post as a tuple (h_i, t_i, g_i), where $h_i \in \mathcal{H}$ is the hashtag, t_i is the timestamp, and g_i is a latitude-longitude coordinate. We assume there is a mapping function that converts latitude-longitude coordinates into *locations*. A location here could correspond to a particular venue (e.g., a restaurant), a city block, equal or variable-sized grid cells, or some other domain-specific collection of latitude-longitude coordinates. We denote $\mathcal{L}$ as the set of all possible locations and $\mathcal{H}$ as the set of all distinct hashtags. Hence, we finally view a tagging action as the hashtag text, the timestamp, and its location: (h_i, t_i, l_i). In total, we call this collection of tuples $\mathcal{P}$ (for posts).

The *hashtag distribution at a location l* is the probability distribution of all the hashtag occurrences at that location, which we denote as θ_l:

$$\theta_l = \{[h_1, p(l, h_1)], [h_2, p(l, h_2)], \cdots, [h_m, p(l, h_m)]\}$$

where $p(l, h_i)$ is the probability of a hashtag h_i in location l. In many scenarios, this distribution is unknown or incomplete due to data sparsity, uncertainty, and other factors. Hence, the *hashtag distribution estimation problem* is to find an estimated hashtag distribution $\widetilde{\theta}_l$ that matches the actual (but, unknown) distribution θ_l as closely as possible

(where measures of closeness are defined in the experiments).

We propose to tackle the hashtag distribution estimation problem by considering multiple sources of evidence – the conceptual community which the location is associated with and the contiguous neighboring areas around a location – and by smoothing a location's tag distribution by incorporating evidence of both the conceptual community in which a particular location belongs to as well as neighboring locations. The key intuition of community-based smoothing estimation is that if a hashtag appears in a community c, which is a group of locations drawn from $\mathcal{L}$, then it is likely to appear in each individual location l_i, where $l_i \in c$.

In the following we address two key challenges to robust hashtag distribution estimation: (1) How do we identify these conceptual communities in the first place? and (2) Given a set of communities, how do we estimate the hashtag distribution for a particular location?

3.2.1 Challenge 1: Finding Conceptual Communities

A *conceptual community* links multiple related locations by their common interests rather than by their geographic proximity (that is, without the constraint of being geographically contiguous). We define the *hidden conceptual community discovery problem* as: given a collection $\mathcal{P}$ of social media posts, identify the set of conceptual communities $\mathcal{C} = \{c_1, c_2, \cdots, c_k\}$, where each community $c_i \in \mathcal{C}$ is a group of locations drawn from $\mathcal{L}$. Toward discovering these conceptual communities, we investigate in this chapter a clustering-based approach. In our preliminary research, we have tested several clustering algorithms – including spectral clustering [112], hierarchical clustering [113], Affinity Propagation [114], etc. We find that while there are qualitative differences in the clusters generated, that the k-means++ algorithm [115] provides a nice balance of efficiency and cluster quality (as measured in the experiments section). Hence, we focus our discussion

here on k-means++, which operates by finding partitions of n locations into k clusters (corresponding to our hidden conceptual communities). It provides a way for choosing the initial centers for k-means instead of random assignment which often lead to poor performance. Some clustering approaches have an embedded method for determining k, while in our case, we prefer a flexible k value since the number of communities is impacted by the performance of tag distribution estimation.

In the following, we tackle two important questions – (i) what is the appropriate representation for a location as a basic unit of community discovery? and (ii) how can we compare two locations, to determine a meaningful notion of conceptual distance for use in identifying related locations?

Representing a Location. We consider two methods for representing a location: by hashtag frequency and by hashtag adoption time.

By Hashtag Frequency. This approach captures the overall distribution of "ideas" associated with each location by considering the relative frequency distribution of hashtags observed at that location. Formally, we can represent each location l by its hashtag frequency :

$$l = \{[h_1, \mathcal{F}(l, h_1)], [h_2, \mathcal{F}(l, h_2)], \cdots, [h_n, \mathcal{F}(l, h_n)]\}$$

where $\mathcal{F}(l, h_n)$ represent the number of occurence, or we say frequency of hashtag h_n at location l.

By Hashtag Adoption Time. While the previous approach may meaningfully capture the overall interests of the location, it ignores the timing information embedded in the timestamps of each hashtag. Hence, this approach intended to capture these differences by considering the first occurrence time of each hashtag h at location l, $\tau(l, h)$ and a location

can then be represented as:

$$l = \{[h_1, \tau(l, h_1)], [h_2, \tau(l, h_2)], \cdots, [h_n, \tau(l, h_n)]\}$$

Unlike hashtag frequency, the adoption time implicitly captures the *influence* of a location, which is helpful for distinguishing between locations with similar topics but different relative importance (as measured by adoption time of ideas).

Measuring Distance Between Locations. Most existing neighborhood finding approaches require nearby (geographic-distance constrained) locations [18], so that the resulting regions are spatially bounded in a single contiguous region. In many applications this is a reasonable assumption, since contiguous regions provide the basis of many real-world ways in which we organize space (e.g., state boundaries, congressional districts). Alternatively, we explore in this section methods for linking locations based on the conceptual distance between them, so that related locations may be connected regardless of geographic distance.

First we consider *content similarity* to characterize the location similarity. We apply well-known Jaccard similarity $\mathcal{S}im_{Jaccard}$ and cosine similarity $\mathcal{S}im_{cos}$ over each pair of locations vectors represented with hashtag frequency. The corresponding distance $\mathcal{D}_{Jaccard}$ and $\mathcal{D}_{cos}$ is simply defined as the inverse of the similarity. Additionally, we propose *temporal distance*, which based on the adoption time of hashtags at different locations. We can measure the adoption time difference between two locations l_1 and l_2 as:

$$T_a = \frac{1}{\|H(l_1) \cap H(l_2)\|} \sum_{h_i \in H(l_1) \cap H(l_2)} \tau_i^{l_1} - \tau_i^{l_2}$$

which can be considered a measure of relative influence between a pair of locations. τ_i^lrepresent the first occurrence time of hashtag h_i at location l. If we take absolute value

over time difference $\tau_i^{l_1} - \tau_i^{l_2}$ in the above formula, then it measures the average hashtag adoption time between a pair of locations. We denote it as $\mathcal{D}_t$. When $\mathcal{D}_t$ is small for two locations, they are more closely related to each other, otherwise, we consider they are far apart. Note however, the adoption time approach does have the drawback of only considering pairwise influence, and so it may miss cases in which a third location is influencing the two locations of interest.

Integrating Geographical Distance. Finally, we augment each of the baseline distance metrics with a geographic distance-based damping factor. The idea is that we can provide a tunable parameter for biasing the conceptual distance between two locations by additionally considering the geographic distance between them. There are mainly two reasons. First, our intuition is drawn from the first law of geography, which states that "Everything is related to everything else, but near things are more related than distant things" [116]. Second, we aim to mitigate the impact of data sparsity on clustering. For example, if we only collected one hashtag "sports" at a particular location, then this location by content similarity, may be clustered with locations that have high frequency of "sports". To avoid degenerate clusters like that, we incorporate a factor damping with distance. We begin by considering the Haversine distance between two locations – where the Haversine distance D_H [117] measures the shortest distance over the earth's surface between two points. We update each of the distance metrics by incorporating this Haversine distance, as shown in Table 3.1. In each case, α is a user-defined distance decay coefficient. Larger values of α will increase the distance between geographically far apart locations, making them less likely to be grouped together.

3.2.2 Challenge 2: Hashtag Distribution Estimation

Given a community c and a target location $l \in c$, we further investigate how to estimate the hashtag distribution for location l. In this section, we consider three approaches:

Distance	$\mathcal{D}_{\mathbf{Jaccard}}{}^*(l_1, l_2)$	$\mathcal{D}_{\mathbf{cos}}{}^*(l_1, l_2)$	$\mathcal{D}_{\mathbf{t}}{}^*(l_1, l_2)$
	$\frac{1}{Sim_{Jaccard}}\alpha^{D_H(l_1,l_2)}$	$\frac{1}{Sim_{cos}}\alpha^{D_H(l_1,l_2)}$	$D_t\alpha^{D_H(l_1,l_2)}$

Table 3.1: Distance metrics that integrate geographical distance

(i) the first considers the conceptual community (the idea-based neighborhood) around a location; (ii) the second considers immediate geographic neighbors (ignoring the conceptual distance of these neighbors, as well as the potential conceptual closeness of more distant neighbors); and (iii) a hybrid approach which seeks to balance both conceptual and geographic distance.

Community-Based Estimation. In this first approach, we estimate the hashtag probability at location l according to the hashtag probability at the conceptual community that l belongs to:

$$\widetilde{p}(l, h) = \mathcal{F}(\mathcal{C}, h) / \sum\nolimits_i \mathcal{F}(\mathcal{C}, h_i)$$

where $\mathcal{F}(\mathcal{C}, h)$ is the frequency of hashtag h in the community $\mathcal{C}$ that l belongs to. The intuition is the hashtag distribution over the community tends to be coherent from location to location. Thus it emphasizes the overall hashtag popularity in the community, resulting in hashtags with high frequency being good candidates for the target location.

Neighbor-Based Estimation. In the second approach, we ignore the conceptual closeness of distant communities in favor of considering neighboring locations as sources of similar hashtag evidence. This method estimates the probability of a hashtag at location l according to the aggregated hashtag distribution of neighboring locations that border with the target location l. The expectation is that locations contiguous with target location share a large portion of the hashtag as what people see and experience tend to be similar. That is, we have:

$$\widetilde{p}(l, h) = \mathcal{F}(\mathcal{N}, h) / \sum\nolimits_i \mathcal{F}(\mathcal{N}, h_i)$$

Where $\mathcal{N}$ represents the locations that border with l.

Hybrid Approach. Finally, we propose a smoothing approach that seeks to balance both community-based and neighbor-based estimation. The intuition is that neither approach in isolation is best for estimating the hashtags of a location; rather, we should adopt a more flexible model to integrate both sources of evidence that can vary from location to location.

Building on the previous two estimations approaches, the hybrid approach is:

$$\widetilde{p}(l,h) = \beta \cdot p(l_{community}, h) + (1-\beta) \cdot p(l_{neighbors}, h)$$

where β is a weight equals to $N_d(h)/N_{max}$. $N_d(h)$ is the total number of distinct hashtags of target location in the dataset used for community discovery – and N_{max} – the maximum number of distinct hashtags one location has in the same dataset. When $N_d(h)$ is large, neighboring locations may only share part of the hashtags with the target location due to the difference in hashtag density. Meanwhile, the hashtag in the target location is more likely to be prevalent in the community, since we expect the community contains some similar locations of the same scale. We can rely more on community distribution as a source of supplementary and richer information for better estimation. For example, for a location like New York City, the aggregated hashtag distribution of nearby locations is not sufficient to represent the hashtag distribution of the urban area. Alternatively, the community which also contains Boston and D.C. may provide more accurate hashtag information. When $N_d(h)$ is small, the target location is less influential in the community and meanwhile, it tends to be well represented by the neighboring locations. Thus more weight should be placed on neighboring distributions. To conclude, we increase β when $N_d(h)$ is large and decrease β as $N_d(h)$ is small.

3.3 Experimental Evaluation

In this section, we present a set of experiments to show the discovered conceptual communities over different granularities and investigate the strengths and weaknesses of hashtag distribution estimation.

3.3.1 Dataset

All of our experiments are over a collection of geo-tagged social media posts sampled from Twitter. We collected 324 million hashtags from tweets that were annotated with a latitude-longitude coordinate over the course of two years (from February 2011 to March 2013). We crawled the dataset using the Twitter streaming API using a bounding box to only gather tweets from particular parts of the world. We use the Universal Transverse Mercator (UTM) [118] geographic coordinate system to grid the area of interest into homolographic subareas, each of which corresponds to a location.

3.3.2 Evaluating Hashtag Distribution Estimation

To evaluate the quality of an estimated hashtag distribution, we consider weighted versions of precision, recall, and Jensen-Shannon Divergence [119].

Weighted Precision. We consider a weighted version of precision which considers the probability of a hashtag:

$$\mathcal{P}@n = \frac{\sum_n r(h_n)\widetilde{p}(l, h_n)}{\sum_n \widetilde{p}(l, h_n)}$$

where $\widetilde{p}(l, h)$ is the probability of occurrence of hashtag h in the estimated distribution, and $r(h)$ is an indicator variable that is 1 when the estimated distribution contains h, and 0 otherwise. $\mathcal{P}@n$ is the percentage of first n estimated hashtags that actually exist at target location. These hashtags are weighted by their probability in the estimated distribution. It is a measurement of relatedness of estimated hashtags. However, it only considers the estimated distribution and ignores the actual distribution.

Weighted Recall. The weighted recall is defined as:

$$\mathcal{R}@n = \frac{\sum_n r(h_n)\widetilde{p}(l, h_n)}{\sum_m \widetilde{p}(l, h_m^*)}$$

where h^* represents all the hashtags in the actual distribution. $\mathcal{R}@n$ is the percentage of the actual hashtags that is covered by the estimated distribution. Hashtags are weighted by the probability in the estimated distribution. It is a measurement of how completely the estimated hashtags match the actual hashtags.

We use $P@n$ and $R@n$ as preliminary performance metrics and for the real distribution matching, we rely on Jensen-Shannon Divergence.

Jensen-Shannon Divergence (JSD): JSD measures the similarity between two distributions. The JSD between two hashtag distribution is defined as:

$$\begin{aligned} JSD(\theta_m, \theta_n) = &\frac{1}{2}\sum_i \ln(\frac{p(m, h_i)}{\bar{p}(h_i)}) \cdot p(m, h_i) \\ &+\frac{1}{2}\sum_j \ln(\frac{p(n, h_j)}{\bar{p}(h_j)}) \cdot p(n, h_j) \end{aligned}$$

where $p(m, h)$ represents the probability of hashtag h in distribution m and $\bar{p}(h) = \frac{1}{2}(p(m, h)+ p(n, h))$. Smaller JSD values indicate the two distributions are more similar.

3.3.3 Conceptual Community Discovery

Here, we show some example patterns of the discovered communities. Since the community discovery is randomized, we present a typical output for each target area. We postpone the discussion of picking the number of communities.

We considered four different geo-spatial granularity with respect to the different areas. The detailed statistics are listed in Table 3.2, for example, at a global-scale, we have tested

a location with an accuracy of 100 km which correspond to an area of approximately $10^4 km^2$. The total number of hashtags being considered in this case is 324 million, and the number of distinct hashtags is 298,000.

Target area	**World**	**US**	**NYC**	**MHTN**
Area (km^2)	10^4	2500	1	10^{-2}
No. of locations	$2,565$	2,482	1,445	$5,529$
Hashtag frequency	$324m$	$29m$	$3m$	$0.8m$
Distinct hashtags	$298k$	$98k$	$45k$	$7k$

Table 3.2: Four social media collections

Figure 3.1 shows us the 13 communities in the world using $\mathcal{D}_{cos^*}$. We observe a language-based pattern: for example, Brazil and Portugal are in the same community due to the common language. Overall, we observe that language (and culture) is the dominating factor for identifying communities based on content similarity.

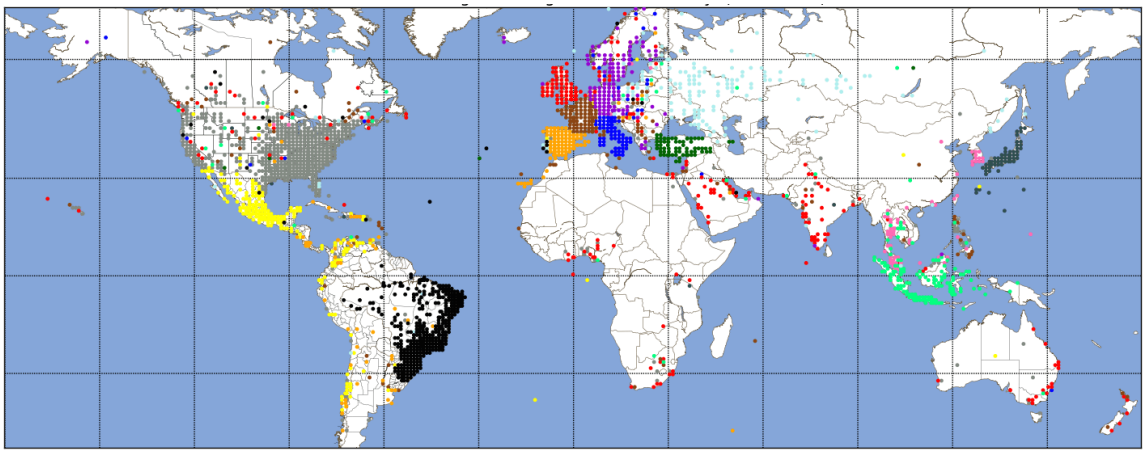


Figure 3.1: World by Content Similarity (Cosine)

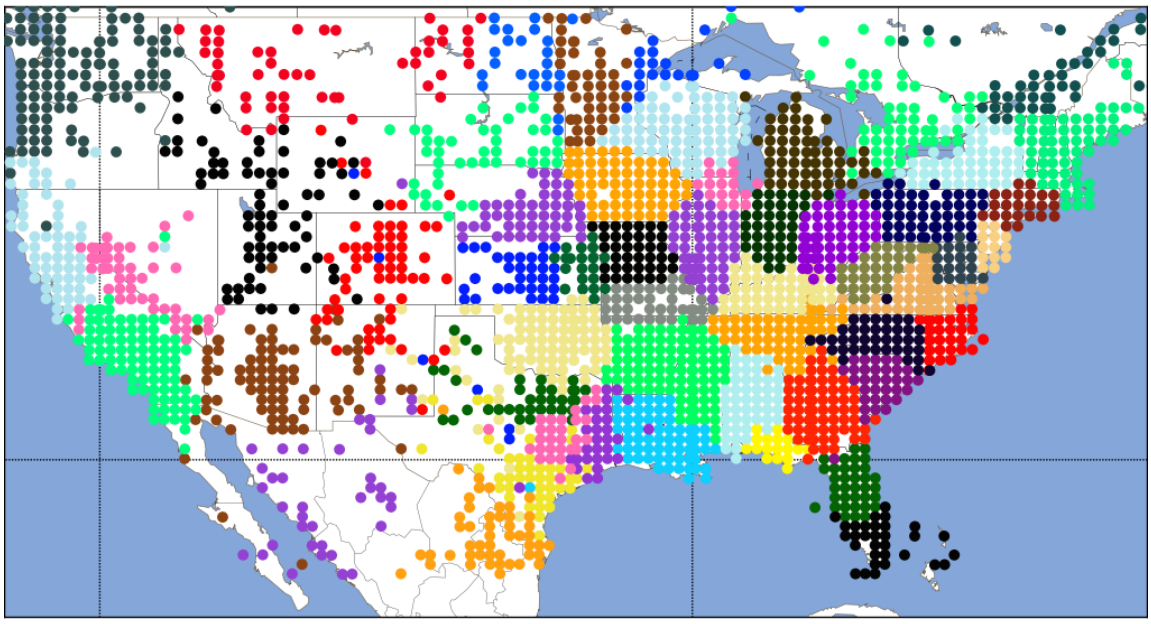


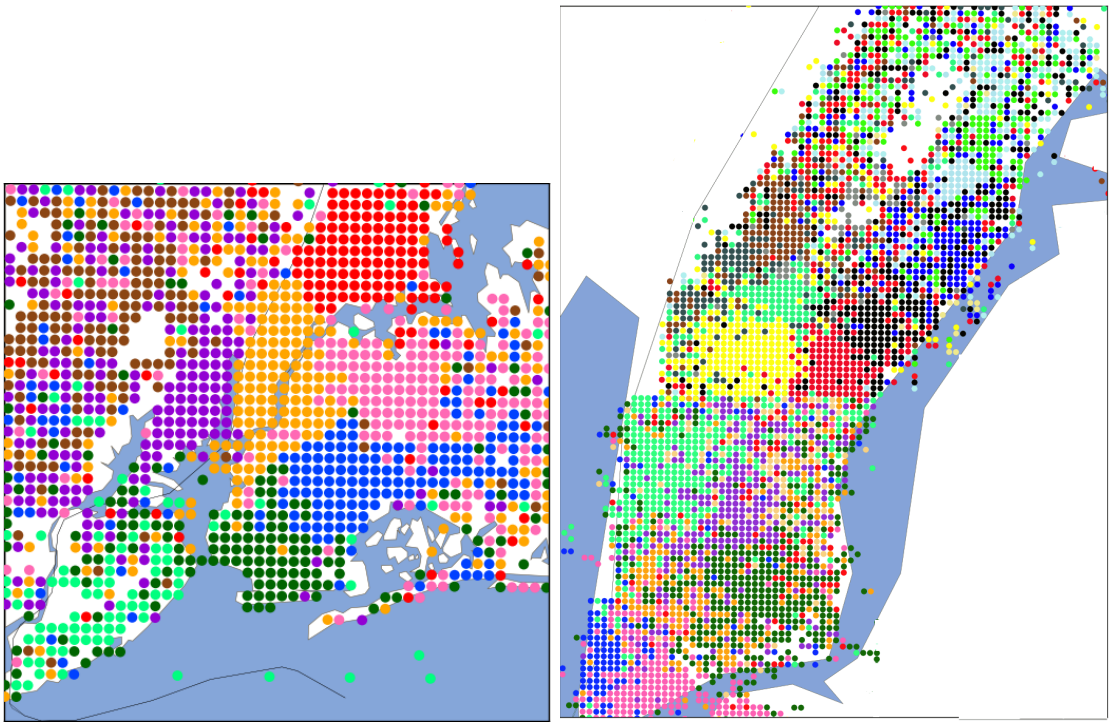
Figure 3.2: US by Content Similarity (Cosine)

Moving to a more focused context, we next consider the communities discovered in the continental US with $\mathcal{D}_{cos^*}$ in Figure 3.2. The 60 communities closely match state contours and metropolitan areas: we can see Chicago area, New York and New Jersey area, Rocky Mountains and many more. In some cases, communities span borders: for example, Arizona and part of New Mexico are in the same community. These results indicate the strength of political boundaries on the "ideas" shared via social media, supporting the argument that culture is strongly impacted by political organization.

We next examine New York City. Here, Figure 3.3a show the discovered 8 communities in New York City by $\mathcal{D}_{cos^*}$. We see a clear delineation of boroughs and neighborhoods, where the contour is consistent with the race and ethnicity map of NYC according to 2010 Census data[2].

Finally, we turn to the most narrowly scoped target area: Manhattan. We again look at

[2]https://www.flickr.com/photos/walkingsf/5559914315/



(a) NYC (b) Manhattan

Figure 3.3: NYC and Manhattan by Content Similarity(Cosine)

cosine-based approaches. Figures 3.3b show the 22 communities discovered in Manhattan. In this figure, we can notice the neighborhood-level communities. As our data at this scale is very sparse, we notice a more scattered community pattern. These community covers blocks where people may hear the same news, see similar events, and share a similar lifestyle.

In summary, we find that locations at different granularities can be grouped into coherent communities based on the approaches defined above. At the world level, language is the dominating factor that influences community discovery. At country level, we see that the communities we find are influenced mostly by the cultural identity, as well as geographic reasons. At more fine-grained levels, we attribute the different communities to demographics and everyday activities. We see that Jaccard and cosine tend to discover

similar communities, whereas adoption time provides a different flavor of finding communities. It is challenging to make a direct and fair comparison with existing neighborhood detection work mentioned in Section II due to the feature and goal differences. Thus we will further demonstrate the quality of these communities via the geo-spatial tag distribution estimation task in the following.

3.3.4 Evaluating the Community-based Approach

We now turn to investigating the quality of conceptual community-based tag estimation, before turning in the following section to comparing the community-based estimator versus a neighbor-based one and a hybrid method. We evaluate the quality of hashtag distribution estimation over the 2,482 US locations based on three proposed community discovery approaches – by Jaccard, by cosine, by temporal distance. As a baseline, we consider a physical community that less than 200 miles in Haversine distance (HD) from target location; that is, the communities are necessarily linked only by distance and not by any tag-related information. For the conceptual community-based estimators, we find initial communities based on 50% of the dataset, then use the estimation methods to infer the estimated distribution for each location with the rest of the data, which is then compared with the actual distribution of the corresponding location for evaluation. This two-fold cross validation for distribution estimation is different from classification, in that we try to keep a balanced partition and avoid estimation data being too sparse, which won't reflect the true estimation performance. We consider all locations and report averages. In our initial experiments we consider $k = 90$ communities and then test the impact of varying k.

Interestingly, we see the precision and recall for the four approaches in Figure 3.4. The x-axis in all cases corresponds to the number of hashtags being considered, ranked by decreasing occurrence probability. In all cases, we see that the two content-based community

discovery methods – Jaccard and cosine – outperform the temporal distance. We see that HD results in the best precision and recall, followed by cosine, then Jaccard, and finally temporal distance. These results indicate the strong locality effects of hashtag adoption as postulated by Tobler's first law of geography – in that communities composed of nearby locations may share more common "ideas" than those composed of distant locations. But can the conceptual communities complement this strong locality impact?

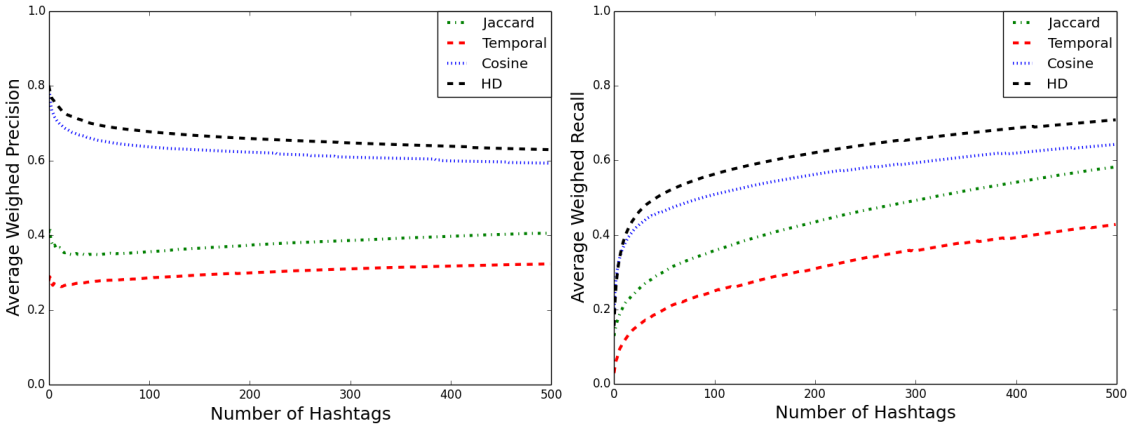


Figure 3.4: Average Weighted Precision and Recall

Integrating Haversine Distance. Next, we consider the impact on hashtag distribution estimation when we integrate Haversine distance into the conceptual distance; does forcing communities to be more compact improve the quality of hashtag estimation? Following experimental results presented in [120], we adopt a distance decaying coefficient $\alpha = 1.01$. In Figure 3.5, we report the precision and recall when the communities are discovered using the Haversine-integrated approach. We do see an improvement on the absolute performance: for example, the increase in precision and recall are especially apparent for Jaccard ($\mathcal{D}_j$) and the temporal distance ($\mathcal{D}_t$). Overall, we find using the Cosine+HD approach ($\mathcal{D}^*_{cos}$) yields the best precision and recall. So yes, integrating the strong locality of

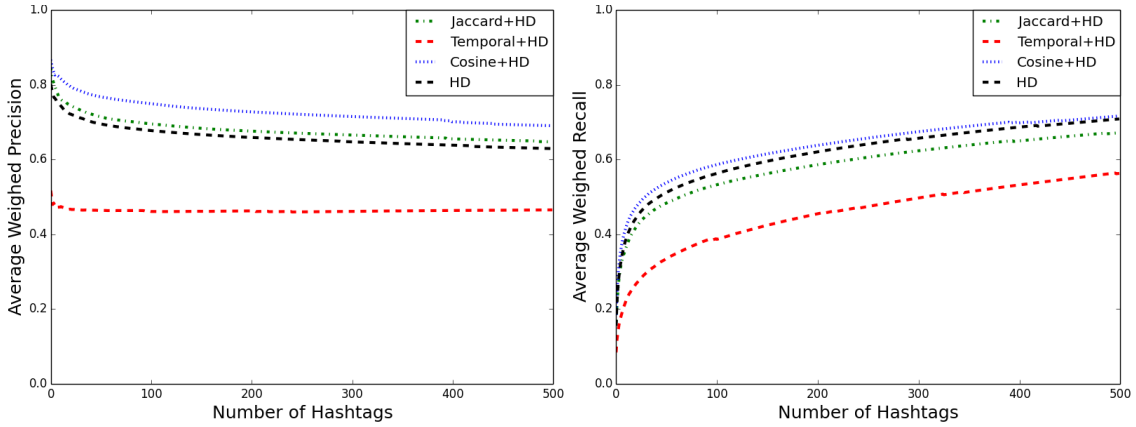


Figure 3.5: Average Weighted Precision and Recall (Integrate Haversine Distance)

nearby locations with more distant conceptual communities can positively impact hashtag distribution estimation. These results confirm the importance of carefully identifying these distant conceptual communities, and of integrating them into more naive distance-based approaches.

Varying the Number of Communities. So far, we have considered a fixed number of communities. But what impact does varying the number of communities have on the hashtag estimation problem? Indeed, there are existing methods that aim to find a proper number of clusters k for k-means [121]. But here, we would like to consider an application driven strategy to decide the number of communities, i.e. by the performance of estimation. Many small communities may favor precision in the hashtags that locations share, but without the overall perspective (and the additional hashtags) that may be present in a few larger communities that are composed of many locations. Conversely, when the number of communities is small, the hashtags shared by its constituent locations may be overly broad, resulting in poor estimation for a specific target location. Hence, in this experiment we vary the number of communities from 30 up to 240 using the Cosine+HD metric ($\mathcal{D}^*_{cos}$). Figure 3.6 presents the F1 score to characterize the performance: for ex-

ample, in the 120 communities case, which has relatively smaller community size, the precision and recall are better than the cases of 30 or 60 communities. So, more smaller communities tend to result in better estimation. However, as the number of communities increases (and the size of each community decreases), we see that the quality of estimation actually degrades. For example, at 240 communities there is worse precision and recall than in the 120 and 180 community case. As the community size shrinks, the hashtags in each community becomes more fragmentary toward representing the target location, thus dragging the precision and overall performance down. These results demonstrate the importance of incorporating location (and domain) specific models of what constitutes a community, so as to balance the finer granularity of small communities with the richness inherent in larger ones.

We further measure the distributional similarity between the estimated hashtag distribution and the actual distribution using Jensen-Shannon divergence in order to compare various distance metrics used for community discovery. For each location l in the community c, we compare the distribution of hashtag at l with the distribution of equal number of hashtag suggested by the community distribution, and then we calculate the corresponding JSD. Again in Table 3.3, we see that the Temporal+HD approach results in the highest divergence (and so, worst performance in terms of estimating the actual distribution). In all cases, we see that the Cosine+HD approach ($\mathcal{D}^*_{cos}$) results in the smallest divergence. This result demonstrates that Cosine+HD is the most effective distance metric, again indicating the strength of content-based similarity versus temporal-based similarity of locations.

3.3.5 Comparing Three Estimation Approaches

Next we compare the performance of the three proposed approaches for estimating the hashtag distribution of the target location. For community-based (CB) estimation, we adopt Cosine+HD combination as it was shown to give the best estimation in all

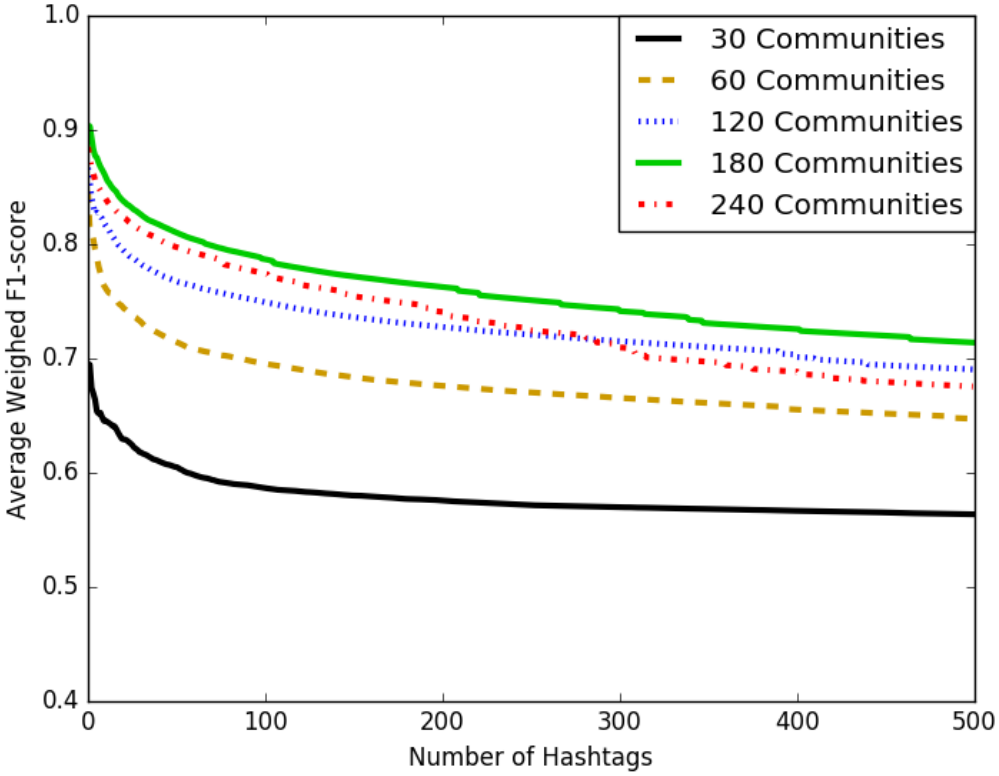


Figure 3.6: Varying the Number of Communities

community-based variations. For the Neighbor-based (NB) approach, we consider two alternatives: only contiguous neighbors of the target location and all locations that are within 200 miles from the target location. For the hybrid approach, we also compare two alternatives: a version which has constant β value and a version with varying β as we defined previously. We report the average for locations that have more than 500 distinct hashtags.

The JSDs of the estimation methods are shown in Table 3.4. We observe two hybrid approaches, with JSD values 0.33 and 0.34, generally perform better than the strictly community-based approach and the neighbor-based approach. On the one hand, neighbor-

Distance metrics	**10%**	**20%**	**50%**	**100%**
Cosine + HD	0.632	0.581	0.494	0.353
Jaccard + HD	0.670	0.632	0.551	0.380
Temporal + HD	0.676	0.648	0.589	0.453

Table 3.3: Average Jensen-Shannon Divergence @P% for Community-based Estimation

Distance metrics	**10%**	**20%**	**50%**	**100%**
CB	0.632	0.581	0.494	0.353
NB	0.628	0.583	0.489	0.386
NB (within 200 miles)	0.645	0.603	0.511	0.400
Hybrid ($\beta = 0.5$)	0.626	0.575	0.466	0.340
Hybrid	0.628	0.576	0.463	0.330

Table 3.4: Average Jensen-Shannon Divergence @P%

based estimation approach tends to identify hashtags that actually exist in a target location, however the popularity of these hashtags is unclear: they might either be popular, or be very local that only appear in a few locations. On the other hand, the community-based approach tends to identify overall popular hashtags that are possibly popular in target location as well. A combination of these two distributions effectively balances the hashtags discovered – leading to an increase in the accuracy of the estimated distribution.

We also observe that varying β is better than a constant β, as we adjust the weight according to how influential the target location is. For a location with a large number of distinct hashtags, hashtags of neighbor locations tend to be sparse compared to the target and thus is prone to be missing and incomplete in representing the target location. By increasing the weight of the community distribution, we increase the probability of seeing the popular hashtags in the community, which are also likely to be popular in the target location. For a location with a small number of distinct hashtags, neighbor hashtags tend

to be more precise than community hashtags, as the target location is not influential in the community and may share limited number of hashtags with the community. Increasing the weight to the neighbor distribution works better.

3.4 Summary

In this chapter, we have proposed and evaluated a community-based framework for tackling the problem of geospatial tag distribution estimation, which is a key component of many new location-augmented search, retrieval, and mining applications. We have investigated two ways to represent locations. Additionally, we have compared three approaches for capturing the hidden conceptual distance between locations, and evaluated three smoothing strategies. Through experimental investigation, we found that geo-locations have a tendency of sharing similar “ideas" and forming geo-spatial communities. Meanwhile, we demonstrated how our community discovering approach and smoothing strategy leads to high-quality hashtag distribution estimation. In our future work, we are interested to study geospatial community formation in alternative social media platforms (e.g., Pinterest) and to incorporate alternative signals of community formation, including activity patterns, temporal changes of idea flow, and topic-sensitive signals (e.g., considering only political hashtags).

4. PROFILING: LOCATION-SENSITIVE USER PROFILING[1]

In the previous chapter, we focused on location profiling based on user-generated hashtags, corresponding to our effort to understand locations based on user behaviors. In this chapter, we aim to unveil its counterpart relation of understanding how geographic signals help to characterize users. We develop a location-sensitive user profiling framework for more accurate characterization of user expertise with location information.

4.1 Introduction

User profiles are a valuable component of many applications, including recommender systems, search engines, question-answering systems, and online social networks. These profiles provide insight into the interests and expertise of each user, and can lead to improved personalization of the underlying system [69, 122, 123]. Many systems rely on an *explicit* definition of a user profile – for example, by filling in an "About" section in a social media profile or by directly selecting topics of interest on a question-answer system. Alternatively, *implicit* user profiles can be uncovered through methods like query log mining, running Latent Dirichlet Allocation (LDA) over a user's posts, or by applying matrix factorization approaches to identify hidden (or latent) topics of interest [124, 125, 126, 127]. In a complementary direction, recent years have seen the development of *crowdsourced* methods to build user profiles, e.g., [6, 70, 128, 129]. In this approach, crowds of users apply descriptive labels on other users, so that in the aggregate these labels provide a crowdsourced user profile of the target user. For example, Twitter Lists and LinkedIn's Skill Tags provide a partial perspective on what users are known for by aggregating crowd la-

[1]Part of this chapter is reprinted with permission from "Location-Sensitive User Profiling Using Crowdsourced Labels" by Wei Niu, James Caverlee and Haokai Lu. 2018. The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18). Copyright 2018 by Association for the Advancement of Artificial Intelligence.

beling knowledge. However, the vast majority of users have no crowd labels; their interests are essentially hidden from important applications such as personalized recommendation, community detection, and expert mining.

Our goal is to extend the reach of these crowdsourced methods, so that we can construct robust user profiles for the long-tail of users for whom we have incomplete labels. A natural approach to extend the reach of these crowd-generated labels is to apply existing *tag recommendation* methods [33, 34, 35, 37, 130]. Typically created for domains like image sharing communities (where many users have tagged different photos) or social bookmarking sites (where individuals have annotated a personal collection of URLs with a set of keywords), tag recommendation seeks to uncover relationships like `energy` $\rightarrow$ `oil and gas`. However, many of these previous approaches have viewed tag relationships without regard for the local variations that are inherent in real-world crowdsourced labels of users. For example, we find in a sample of Twitter Lists that the label `energy` in San Francisco is more associated with the `green movement`, whereas in Houston it is more associated with `oil and gas`. These spatial variations are a critical component of crowdsourced labels and require careful consideration: beyond just the presence of different relationships across locations, there is often a variation in the strength of this relationship from location to location. For example, `stock` and `finance` are more closely related in New York City than in Portland. Further, there is even a potential for varying location-specific senses of a tag (polysemy). For example, the tag `rockets` in the Houston area may be associated with the local NBA team instead of other senses of the word.

Hence, we explore the impact of spatial variation on the construction of *location-sensitive user profiles*. Our main intuition is that spatial variation over crowdsourced labels can be modeled in a location-sensitive folksonomy to provide a comprehensive and up-to-date picture of location-aware topics, topic relations, and a fine-grained topic level view of

the social media corpus, which may mitigate the sparsity inherent in the raw labels. Recent studies [131, 132] have shown how hierarchical topic structures can improve ranking and recommendation, indicating the importance of folksonomies. Hence, we aim to study the impact of location-sensitive hierarchical structures of crowdsourced tags on user profiling.

Concretely, we first demonstrate evidence of spatial variation over a collection of Twitter Lists, wherein we find that crowdsourced labels are constrained by distance, and that labels themselves and relation between pair of labels are not uniformly distributed across locations. These observations motivate our proposed approach for *location-sensitive user profiling* as illustrated in Figure 4.1 to construct robust user profiles for the long-tail of users for whom we have incomplete labels. First, we construct a crowdsourced label similarity graph induced from crowdsourced labels, where each labeler and labelee are annotated with a geographic coordinate; this similarity graph varies by location to capture spatial variations of the kind identified above (e.g., the similarity graph for San Francisco will link `energy` with `green movement`). Second, we transform this similarity graph into a directed weighted tree that imposes a hierarchical structure over these labels, such that labels like `sports` are higher in the tree, whereas labels like `rockets` are lower, thereby providing finer granularity for building user profiles. Finally, we embed this location-sensitive folksonomy into a user profile ranking algorithm that outputs a ranked list of candidate labels for a partially observed user profile. Through extensive experiments over a Twitter list dataset, we demonstrate the effectiveness of this location-sensitive user profile estimation.

4.2 Spatial Variation in Crowd-sourced Labels

We begin by examining the evidence of spatial variation in crowdsourced labels. We have argued that there could be differences from location to location, e.g., the label `energy` in San Francisco linking to the `green movement`, whereas in Houston it is more associ-

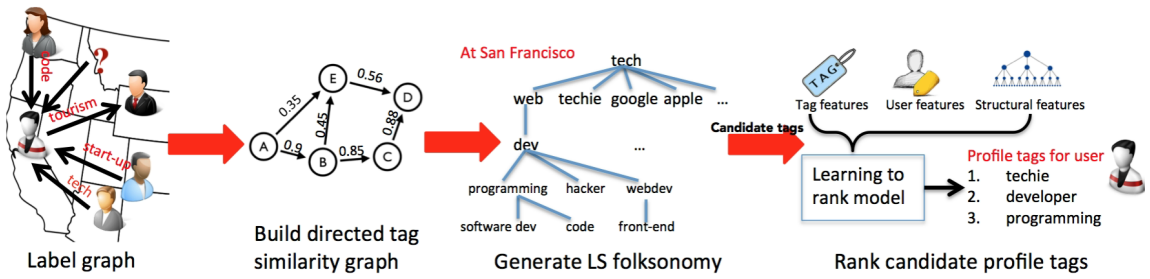


Figure 4.1: Overall approach: constructing location-sensitive user profiles from crowdsourced labels

ated with `oil and gas`. But does this spatial variation actually manifest in how crowdsourced labels (or tags) are applied? In this section, we provide data-driven evidence from a collection of Twitter lists (described more fully in Section 4.4.1). Twitter lists are one form of crowdsourced tagging, whereby individual Twitter users can add other users to a curated list annotated with a name. For example, a user could add a local blogger to a list called "Bloggers". In the aggregate these list names can be used to build user profiles as in related works like [6, 70, 128, 129]. The dataset we use here includes geo-coordinates for both the list labeler and the list labelee.

4.2.1 How does distance impact tagging?

We begin by investigating in Figure 4.2 the impact of distance on the probability that a list labeler will include another user on a list. We observe that the probability of tagging is exponentially decaying with distance, which indicates a user is less likely to be tagged by a labeler as the distance between them increases. This spatial locality is a well-known property of many off-line relationships and has been confirmed repeatedly even in online scenarios where distance is not inherently a limiting factor. This locality of tagging suggests that a method for user profile prediction that is induced from these crowd-based tags should reflect local knowledge; that is, since tags are not uniformly applied across distances, there may be local variations of interest.

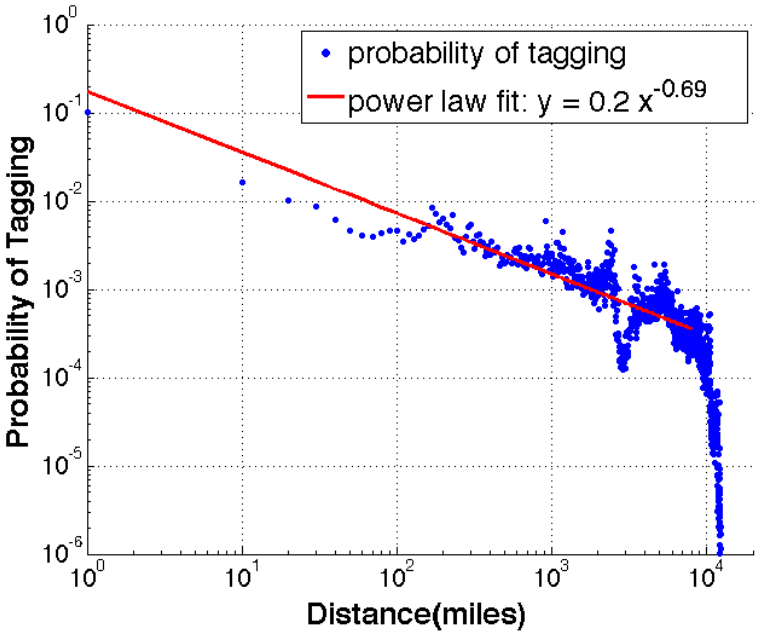


Figure 4.2: Probability of tagging as a function of distance between labeler and user. Tags are not uniformly applied across distances indicating that there are local variations of interest.

4.2.2 Are tags evenly distributed across locations?

We pair this first observation over users with a second investigation of the geographic distribution of the tags themselves. To study the relationship between a tag and its geolocation, we first define the location entropy as:

$$H(t) = -\sum_i p(l_i, t) \log(p(l_i, t))$$

where $p(l_i, t)$ is the probability of a tag t appearing in a location l_i. Location entropy measures how each tag is distributed across different locations. Tags that are popular globally will have a high entropy; tags that are more localized will appear in fewer locations and have a lower entropy. In Figure 4.3 we present the tag entropy across all tags over nine

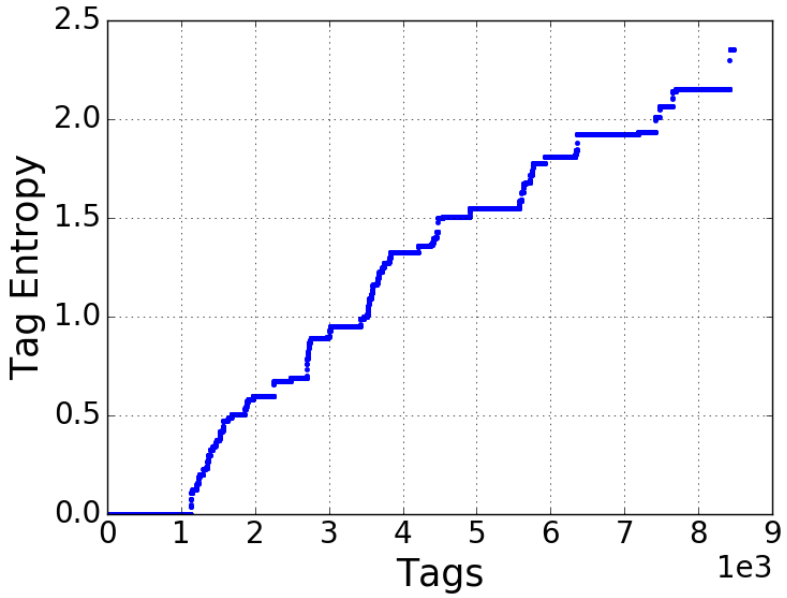


Figure 4.3: Entropy of each tag across nine locations. Tags are not homogeneously distributed across all locations. Some global tags like `news` and `sports` have a very high entropy, while local tags have low entropy.

locations in our dataset — New York, San Francisco, Seattle, London, Dallas, Houston, Chicago, Miami, and Los Angeles. We observe from the figure that these tags are not homogeneously distributed across all locations. Some global tags like `news` and `sports` have a very high entropy (the top-right of the distribution). On the other hand, tags related to specific sports team tend to have relatively low entropy, since fans mostly concentrate in a single home location. Other tags like `politics` and `movies`, though quite popular, have distributions that are not absolutely even. For example, we find many movie-related tags around Los Angeles and politics-related tags around Washington, DC. These differences suggest that a folksonomy induced over these tags can potentially organize tags according to their locality, with broadly popular tags like `sports` at higher positions in the folksonomy and specific local tags (like for specific sports teams) at lower positions in the folksonomy.

4.2.3 Example location-sensitive relationships.

Finally, we consider the relationships between pairs of labels across different locations in our dataset. Representing each tag as a location-specific vector (see the following section for additional details), we show in Figure 4.4 the relationships of a group of tag pairs at multiple locations using cosine similarity. The x-axis shows each tag pair and each color bar represents the similarity at a location, where a missing bar means the similarity at that location is less than 0.1. We observe that the magnitude of tag-pair relations varies across different locations. For example, we find the similarity between general concepts like `nba` and `basketball` tends to be relatively even across locations, with London having the lowest value; `finance` and `stock` have highest confidence in New York while lowest in Houston. Another typical example is the similarity between `energy` with `green` and `oil`. Interestingly, we notice `energy` and `oil` have the strongest relationship in Dallas and Houston, while `energy` and `green` bond closest in San Francisco. This fits our understanding of these locations, since Texas is a major oil and gas hub, while San Francisco is a more eco-friendly community. These phenomena suggest that location-sensitive user profiling has good potential to reflect the characteristics of these locations.

4.3 Location-Sensitive User Profiling

4.3.1 Problem Statement

Given these observations of the spatial variation of crowdsourced labels, we turn in this section to building *location-sensitive user profiles*. Our overarching goal is to estimate high-quality user profiles that respect this observed spatial variation. We assume some partial coverage of users via existing crowdsourced tags (e.g., from Twitter Lists or LinkedIn's Skill Tags), but that many tags are unknown. That is, given a user u's full (but hidden) tag profile $P(u)$, we have visibility only to some portion of this profile $P_k(u)$ where $P_k(u) \subset P(u)$. The goal is to estimate the unseen tags t_i of u where

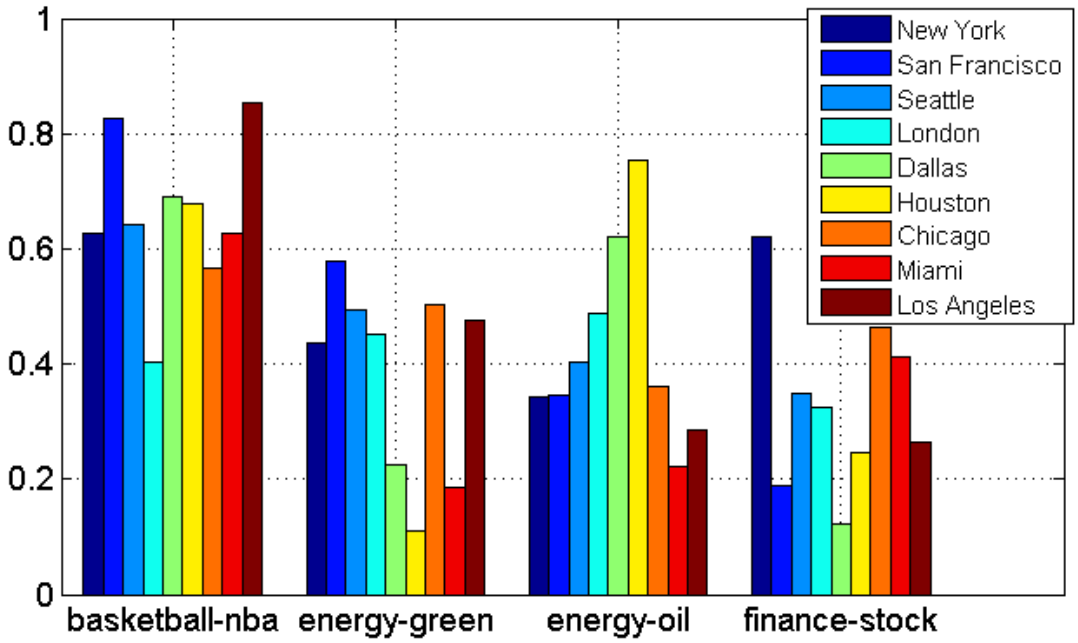


Figure 4.4: Example Tag Pairs Similarity. As an example, note that `energy` and `oil` have the strongest relationship in Dallas and Houston, while `energy` and `green` are closest in San Francisco.

$t_i \in P(u) - P_k(u)$.

4.3.2 Overview of Proposed Framework

Our intuition is that the spatial variation of how tags are applied can be carefully modeled to create high-quality user profiles. Concretely, we propose a three-step framework (as shown in Figure 4.1):

1. Crowdsourced Label Similarity Graph. We propose a location-sensitive distance weighting scheme that weights user influence according to distance, toward building a crowdsourced label similarity graph that integrates location similarity. This similarity graph varies by location to capture spatial variations of the kind identified above (e.g., the similarity graph for San Francisco will link `energy` with `green movement`).

2. Location-Sensitive Folksonomy Construction. Building on this crowdsourced la-

bel similarity graph, we propose a location-sensitive folksonomy construction framework that naturally integrates the geo-spatial scope of crowdsourced labels. This approach generalizes from previous folksonomy induction approaches, providing the basis for building user profiles.

3. Folksonomy-Informed Tag Prediction. Finally, we embed this location-sensitive folksonomy into a user profile ranking algorithm – first, we identify candidate tags from the location-sensitive folksonomy, and then we use a learning-to-rank approach to identify high-quality user profile tags for augmenting a partially observed user profile.

Our goal of the first two steps is to build a *location-sensitive folksonomy* that captures the spatial variation across locations. Given a group of users $\mathcal{U} = \{u_1, u_2, ..., u_n\}$, where each user u is associated with a geographic coordinate l_u and a tag profile P_u which contains a variable number of (tag, frequency) pairs $\{(t_1, f_1), (t_2, t_2), ...\}$. Our goal is to induce a folksonomy bound to a target location l, which is represented as directed rooted tree (arborescence) $\mathcal{T} = (\mathcal{V}, \mathcal{E})$. The node set $\mathcal{V}$ contains all unique tags that $\mathcal{U}$ has been labeled with, and edge set $\mathcal{E}$ contains subsumption relations of tag pairs. This implies that the tag tree is connected and each tag has a unique parent. The abstractness of each tag is controlled by its level in the tree. We can further assign weights to edges to capture the similarity between tag pairs. Note that we will build a different location-sensitive folksonomy for each location of interest (e.g., one for Los Angeles, one for Chicago).

4.3.3 Crowdsourced Label Similarity Graph

To build this folksonomy, we begin by proposing a distance weighting scheme which weights the profile tags of a user according to how far this user is from the target location. Our intuition is a distant labeler is considered less knowledgeable about local users. We adopt a model popularized in the GIS literature – the *zone of indifference model* – for capturing this spatial influence. The key idea is to combine the inverse distance with a

fixed distance band model. In this model, all users within the distance band are considered equally important and once beyond the threshold distance, a user's influence drops off quickly following an exponential rate. We empirically set the distance band as 50 miles for large cities, which defines a circle area centered at the target location. Hence, the weight of a user w.r.t the target location l_t is

$$w_u(l_t) = \begin{cases} 1 & \text{if } d \leq 50 \\ (\frac{d-50}{50})^{-\alpha} & \text{else} \end{cases}$$

where d is the distance from a user's location l_u to l_t and α is a constant, set experimentally. In this way we utilize the whole dataset for constructing the location-sensitive folksonomy for each location. This avoids the sparsity issues that may arise (if we were to build a location-sensitive folksonomy using only locally-available tags) and mitigating data imbalances across locations (so a smaller city is not penalized in folksonomy creation relative to a larger city). We represent each tag t_i at target location l_t as a tf-idf vector of the users who are labeled with the tag, and each user is weighted by her corresponding influence $w_u(l_t)$: $\boldsymbol{t_i'} = \boldsymbol{t_i} \cdot \boldsymbol{w_i}$. Thus, the tag vectors vary in each location of interest according to the change in users and weights.

With these tag vectors $\boldsymbol{t_i'}$, we can begin to construct a directed tag similarity graph. We compare four similarity measures to calculate the similarity between tag pairs. Three are symmetric measures including cosine, RBF kernel, and pointwise mutual information (PMI). The fourth measure is imbalanced (meaning the strength of one tag to a second tag is not necessarily the same as in the reverse situation) and based on a modified version of the traditional association rules notion of confidence (what we term *modified confidence*, or MConf).

Concretely, we calculate the tag similarity across crowd labels as follows:

Cosine similarity. This is the standard approach popularized in the information retrieval community, capturing the cosine of the angle between the vectors associated with two tags:

$$Cos(t_i, t_j) = \frac{d_i \cdot d_j}{|d_i||d_j|}$$

.

RBF kernel. It is defined as transformation of the squared Euclidean distance using a gaussian function:

$$K_r(t_i, t_j) = \exp(-\frac{\|d_i - d_j\|^2}{2\sigma^2})$$

We set the value of $sigma$ as the average Euclidean distance by default.

Pointwise mutual information. This approach is based on an information theoretic comparison of the distributions of the two tags, defined as:

$$PMI(t_i, t_j) = \log \frac{p(t_i, t_j)}{p(t_i)p(t_j)} = \log \frac{N(t_i, t_j)N_u}{N(t_i)N(t_j)}$$

where $p(t_i, t_j)$ is the joint probability of tag t_i and t_j, N_u is the number of total user considered, $N(t_i, t_j)$ is the number of user that has both t_i and t_j. PMI is a measure of how often two tags t_i and t_j occur, compared with what we would expect if they are independent, where the value equals to 0.

Modified confidence. Finally, we adopt a common approach from association rule mining which has been used for inferring subsumption relations between tags. Yet, it is not directly suitable for constructing a hierarchical structure. First, this approach models co-occurrence of tag as binary, i.e. it answers whether two tags show up together for a user or not, however, failing to consider the relative closeness of the two tags. For example, user u_a has been labeled with `sports` 5 times, `football` 6 times and `reading` 1 time. The contribution from u_a to the relation of tag pairs (sports, football) and (sports, read-

ing) are considered the same. However, these relations are actually in different magnitude and should be measured more properly. Second, it neglects relative frequency difference. Among the high confidence relations, many are actually resulting from an imbalance in frequency between the tag pairs. For example we may find higher confidence for the rule `LA Lakers` $\rightarrow$ `sports` instead of `LA Lakers` $\rightarrow$ `nba` as every time `LA Lakers` occurs, `sports` may also occur thus leading to a confidence of 1. However, this is not optimal for navigation and organization purpose. In terms of profiling, guiding by confidence will make the prediction task meaningless. Since instead of predicting very general tags, we would like to predict tags that are as specific as possible. Thus we propose a modified confidence metric for a tag pair $(t_i \Rightarrow t_j)$:

$$MConf(t_i \Rightarrow t_j) = \frac{C(t_i, t_j)}{F(t_i)} \cdot (1 - \frac{|\log(F(t_j)) - \log(F(t_i))|}{\log(\max F)})$$

where $C(t_i, t_j) = \min(f_1, f_2)$ is the co-occurrence frequency of the pair, $F(t_i) = \sum_u f_i$ is the overall frequency of t_i in the corpus and $\max F$ is the overall frequency of the most used tag in the corpus. Here we use confidence as a criteria for ordering tags and we only consider confidence for cases that $conf(t_i \Rightarrow t_j) \geq conf(t_j \Rightarrow t_i)$. The intuition of multiplying a weight is to avoid connecting tags with large frequency difference as mentioned in the second drawback. Even if confidence is high, there might be some intermediate nodes that fit between the nodes. As tag frequency versus number of tags follow a power law distribution, we model frequency difference with damping factor which is a fraction in log scale. When there is no frequency difference, the factor is 1, when there is large frequency difference, the factor decays to 0.

4.3.4 Location-Sensitive Folksonomy Construction

Given these measures of tag similarity that capture both user and location influences, we next turn to forming the directed weighted graph $\mathcal{G}$, representing the hierarchical struc-

ture organizing these tags. In order to define an order of abstractness for the tags, we calculate the closeness centrality of each tag, defined as

$$Centrality(t) = \sum_{j} sim(t, t_j)$$

which sums up each tag's similarity with all other tags. This definition forces general tags to have high centrality. Since the modified confidence definition already assigns a direction for a pair of tags, this step is exempted. Then we organize the tags and relations into a directed weighted graph $\mathcal{G}$. To do this, we initialize $\mathcal{G}$ with a ROOT node and add an edge for the tag pair when the similarity is above a pre-defined threshold (when 50% of tag-pairs are related) in Algorithm 1 line 1-5. The weight of the edge is set as the similarity value. Then, we assign a direction for each tag pair from the high centrality node to the low centrality node. As the graph is very likely not connected, we make the ROOT node point to every other node, with edge weight equal to the pre-defined threshold to make the graph weakly connected.

Now that we have the directed similarity graph, the goal is to find a tree such that our ROOT node is the root of the tree. From a graph perspective, constructing such a location-sensitive folksonomy is analogous to finding a spanning arborescence that satisfies some criteria in the directed acyclic graph. In graph theory, an arborescence is a directed rooted tree that has a root vertex and exactly one directed path from root to any other vertex; it is the directed analog of the minimum spanning tree.

A straightforward criteria is to find a tree that maximizes the edge weights. In essence, this follows a greedy strategy which was used previously by Heymann et al. [133]. They proposed to iteratively add nodes in a decreasing centrality order to a tree which maximizes similarity. From a graph point of view, we can apply Chu-Liu/Edmond's algorithm [134] over the similarity graph. The core procedure is finding the edge incoming to node t of

highest weight (with ties broken arbitrarily) for each t other than the ROOT. Since the edge order is pre-defined according to centrality, the graph is guaranteed to have no cycles and we can simplify the algorithm to forgo this cycle check.

Generalized Cost Function Although different metrics are adopted for characterizing the relation between a pair of tags, they share the similar greedy strategy of minimizing the cost function

$$cost = \sum_{e \in \mathcal{G}} W(e) - \sum_{t \in \mathcal{G}} sim(t, t_p)$$

where $sim(t, t_p)$ represents the similarity between a tag t and its parent. Here we introduce a new minimum cost tree formation algorithm which builds upon the simplified Chu-Liu/Edmond's algorithm that generalizes the cost function. Concretely, the proposed folksonomy generation algorithm can be formalized as Algorithm 1.

After constructing a directed weighted graph $\mathcal{G}$, we next convert this graph to a tree $\mathcal{T}$ with minimum cost, where the cost here characterizes the structural change to $\mathcal{G}$. We define the cost for deleting the edge as $sim(t_i, t_j) \cdot d_{i,j}$, where $d_{i,j}$ represent the shortest path length from t_i to t_j in the graph which excludes the edge (t_i, t_j). After deleting an edge, the two corresponding nodes are disconnected and we need to identify a new shortest route that connect these two nodes. The intuition is we want to maintain the coherence of the structure after deleting edges such that more similar tags tend to stay closer to each other. To do so, in each iteration, we calculate the cost of deleting each remaining edge in the graph $\mathcal{G}'$, and then find an edge with minimum cost which is not the last remaining edge pointing to the corresponding child node (so the node is not isolated). The algorithm stops when $n - 1$ edges are left, with each node having exactly one parent. Thus our goal is to minimize the *structure conservation* cost function for converting $\mathcal{G}$ to a tree $\mathcal{T}$:

$$cost = \sum\nolimits_{e \in \mathcal{G}, e \notin \mathcal{T}} sim(t_i, t_j) \cdot d_{i,j}$$

Algorithm 1: Mincost Tree Formation

Input: Tag vectors

Calculate similarity between each pair of node (t_i, t_j)
Initialize directed weighted graph $\mathcal{G}$ with ROOT node
Add an edge (t_i, t_j) when $Sim(t_i, t_j) > threshold$
Assign direction for edges following centrality order
Add an edge between ROOT and each node with $weight = threshold$
while *n(edge)>n-1* **do**
    for *each edge (t_i, t_j)* **do**
        G' ← remove edge (t_i, t_j) from graph G
        Find shortest path from t_i to t_j in G'
        Calculate cost of deleting edge(t_i, t_j)
    end
    i=0
    while *edge not removed* **do**
        edge=increasing_cost_sequence[i]
        if *the edge is not the last incoming edge to t_j* **then** remove edge from G and break;
        i+=1
    end
end
return $\mathcal{G}$

.

However, this algorithm is computationally costly as whenever a new edge is deleted, it is required to recompute the shortest path between each pair of nodes. This is an O(E^2) shortest path calculation. Hence, we provide an approximation for the calculation shown in algorithm 2, where we only calculate the cost of deleting each edge in the original graph. According to the cost from low to high, we iteratively delete the edges until there is a unique parent for each node. Finally, the output is a location-sensitive folksonomy.

Algorithm 2: Approximation Algorithm

Input: Tag vectors
Constructing the Directed Weighed Graph $\mathcal{G}$ according to lines in Algorithm 1
for *each edge (t_i, t_j)* **do**
    G' $\leftarrow$ remove edge (t_i, t_j) from graph $\mathcal{G}$
    Find shortest path from t_i to t_j in G'
    Calculate cost of deleting edge(t_i, t_j)
end
while *n(edge)>n-1* **do**
    for *each edge in increasing cost* **do**
        if *the edge is not the last incoming edge to t_j* **then** Remove edge from $\mathcal{G}$;
    end
end
return $\mathcal{G}$

4.3.5 Folksonomy-Informed Profiling

We turn in this section to apply the location-sensitive folksonomy for profile construction. We begin by finding candidate tags from this folksonomy, and then embedding these candidates in a learning-to-rank framework for finding the best order of the tags. We continue to study how to acquire a set of candidate tags from location sensitive taxonomy. Then we introduce a learning to rank based method that learns the optimal weight for

features to rank the candidate tags for prediction.

Finding Candidate Tags. Given a user's seen tag profile $P_s(u)$, we first leverage the location-sensitive folksonomy and select a set of candidate tags. To accomplish this, we locate each seen tag in $P_s(u)$ in the folksonomy and collect parent, children, and sibling tags of this seen tag as candidate tags. The hierarchical structure acts as a good filter and thus controls the number and quality of candidate tags. Then we order the candidate tags according to different strategies for prediction. The formal definition for this problem is given user u and a set of candidate tags $T_c(u) = \{t_1, ..., t_k\}$, we aim to find a scoring function to rank tags in $T_c(u)$ for u.

Ranking Candidate Tags. We propose to apply a learning to rank approach for ranking the candidate tags for a user. The advantage is that it automatically chooses the optimum weight for each feature, which leads to a high quality order. We apply a pairwise learning algorithm RankSVM [135]. Here we consider each user as a query and we assign each candidate tag an integer ranking score in the range of 3 to 1 according to the actual number of tag in the user's unseen profile. RankSVM first generates a set of pairwise constraints and then transform the problem to a two-class classification problem according to those constrains and an SVM model is learned. Finally, in the ranking phase, rank scores are calculated according to the margin value. Note that we train the ranking model with the training set and an L2 norm regularization term is added to prevent overfitting.

Here we introduce a set of features that we rely on to generate a preference order of the candidate tags for prediction in Table 4.1. A total of 13 features are used for training the model include features introduced above. Features can be grouped into three categories: user specific features, tag features, and folksonomy structure features. User specific features include $f_u(s)$, $f_u(t)$, H_{sim}, Sum_{sim}, Hf_{sim}, S_{mv}^{w1}, S_{mv}^{w2}. These features are retrieved from a user's seen profile, which represents characteristics of the user. Tag features in-

Features	Descriptions
$f(s)$	log scale overall frequency(freq) of the seed tag
$f(t)$	log scale overall freq of the tag
$f_u(s)$	log scale unique user freq of the seed tag
$f_u(t)$	log scale unique user freq of the tag
S_{sim}	similarity with seed tag
H_{sim}	highest similarity with existing tags
Hf_{sim}	H_{sim} weighted by freq
Sum_{sim}	sum of similarity with all seen tags
S_{mv}^{w1}	sum of similarity with existing seed tags weighted by $f(s)$
S_{mv}^{w2}	sum of similarity with existing seed tags weighted by $f_u(s)$
p_{cnt}	freq of the candidate tag being a parent
s_{cnt}	freq of the candidate tag being a sibling
c_{cnt}	freq of the candidate tag being a child

Table 4.1: Features for ranking candidate tags from the location-sensitive folksonomy.

cludes $f(s)$, $f(t)$, and S_{sim}. These features only provide intrinsic properties of candidate tags. And folksonomy structure features include p_{cnt}, c_{cnt}, s_{cnt}, which are uniquely defined by the folksonomy to provide extra clues for making good predictions. The intuition here is that predicting a parent tag is more likely to be correct than a sibling or child, as parent tags are more general, having the largest overlap with the candidate. For example, inferring `football` to parent `sports` is more likely to be correct than to sibling `volleyball` and child `football player`.

4.4 Experiments

In this section, we conduct a set of experiments to evaluate the location-sensitive user profile prediction. Specifically, we first introduce the data preparation workflow and basic experimental setup. Then, we begin by comparing the learning to rank based tag ranking with baselines, followed by a comparison of location-sensitive folksonomy with general folksonomy. Finally, we study the effectiveness of folksonomy-informed approach by

comparing with collaborative filtering and BPR-MF for implicit feedback baselines.

4.4.1 Setup

Data Preparation. We rely on a Twitter list dataset containing 15 million list relationships in which the geo-coordinates of the labelers and users are known. In our experiments, the tags we included in the folksonomy are extracted from each list name, and users in the list will be endowed with the tags in their profile. These tags contain multifaceted opinions of actual labelers, which means they can be complex and noisy. Hence, we apply text processing techniques such as case folding, stopword removal, and noun singularization. We also separate the string pattern like 'FoodDrink' into two words 'food' and 'drink'. We use language identification package [136] to filter out non-English tags. To guarantee the informativeness and quality of the tags, we filter out infrequent tags with fewer than 5 labelers and 10 users. Twitter has a 25-character length limit for list names, and empirically we find nearly all list names do not exceed three words. Many single words may lose the theme of the list – for example, `machine learning` will provide tags `machine` and `learning` for the users in the list, which loses its original meaning. We thus include bigrams that have a frequency above another threshold. Meanwhile, we remove unigrams that always exist in such bigrams. Finally, we fixed the size of the tagset at 10,489.

Profile Prediction Setup. For each of nine selected locations, a random sample of local users is held out. We construct a location-sensitive folksonomy given the location based on the rest of whole dataset. Following that we predict the user profiles for users in the hold-out data.For each user, the seen tag set $P_k(u)$ is a random 25% of his profile $P(u)$. Then we try to predict tags in the rest 75% unseen tags.[2] The result reported for every profiling experiment in this chapter, including baselines, are based on four-fold cross validation and averaged over the nine locations.

[2]We only consider users with overall more than 10 tags.

Threshold for Subsumption. To decide the threshold for subsumption for different similarity measures, we recruit two human judges to manually label 100 tag pairs within different similarity range. They are asked to decide whether there is recognizable similarity between two tags. We repeat the process for each of the similarity measure. We use Cohen's kappa to measure the inter-rater agreement, and a value of 0.64 is reached, thus we conclude the judgments are consistent. Finally we pick a threshold that at least 50% of the relations are related. For example, the final threshold for cosine is 0.3.

Baselines. We consider two approaches based on collaborative filtering and Bayesian personalized ranking as baselines.

Collaborative Filtering-K nearest neighbor(CF-KNN). In CF-based prediction approach, we first identify the top-k local users that share the most similar tags with the target user. To maintain consistency with other approaches, we assume each user profile only contains 1/4 of the tags). Here, we apply cosine similarity to measure user similarity. Then, we aggregate the tags of the 50 nearest neighboring users weighted by their similarity and make predictions based on decreasing tag frequency in the collective neighbor profile.

Bayesian Personalized Ranking-Matrix Factorization(BPR-MF). We consider these tags as implicit feedback and our goal is identifying an optimal preference ranking of tags for each user. We thus experiment with two variations of state-of-the-art Bayesian personalized ranking criteria [79], where the goal is to choose parameters that maximize the posterior probability. In the first setting, we train a unique model for each location by only considering its local users, denoted as "LBPR-MF". We model a user i's affinity to tag j as $r_{ij} = p_i q_j + b_j$, where p_i and q_j represent latent factor of user and tag, respectively. b_j represents the overall preference of the tag j. In the second setting, we train with whole dataset and explicitly model location-aware preferences, denoted as "LABPR-MF". We define a user i's affinity to tag j as as $r_{ij} = p_i q_j + g_{l(i)j} + b_j$, where latent factor $g_{l(i)j}$

represents the regional popularity of tag j at the user i's home location [5]. For reproducibility, the number of negative samples is set as 200, number of iterations is set as 100, number of latent factors of user and tag are set to 20, and the regularization weights are set as 0.02.

Evaluation Metrics. The evaluation metrics we use are Precision@k (P@K) and Actual Frequency@k (AF@k). P@k measures how reliable predictions can be made. A high P@k value implies users have been labeled with the predicted tag, while high AF@k represents that users have been labeled many times with the predicted tags. Both measurements reported later are averaged over the test data. We consider the quality of prediction for the 1^{st} tag as well as top-5 tags. AF@k is defined as:

$$\text{AF@}k = \sum\nolimits_k f_u(t_k)/k$$

where t_k is the kth predicted tag.

When comparing across the similarity measures in folksonomy-informed approaches, WE CONSIDER another metric, the average number of candidates selected from user profile: this metric is an efficiency indicator. A larger value means the tag candidate set is large, which requires more computation.

$$NumC = \sum\nolimits_i (p(o_i) + c(o_i) + s(o_i))/n$$

where o_i represent the ith observed tag in the user's profile. p, c, s represent the number of parent, children and sibling of the observed tag.

We now turn to the task of user profile construction based on location-sensitive folksonomies. We first compare the performance based on different ranking strategies, followed by profiling performance across different folksonomy generation approaches and

Methods	P@1	P@5	AF@1	AF@5
Highest similarity	0.333†	0.285†	4.82†	4.24†
Frequency & similarity	0.507†	0.360†	14.6†	6.87†
Overall popularity	0.607†	0.501†	25.9	15.1†
$Sum_{similarity}$	0.651†	0.552†	25.3†	18.5
Learning to rank	0.763	0.677	26.5	19.2

Table 4.2: Comparing Tag Ranking Approaches. We observe that the LTR based approach results in the best precision, and also identifies the tags used most often (AF). '†' marks statistical significant difference with LTR according to paired t-test at significance level 0.05.

local and general versions. Finally, we compare location-sensitive folksonomy informed profiling with the other baselines.

4.4.2 Comparing Ranking Strategies

Given the candidate tags identified from the folksonomy, our goal is to generate a personalized ranking over these tags so that the actual tags rank top. Here we compare the learning-to-rank (LTR) based approach with several baselines in Table 4.2, 1.ranking the candidate tags according to the decreasing order of similarity with the seen tag which subsumes the candidate tag. 2.ranking with a hierarchical criteria, primarily according to the frequency of corresponding seen tag associated with the user and secondarily by decreasing order of similarity with the corresponding seen tag. 3.ranking by overall tag popularity. 4.ranking according to the aggregated similarity with the seen tag set.

We observe the LTR based approach outperforms all baselines in terms of precision, indicating high rank tags are more likely to be actual tags. Moreover, we find a similar trend in terms of AF@5, which represents the actual number of predicted tags that a user possesses. All these results imply the effectiveness of proposed features and feature weight scheme. Among the baselines, we find the "overall popularity" and "$Sum_{similarity}$" are relatively strong predictors.

4.4.3 Location Sensitive vs General Folksonomy

Methods	LS Folksonomy		LS Folksonomy		
	P@1	P@5	AF@1	AF@5	NumC
MaxSim-MC	0.754	0.663	26.0	19.2	71.4
StrCon-MC	0.763	0.677	26.5	19.2	70.9
MaxSim-COS	0.751	0.6561	25.3	16.4	12.1
StrCon-COS	0.756	0.663	26.4	18.1	11.7
MaxSim-PMI	0.389	0.349	2.33	1.24	5.28
StrCon-PMI	0.402	0.362	2.42	1.21	5.11
MaxSim-RBF	0.566	0.413	15.8	9.94	7.87
StrCon-RBF	0.581	0.430	14.3	10.1	8.20
Methods	**General Folksonomy**		**General Folksonomy**		
	P@1	P@5	AF@1	AF@5	NumC
MaxSim-MC	0.653	0.571	23.3	15.2	162
StrCon-MC	0.656	0.571	23.2	15.4	166
MaxSim-COS	0.662	0.521	19.1	11.9	10.0
StrCon-COS	0.662	0.531	19.2	12.8	9.57
MaxSim-PMI	0.352	0.320	1.85	1.53	13.2
StrCon-PMI	0.363	0.332	2.01	1.62	13.5
MaxSim-RBF	0.471	0.422	12.1	8.22	10.5
StrCon-RBF	0.465	0.410	14.0	12.0	12.3

Table 4.3: Comparing Location-Sensitive Folksonomy and General Folksonomy in Profile Tag Prediction. All location-sensitive versions are statistical significantly different with general version according to paired t-test at significance level 0.05.

In Table 4.3, we compare the location-sensitive folksonomy versus the general folksonomy over all design choices for the application of profile construction. For each design choice, we generate a location-sensitive version for each of the 9 locations mentioned in Table 4.4, where the shown result for location-sensitive folksonomy is averaged across locations, as well as general folksonomy-informed version which is constructed using the whole dataset excluding distance and location factors. We observe that overall, location-sensitive versions always beat its general counterpart in terms of precision@k and AF@k,

regardless of design choice. The priority in terms of P@5 is around 0.1 and AF@5 is above by 20%. This result justifies the location-sensitive folksonomy since it better captures the local knowledge structures. It also demonstrates the effectiveness of how we model distance influence. We also find that the performance is consistent across locations.

We next compare the four similarity metrics used for constructing the tag similarity graph at the heart of location-sensitive folksonomy construction. Design choices with modified confidence (MC) perform best in terms of P@k, and AF@k, with the folksonomy built on top of cosine similarity slightly lower. RBF kernel ranks the third and PMI last. We also notice that for the average candidate tag considered, the MC is much higher than other approaches. The reason is that the folksonomy generated with MC is shallower compared with the folksonomy generated using cosine and RBF kernel, meanwhile the number of sibling and children node for each tag are more. These factors lead to more candidate tags for ranking which explains why the performance is higher in terms of P@k and AF@k. The lesson here is a trade-off between efficiency and performance. Next we discuss the extremely low performance for PMI, which indicates it may not be suitable for this scenario. After inspecting the folksonomy, we observe many of the hierarchical relations are incorrect or meaningless. PMI only considers co-occurrence of tags without taking the relative frequency difference into account. In our experiment design, as we don't set a minimum tag occurrence for each user to avoid sparsity, thus many tags only show up once on a user, which meanwhile, creates noise for an approach like PMI.

Last but not least, we evaluate the proposed *structure conservation* cost function (StrCon) and compare with its *maximum similarity* cost function (MaxSim) baseline. The structure conservation cost function aims to construct a folksonomy that makes the smallest change to the similarity graph. We observe in Table 4.3 that applying StrCon leads to an incremental change in profile construction. For example, in the cosine case, we notice the P@5 and AF@5 are slightly better for StrCon. We notice about 9.5% of the relations

Methods	P@1	P@5	AF@1	AF@5	TFIDF@5
CF-KNN	0.656†	0.542†	25.6	18.0†	36.1†
LBPR-MF	0.731†	0.650†	22.6†	12.3†	30.2†
LABPR-MF	0.771	0.673	24.1†	16.2†	34.6†
LS Folk.	0.763	0.677	26.5	19.2	42.6

Table 4.4: Comparing Tag Prediction Approaches. The BPR-MF improves upon the more naive CF-KNN approach. Location-sensitive folksonomy informed approach achieves comparable precision with LABPR-MF.'†' marks statistical significant difference with location-sensitive folksonomy according to paired t-test at significance level 0.05.

are different among folksonomy generated using the two cost functions, meaning that the StrCon approach made some structure adjustments with some sacrifice in connecting most similar first strategy. Considering the limited difference in the two folksonomies, this increase in performance can be attributed to a better macro-structure.

4.4.4 Evaluate User Profiling

Finally, we compare location-sensitive folksonomy-informed user profiling with the CF-KNN and BPR-MF baselines. As we observe in Table 4.4, the proposed approach outperforms the CF-KNN and locally trained BPR-MF in both P@k and AF@k and exhibits similar performance compared with the location-aware BPR-MF approach. All three baselines yield better results than many aforementioned folksonomy-informed approaches with simple ranking strategy. For example LBPR-MF reaches 0.65 for P@5, higher than all other alternatives of folksonomy-informed method. Even though BPR-MF does not consider frequency which is also important, the latent factors effectively capture user preferences over tags and location-based preference for tags. However, BPR-MF is computationally costly as the dimension of user and tag increase. The CF-KNN approach is not robust in sparse condition, for example, when there are few similar users, the prediction made by CF-KNN could be very inaccurate.

We leverage average TFIDF score for top five predicted tags as a metric to reflect how important and informative the predicted tag is to a user in the actual tag collection. The score is averaged for users and locations. We notice the proposed LS-Folk yield the highest TFIDF@5, showing the capability of identifying uniquely important tag for the user.

We notice that CF-KNN and BPR-MF based approach have a strong tendency to predict general high frequency tags. For CF-KNN, highly general tags are very likely to rank top in the sequence. For BPR-MF, the implicit feedback formulation neglects the difference in importance of the seen tags and has a tendency to predict tags that are seen on many users. These tags are often on a high abstraction level and thus provide only vague insight to a user. For example, if the predicted tag is `peep` which is short for "people", there is little new information contributed to the target user. In order to precisely profile a user, we expect to have concrete and specific tags, additionally, we wish to have a diverse tag space. In contrast, we observe that the location-sensitive folksonomy based approach performs much better in predicting diverse specific tags. For example, we find `tech-news` instead of only `news` in other methods.

4.5 Summary

In this chapter, we introduced a framework based location-sensitive folksonomy and tag ranking towards the goal of improved user profiling. Our key motivating intuition is that spatial variation of user tagging manifests in how users organize and apply tags, and is critical for building more robust folksonomies. Concretely, we have formulated the folksonomy generation problem as identifying a certain spanning arborescence from a directed acyclic graph. We have integrated location information through a weighting scheme that decrease the user's influence according to distance. We have presented a new cost function generalized over the Chu-Liu/Edmond's algorithm for finding a folksonomy that maximumly preserves the structure of the graph. Through extensive experiments, we

have demonstrated the impact of such a location-sensitive folksonomy on finding relevant tags for user profiling. We have seen that the location-sensitive folksonomy-informed user profiling is more effective in finding relevant tags, and learning to rank strategy is helpful for optimizing weights of each feature and leads to high quality user profile tags. In our future work, we plan to investigate how the location-sensitive folksonomy can enhance other local search applications.

5. RANKING: LOCAL EXPERT DISCOVERY[1]

In chapter 3 and chapter 4, we focused on the location and user profiling problems and we investigated how we can improve the characterization of locations and users within a geo-social system. In this chapter, we investigate how we can jointly model both locations and users towards tackling the problem of local expert discovery.

5.1 Introduction

Identifying *experts* is a critical component for many important tasks, including search and recommendation systems, question-answer platforms, social media ranking, and enterprise knowledge discovery. Indeed, there has been a sustained research effort to develop algorithms and frameworks to uncover experts, e.g., [62, 65, 66, 69, 70, 137, 138, 139, 140]. These efforts have typically sought to identify *general topic experts* – like the best Java programmer on github – often by mining information sharing platforms like blogs, email networks, or social media. However, there is a research gap in our understanding of *local experts*. Local experts, in contrast to general topic experts, have specialized knowledge focused around a particular location. To illustrate, consider the following two local experts:

- A "health and nutrition" local expert in San Francisco is someone who may be knowledgeable about San Francisco-based pharmacies, local health providers, local health insurance options, and markets offering specialized nutritional supplements or restricted diet options (e.g., for gluten allergies or strictly vegan diets).

[1]Part of this chapter is reprinted with permission from "On Local Expert Discovery via Geo-Located Crowds, Queries, and Candidates" by Wei Niu, Zhijiao Liu and James Caverlee. 2016. ACM Transaction on Spatial Algorithms and Systems. 2, 4, Article 14 (November 2016), 24 pages. DOI: https://10.1145/2994599. Copyright 2016 by Association for Computing Machinery (ACM). Part of this chapter is reprinted with permission from "LExL: A Learning Approach for Local Expert Discovery on Twitter." Ferro N. et al.(eds) Advances in Information Retrieval. ECIR 2016. Copyright 2016 by Springer International Publishing Switzerland.

- A "techie" local expert in Seattle is someone who is knowledgeable about the local tech scene, and may be able to answer local information needs like: who are knowledgeable local entrepreneurs, what are the tech-oriented neighborhood hangouts, and who are the top local talent (e.g., do you know any experienced, available web developers?).

Identifying local experts can improve location-based search and recommendation, and create the foundation for new crowd-powered systems that connect people to knowledgeable locals. Furthermore, after these local experts have been detected, their knowledge can provide the foundation for many new location-centric information applications. Examples include: (i) local query answering, whereby complex information needs that cannot be satisfied by traditional search engines could be routed to knowledgeable locals; (ii) curated social streams, whereby Facebook and Twitter-like social streams can be re-organized to focus on locally significant events and topics (for example, to ameliorate panic during disease outbreaks by alerting residents of facts from local health experts, rather than on global reports – in the recent panic surrounding ebola, Dallas residents could be made aware of local precautions rather than focusing on nation-wide news summaries); (iii) location-based recommendations, whereby entities of interest to local experts can be recommended to newcomers (for example, to recommend good venues for meeting local entreprenuers); and on and on. A critical first step in all these cases is in the *identification of local experts*.

Compared to general topic expert finding, there has been little research in uncovering who are these local experts. Most existing expert finding approaches have typically focused on either small-scale, difficult-to-scale curation of experts (e.g., a magazine's list of the "Top 100 Lawyers in Houston") or on automated methods that can mine large-scale information sharing platforms, e.g., [62, 65, 66, 69, 70, 137, 138, 139, 140]. These approaches, however, have typically focused on finding general topic experts, rather than

local experts. And yet there is growing evidence of the importance of location-centered services: according to a recent Pew Research study: "Location tagging on social media is up: 30% of social media users now tag their posts with their location. For mobile location services, 74% of smartphone owners get directions or other information based on their current location, and 12% use a geo-social service such as Foursquare to "check in" to locations or share their whereabouts with friends." [141].

Hence, our focus in this chapter is in developing robust models of *local expertise* that opportunistically leverage this recent rise of *location* as a central organizing theme of how users engage with online information services and with each other. Concretely, we propose and evaluate a geo-spatial learning-to-rank framework called **LExL** for identifying local experts that leverages the fine-grained GPS coordinates of millions of Twitter users and their relationships in Twitter lists, a form of crowd-sourced knowledge. The framework investigates multiple classes of features that impact local expertise including: (i) user-based features (e.g., the number of users a candidate is following, the number of posts this candidate has made); (ii) tweet content features (e.g., tweet-based entropy of a candidate, the TFIDF score of a topic keyword in a candidate's tweets); (iii) list-based features (e.g., the number of lists the candidate is a member of, the number of lists the candidate has created); (iv) local authority features (e.g., the distance between candidate and the query location, the average distance from a candidate's labelers to the candidate); and (v) features based on a location-sensitive random walk that propagates crowd knowledge of a candidate's expertise.

Through a controlled study over Amazon Mechanical Turk, we find that the proposed local expert learning approach results in a large and significant improvement in Precision@10, NDCG@10, and in the average quality of local experts discovered versus two state-of-the-art alternatives. We additionally investigate the relative impact of different classes of features, and examine the generalizability of the approach in terms of reusing

the learned model in different topics. Our findings indicate that careful consideration of the relationships between the location of the query, the location of the crowd, and the locations of expert candidates can lead to powerful indicators of local expertise. We also find that high-quality local expert models can be built with fairly compact features, meaning these models can be potentially adapted to more constrained scenarios (e.g., in domains with only partial features). Finally, we find that the proposed local expertise models are generalizable: in many scenarios, local experts can be discovered on new topics and in new locations, which is important for uncovering previously unknown experts in emerging areas or in nascent communities.

5.2 Learning Approach to Local Expert Finding

In this section, we introduce the learning approach framework for finding local experts – **LExL**: **L**ocal **Ex**pert **L**earning. Given a query, composed of a topic and a location, the goal of LExL is to identify high-quality local experts.

5.2.1 Problem Statement

We assume there is a pool of local expert candidates $V = \{v_1, v_2, ..., v_n\}$, each candidate is described by a matrix of topic-location expertise scores (e.g., column i is College Station, while row j is "web development"), and that each matrix element indicates to what extent the candidate is an expert on the corresponding topic in the corresponding location. Given a query q that includes both a topic t and a location l, our goal is to find the set of k candidates with the highest local expertise in query topic t and location l. For example, find the top experts on t_q = "web development" in l_q = College Station, TX. Note that the query location l can be represented at multiple granularities – e.g., a city name, a latitude-longitude coordinate. This location indicates the region of interest for the query issuer, and so is not constrained to the home location or current position of this query issuer; for example, a user in San Francisco may issue a local expert query for Houston

before visiting on a business trip.

5.2.2 Overview of Approach

To tackle the local expert finding problem, we propose a geo-spatial approach that integrates geo-crowd knowledge about each candidate with a learning-to-rank framework. Concretely, we exploit the crowd wisdom embedded in millions of geo-located Twitter lists, coupled with a learning framework to isolate the critical features that are correlated with local expertise.

Geo-Located Twitter Lists. A Twitter list allows an individual on Twitter to organize who she follows into logical lists. For example, Figure 5.1 shows one list named "Food!" which contains 14 Twitter accounts including Alton Brown, Mind of a Chef, and America's Test Kitchen. Collectively, Twitter lists are a form of crowd-sourced knowledge, whereby aggregating the individual lists constructed by distinct users can reveal the crowd perspective on how a Twitter user is perceived [70]. In this chapter, we exploit the geo-social information of 13 million lists – provided to us by the authors of [6] – in which we have the fine-grained location information of both the list creator (or *labeler*) and the member of the list (or *labelee*). In total, there are 86 million user occurrences on these lists, of which we have 15 million geo-tagged list relationships. So, we know for example that Alice from Houston has labeled Bob from College Station with the label Foodie. Thus, the aggregate list information may reveal not just the general crowd perspective on each user, but the *local crowd's perspective*. High-level statistics of the dataset are listed in Table 5.1. Further details of the dataset collection method can be found in [6]. In addition to this list information, we also crawl the content associated with these users for the period of May 2015 to September 2015.

Figure 5.1: Twitter List Example

Data Type	Total # of Records
Lists	12,882,292
User List Occurrences	85,988,377
Geo-Tagged List Relationships	14,763,767

Table 5.1: Geo-tagged Twitter list data

5.2.3 Learning Approach

A previous approach by [6] focused on the local expert ranking problem using a linear combination of topical authority and local authority. In that work, topical authority was designed to capture the candidate's expertise on a topic area, e.g., how much does this candidate know about web development? They adopted a language modeling approach [62] adapted to Twitter lists, where each candidate was described by a language model based on the Twitter list labels that the crowd has applied to them.

Local authority was designed to capture a candidate's authority with respect to a location, e.g., how well does the local community recognize this candidate's expertise? Several approaches were suggested, including one which measured the average distance spread of list labelers to a candidate, with respect to a query location – so that candidates who were listed by many people in an area of interest (e.g,. Joe has been labeled by 100 people from

College Station) would be considered locally authoritative (e.g., Joe is well-recognized in College Station). These two aspects of local expertise – topical authority and local authority – were combined in a linear fashion to arrive at an overall score for each candidate.

More generally and in the presence of ground truth training data (see the evaluation setting in Section 5.4.2), we propose to transform the local expert ranking problem from an unsupervised linear combination of local authority and topical authority into a supervised learning-to-rank framework that can combine any number of local expertise features, using a tool such as LambdaMART [142, 143]. While we experiment with four different learning to rank algorithms in the experiments, we focus our discussion here on LambdaMART, as a representative learning-to-rank framework (which we find experimentally has the best performance). LambdaMART is an instance of Multiple Additive Regression Tree (MART), which is based on the idea of boosting. It trains an ensemble of weak regression tree models and then linearly combines the prediction of each one of them into a final model which is stronger and more accurate. In ranking tasks, the measures that are typically optimized include NDCG, MAP, or MRR. Unfortunately, gradient boosting is not suitable if we directly consider these metrics as loss functions since they are not differentiable at all points. Hence, LambdaMART tunes the parameters of the regression trees using a variable λ, indicating whether the documents should be moved up or down in the rank list, as the gradient of parameters based on the evaluation metric used, such that the evaluation metric is optimized directly while learning. In our experiments, we adopt NDCG as the evaluation metric.

5.3 Features for Local Expertise

In this section, we describe five classes of features that potentially contribute to local topic expertise of a user: user-based features, tweet content features, list-based features, local authority features and distance-biased random walk features. The user-based, list-

		User-based Features
1	$N_{follower}$	The number of followers this candidate has.
2	N_{friend}	The number of users this candidate is following.
3	N_{fav}	The number of tweets this candidate has favorited in the account's lifetime.
4	N_{status}	The number of tweets (including retweets) posted by the candidate.
5	T_{create}	The UTC datetime that the user account was created on Twitter.
		Tweet Content Features
6	twP_c	Avg. number of tweets that a candidate c posted in one week.
7	tw_H	Avg. tweet Entropy of a candidate c.
8-15	twB_t	Topic bayesian scores of a candidate c in 8 topics, t.
16-19	twB_l	Location bayesian scores of a candidate c in 4 topics, l.
		List-based Features
20	N_{listed}	The number of lists that this candidate appears on.
21	T_{listed}	The number of on-topic lists that this candidate appears on.
22	N_{list}	The number of lists this candidate has created.
23	T_{list}	The number of on-topic lists this candidate has created.
24	$list\_score_c$	The average quality of the lists that the candidate is a member of.
		Local Authority Features
25	d_c	Avg. distance from candidate c to all the users who appear on c's lists.
26	d_{ct}	Avg. distance from candidate c to all the users who appear on c's on-topic lists.
27	d_u	Avg. distance from candidate c to all the labelers of c.
28	d_{ut}	Avg. distance from candidate c to all the labelers whose on-topic list has c.
39	d_{uq}	Avg. distance from query location to labelers whose on-topic list has c.
30	d_{cq}	Distance between candidate c and the query location l_q.
31	$Prox_c$	Candidate Proximity, as defined in Section 5.3.4.
32	$Prox_{spread}$	Spread-based Proximity, as defined in Section 5.3.4.
		Distance-Biased Random Walk Features
33	ha_0	$\mathcal{A}^n(c) = p \cdot \sum_{u_i} \mathcal{H}^{n-1}(u_i) \cdot \frac{1}{\mathcal{O}(u_i)} + \frac{1-p}{N}$ $\mathcal{H}^n(u) = p \cdot \sum_{c_i} \mathcal{A}^{n-1}(c_i) \cdot \frac{1}{\mathcal{I}(c_i)} + \frac{1-p}{N}$
34	ha_1	$\mathcal{A}^n(c) = p \cdot \sum_{u_i} \mathcal{H}^{n-1}(u_i) \cdot P_1(u_i, c) + \frac{1-p}{N}$ $\mathcal{H}^n(u) = p \cdot \sum_{c_i} \mathcal{A}(c_i) \cdot \frac{1}{\mathcal{I}(c_i)} + \frac{1-p}{N}$
35	ha_2	$\mathcal{A}^n(c) = p \cdot \sum_{u_i} \mathcal{H}^{n-1}(u_i) \cdot \frac{1}{\mathcal{O}(u_i)} + \frac{1-p}{N}$ $\mathcal{H}^n(u) = p \cdot \sum_{c_i} \mathcal{A}^{n-1}(c_i) \cdot P_1'(u, c_i) + \frac{1-p}{N}$
36	ha_3	$\mathcal{A}^n(c) = p \cdot \sum_{u_i} \mathcal{H}^{n-1}(u_i) \cdot P_2(u_i, c, l_q) + \frac{1-p}{N}$ $\mathcal{H}^n(u) = p \cdot \sum_{c_i} \mathcal{A}^{n-1}(c_i) \cdot \frac{1}{\mathcal{I}(c_i)} + \frac{1-p}{N}$
37	ha_4	$\mathcal{A}^n(c) = p \cdot \sum_{u_i} \mathcal{H}^{n-1}(u_i) \cdot \frac{1}{\mathcal{O}(u_i)} + \frac{1-p}{N}$ $\mathcal{H}^n(u) = p \cdot \sum_{c_i} \mathcal{A}^{n-1}(c_i) \cdot P_3(u, c_i, l_q) + \frac{1-p}{N}$
38	ha_5	$\mathcal{A}^n(c) = p \cdot \sum_{u_i} \mathcal{H}^{n-1}(u_i) \cdot P_2(u_i, c, l_q) + \frac{1-p}{N}$ $\mathcal{H}^n(u) = p \cdot \sum_{c_i} \mathcal{A}^{n-1}(c_i) \cdot P_3(u, c_i, l_q) \frac{1-p}{N}$
39	ha_6	$\mathcal{A}^n(c) = p \cdot \sum_{u_i} \mathcal{H}^{n-1}(u_i) \cdot P_4(u_i, c, l_q) + \frac{1-p}{N}$ $\mathcal{H}^n(u) = p \cdot \sum_{c_i} \mathcal{A}^{n-1}(c_i) \cdot P_4'(u, c_i, l_q) + \frac{1-p}{N}$

Table 5.2: List of features used for ranking local expert candidates

based, and local authority features capture key characteristics of users and locations of interest, toward assessing local topic expertise of our candidates. Compared to previous work that has only used direct keyword match to find relevant users (e.g., a query for "sports" matches a user who has been labeled with "sports"), we propose to integrate richer topical information from content that users have actually posted (the *tweet content* features). The last class of features – *distance-biased random walk* – is especially important as they naturally integrate expertise propagation into a framework that models the query location, candidate location, and the location of list labelers. Compared to previous works, these distance-biased random walk features model candidates through two perspectives – both as labelers and as labelees – and directly embed distance into the transition probabilities to more robustly model location preferences of users and candidates. In this work, we focus on 39 features, summarized in Table 5.2.

5.3.1 User-Based Features

The first group of features captures user-oriented aspects that are independent of the query topic and query location. These features are simple measures of the popularity, activity, and longevity of each candidate, and can be seen as crude first steps toward capturing how knowledgeable each user is:

- *User Network ($N_{follower}$, N_{friend})*: The first two features measure the number of followers that a candidate has, as well as the number of friends that this candidate has, where friends represent users that is both following and followed by the candidate.

- *User Activity (N_{fav}, N_{status})*: These two features are crude measures of a user's activity-level on Twitter, where N_{fav} capturing the number of favorite tweets the user marked and N_{status} is the number of tweets posted by the user.

- *Longevity (T_{create})*: The final user feature is simply the UTC datetime that an account

was created. In this way, the longevity (or freshness) of the user can be integrated into the ranking model.

5.3.2 Tweet Content Features

Naturally, a more significant contributor to a candidate's perceived expertise is the content that user actually contributes to the community. Hence, we next consider a group of features that seek to summarize a candidate's content. Note that since tweets are inherently limited in size, we aggregate a candidate's posts over each week into a larger pseudo-document.

- *Posting Frequency (twP_c)*: The average number of tweets that the candidate c posted in one week.

- *Tweet Entropy (tw_H)*: The average entropy of a candidate's tweets in each week. This feature shows how informative the candidate's tweets are.

$$tw_H = -\sum_{j=1}^{n_{document}} \sum_{i=1}^{n_{term}} p(t_{ij}) \cdot logp(t_{ij})$$

  where $p(t_{ij})$ is the probability of a term t_i shows up in $document_j$.

- *Topic Bayesian Scores (twB_t)*: The posterior probability $P(topic_j|c)$, which represents the probability of $topic_j$ given candidate c. Here we apply naive bayes method to get the posterior probability. We assume equal prior for topics. The observation $P(t_i|topic_j)$, probability that term t_i appears in $topic_j$, is learnt from a corpus of topic relevant pages crawled from Wikipedia.

- *Location Bayesian Scores (twB_l)*: Similar to Topic Bayesian Scores. The observation $P(t_i|l_j)$, probability that term t_i appears in location l_j, is trained from Wikipedia pages relevant to the location l_j.

5.3.3 List-Based Features

The third group of features extract expertise evidence directly from the Twitter list evidence, but ignoring the geo-spatial features of the lists (those aspects are part of the following two groups of features). Twitter lists have been recognized as a strong feature of expertise in previous work [70]. In particular, lists can shed light on a candidate from two perspectives:

- *Appearing on Lists (N_{listed}, T_{listed})*: On one hand, lists that a candidate appears on will reflect how that candidate is perceived by others. The aggregated information from all lists indicates how well the candidate is recognized.

- *Maintaining Lists (N_{list}, T_{list})*: On the other hand, lists the candidate creates (if any), reflect the candidate's personal interest, which may reflect his expertise. For example, a candidate with a list about food may himself be a foodie.

For these features, we consider all lists as well as a more focused group of on-topic lists (e.g., if the query is for "entrepreneurs", we only consider entrepreneur-related lists; these lists are selected by keywords matching). Moreover, we define a new feature to characterize the quality of a candidate's on-topic lists. This new feature – $list\_score_c$ – is defined as:

$$list\_score_c = \frac{\sum_{i=1}^{N_{on\_topic}(c)} Q_{list}(i)}{N_{on\_topic}(c)}, \quad where\ Q_{list} = \frac{1}{k}\sum_{j=1}^{k} N_{on\_topic}(j)$$

$Q_{list}(i)$ is the quality of i's list and $N_{on\_topic}(c)$ is the number of on-topic lists the candidate is in. Here, Q_{list} represents the average number of times each user in the list has been labeled with the topic of interest and k is the number of users in the list.

5.3.4 Local Authority Features

The fourth set of features focus on the local authority of a candidate, as revealed through the geo-located Twitter lists. The main idea is to capture the "localness" of these lists. Intuitively, a candidate who is well-recognized near a query location is considered more locally authoritative. We measure the local authority of a candidate in multiple ways:

- *Candidate-List Distance (d_c, d_{ct})*: The first two features measure the average distance from candidate c to all the users who appear on c's lists. The main idea here is that a candidate is considered a local expert if she is closer to the people on the lists she maintains. We consider one version that captures all of the lists (d_c) and one that only considers on-topic lists (d_{ct}).

- *Candidate-Labeler Distance (d_u, d_{ut})*: The next two features measure the average distance from a candidate c to all of the labelers of c, capturing the localness of the people who have listed the candidate. Again, we consider one version with all lists (d_u) and one with on-topic lists (d_{ut}).

- *Candidate-Query Distance (d_{uq}, d_{cq})*: These two features measure distance from the query location. The first (d_{uq}) is the average distance from a candidate's labelers to the query location; labelers who are closer to the query location are considered more authoritative. The second (d_{cq}) is the distance from a candidate to the query location; candidates closer to the query location (regardless of whether they have been labeled by locals) are considered more authoritative.

In all cases, we measure distance using the Haversine distance, which gives the great-circle distance around the earth's surface. Apart from these six basic distance features, we also adopt two features used in a previous study of local experts [6]: Candidate Proximity $Prox_c$ and Spread-Based Proximity $Prox_{spread}$.

$$Prox_c(c, l_q) = \left(\frac{d_{min}}{d(c, l_q) + d_{min}} \right)^{\alpha}$$

where $d(c, l_q)$ denotes the Haversine distance between the candidate c's location and the query location l_q, and we set d_{min} = 100 miles. In this case α = 1.01, indicates how fast the local authority of candidate c for query location l_q diminishes as the candidate moves farther away from the query location.

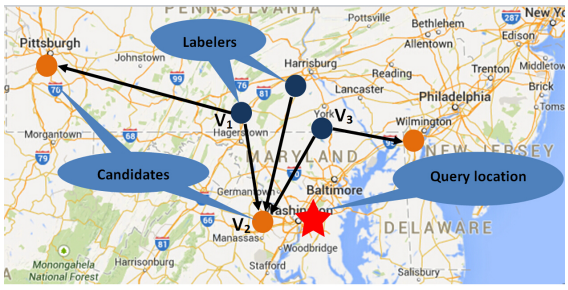
The Spread-Based Proximity captures the average "spread" of a candidate's labelers with respect to a query location:

$$Prox_{spread}(U_c, l_q) = \sum_{u \in U_c} Prox_c(u, l_q) / \mid U_c \mid$$

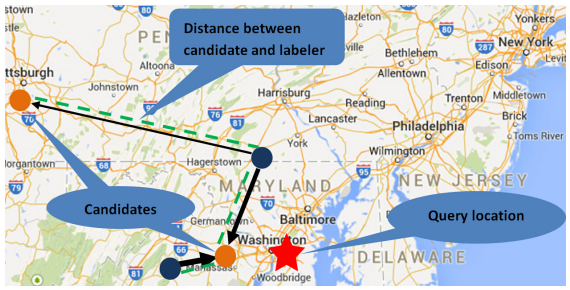
where u denotes one of the labelers U_c of candidate c. The "spread" measure considers how far an "audience" u is from the query location l_q on average. If the "core audience" is close to a query location on average, the candidate gets a high score of $Prox_{spread}$.

5.3.5 Distance-Biased Random Walk Features

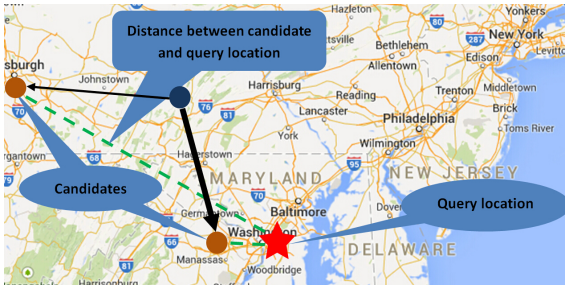
While the previous local authority features consider relationships between a labeler and a candidate, they only consider direct evidence. That is, only the one-hop distance is ever considered. We introduce in this section a set of features that incorporate additional network context beyond these one-hop relationships. Concretely, we explore features based on a random walk model that directly incorporates the location of interest (the query location), the location of a candidate expert, and the location of external evidence of a candidate's expertise (e.g., in the case of the Twitter lists, the location of the list labeler). The main intuition is to bias a random walker according to the distances between these different components (the query location, the labeler, the candidate) for propagating local expertise scores. In this way, each candidate can be enriched by the network formed



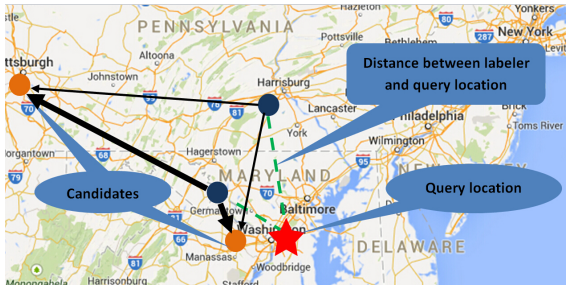
(a) Candidates, labelers, and query location: random walk



(b) Candidate-Labeler Bias



(c) Candidate-Query Location Bias



(d) Labeler-Query Location Bias

Figure 5.2: Three Distance Biases Defined over Random Walk in Twitter List Network

around them via Twitter lists.

5.3.5.1 Baseline Random Walk over Twitter Lists

We begin by modeling a graph based on Twitter lists, which contains the set of users and labeler-candidate relations on topic t, as a directed graph $G = (V, E_t)$. The nodes $V = \{v_1, v_2, ..., v_n\}$ correspond to users. A directed edge $e = (v_1, v_2)_t$, where $e \in E_t$, indicates the presence of a labeler-candidate relation (which we will use listing for short) from user v_1 to user v_2 on topic $t \in T$. Here $E_t = \{e_1, e_2, ..., e_m\}$ represents the set of listings on topic t and $T = \{t_1, t_2, ..., t_p\}$ is the set of topics. Further, each user v_i has an associated location $l(v_i)$.

Over this graph, we can define a transition probability for a random walker to move from one user to the next, either by following forward links or by following backward

links. As a baseline, consider a simple random walker model akin to Kleinberg's Hubs and Authorities [144]. In Figure 5.2a, suppose the walker starts from a user v_1, selects a particular topic t, and then randomly selects a member v_2 in the list to follow the outgoing link (or we say forward link). The random walker further checks the topic t lists of which v_2 is member and reversely follows an incoming link (or we say backward link) to user v_3 who has labeled v_2 in a topic t list. The random walker alternatively follows a forward link and a backward link in a strict manner and continues this process forever. We further incorporate additional randomness: at each step the walker can either follow the links or jump to a random user $v_m \in U$. In summary, we can assign each candidate a local expertise authority score $A^n(c)$ (reflecting local expertise) and each labeler a local expertise hub score $H^n(u)$ (reflecting how well this labeler is a conduit to other local experts):

$$\mathcal{A}^n(c) = p \cdot \sum\nolimits_{u_i} \mathcal{H}^{n-1}(u_i) \cdot \frac{1}{\mathcal{O}(u_i)} + \frac{1-p}{N}$$

and

$$\mathcal{H}^n(u) = p \cdot \sum\nolimits_{c_i} \mathcal{A}^{n-1}(c_i) \cdot \frac{1}{\mathcal{I}(c_i)} + \frac{1-p}{N}$$

where p is the probability of following a link and $1-p$ is the probability of teleporting. $\mathcal{O}(u)$ is the outdegree of a labeler and $\mathcal{I}(c)$ is the indegree of a candidate.

5.3.5.2 Integrating Local Bias

The above random walk is defined for capturing overall authoritative (and hub-like) candidates, without regard for localness. Hence, we propose three approaches for directly integrating distance bias:

Candidate-Labeler Bias: The first approach is to increase the likelihood of following a link to closer candidates (or conversely, to decrease the likelihood of following links to

distant candidates). The intuition here is that a labeler may tend to have stronger knowledge of candidates who are closer, reflecting their local connection. For example, in Figure 5.2b, the probability of transitioning from a labeler to a candidate is higher for closer nodes (represented here by a thicker arrow). The transition probability is lower for more distant nodes (represented by a thinner arrow). Formally, we define these probabilities in the following way: Let $d(u, c)$ be the Haversine distance between labeler and candidate, D_1 is a non-linear mapping in the form of Candidate Proximity discussed in Section 5.3.4, which maps $d(u, c)$ to a real value in [0,1]. Other mappings are possible, but we find good results using this approach. The probability of following a forward link from u to c is defined as:

$$P_1(u, c) = \frac{D_1(d(u, c))}{\sum_{c_j : u \to c_j} D_1(d(u, c_j))}$$

The probability of following a backward link from c to u is defined as:

$$P_1'(u, c) = \frac{D_1(d(u, c))}{\sum_{u_j : c \to u_j} D_1(d(u_j, c))}$$

Candidate-Query Location Bias: The second approach is to increase the likelihood of following a forward link to a candidate who is closer to query location (and conversely, to decrease the likelihood of following links to candidates who are distant to the query location). The intuition is that a candidate who is closer to query location is more likely to be knowledgeable about the topic at the query location. For example, in Figure 5.2c, we see that the probability of transitioning to the candidate close to the query location is larger (represented by the thicker arrow), versus the probability of transitioning to the other candidate (represented by the thinner arrow). Formally, let $d(c, l_q)$ be the Haversine distance between the location of candidate and query location, D_2 maps $d(c, l_q)$ to a real value to a real value in [0,1], where D_2 is a mapping similar to D_1. The probability of

following a forward link from u to c is defined as:

$$P_2(u, c, l_q) = \frac{D_2(d(c, l_q))}{\sum_{c_j:u\to c_j} D_2(d(c_j, l_q))}$$

Labeler-Query Location Bias: The third approach is to increase the likelihood of following a backward link to a labeler who is closer to the query location (and conversely, to decrease the likelihood of following a backward link to a labeler who is distant to the query location). The intuition is that a labeler who is closer to query location is more likely to label high quality local expert candidates at the query location. For example, we can see in Figure 5.2d that the labeler who is closer to the query location has a higher probability associated with that edge (as represented by the thick arrow). Otherwise, the probability is small (as represented by the thin arrow). Formally, let $d(u, l_q)$ be the Haversine distance between labeler and query location, D_3 maps $d(u, l_q)$ to a real value in [0,1], where D_3 is a mapping similar to D_1. The probability of following a backward link from c to u is defined as:

$$P_3(u, c, l_q) = \frac{D_3(d(u, l_q))}{\sum_{u_j:c\to u_j} D_3(d(u_j, l_q))}$$

Combining Bias Factors: Finally, we can combine the different bias factors in various ways. For illustration, assume we want to combine all three bias factors: $\{d(c, l_q), d(u, l_q), d(u, c)\}$. We can define the probability to follow a forward link as:

$$P_4(u, c, l_q) = \frac{D_1(d(u, c)) \cdot D_2(d(c, l_q))}{\sum_{c_j:u\to c_j} D_1(d(u, c_j)) \cdot D_2(d(c_j, l_q))}$$

and in the same fashion, we can get the probability P_4', which characterizes the probability of following a backward link.

Finally, we can embed these different distance-bias factors into the local expertise authority score $A^n(c)$ described above to generate a series of new features. Specifically, we

generate seven new features based on the distance-biased random walk (as shown in Table 5.2). ha_0 uses the original setting with no distance influence. ha_1 and ha_2 takes the distance between labeler and candidate into account ($P_1(u, c)$ and $P_1'(u, c)$). ha_3 considers the distance between the location of a candidate and the query location ($P_2(u, c, l_q)$). ha_4 considers the distance between the labeler and the query location ($P_3(u, c, l_q)$). ha_5 considers how distant a candidate ($P_2(u, c, l_q)$) and the labelers ($P_3(u, c, l_q)$) are from the query location. Finally, h_6 considers the distance between each pair of the three entities ($P_4(u, c, l_q)$).

5.4 Evaluation

In this section, we present the experimental setup, including the collection of ground truth data via AMT, alternative local expert ranking methods, and metrics for comparing these methods. We then report on a series of experiments designed to answer the following questions: How does the learning-based local expert ranking approach compare to existing methods? How stable are the results across different topics and locations? What features are most important for identifying local experts? Can a local expert model trained on one topic generalize to other topics?

5.4.1 Experimental Setup

Our experiments rely on the dataset described in Section 5.2.2, totaling 15 million geo-tagged list relationships.

Queries. We adopt a collection of eight topics and four locations that reflect real information needs. The topics are divided into broader local expertise topics – "food", "sports", "business", and "health" – and into more specialized local expertise topics which correspond to each of the broader topics – "chefs", "football", "entrepreneurs", and "healthcare". The locations are New York City, San Francisco, Houston and Chicago, which all have relatively dense coverage in the dataset for testing purposes.

Retrieving Candidates. For each method tested below, we retrieve a set of candidates for ranking based on topics derived from list names. For each list name, we apply tokenization, case folding, stopword removal, and noun singularization. We separate string patterns like "FoodDrink" into two tokens "food" and "drink". We consider each of the remaining keywords as a *topic*. Finally, each candidate is associated with all topics derived from this process, resulting in a set of potential candidates to be ranked.

Proposed Method: Local Expert Learning (LExL). There are a wide variety of learning-to-rank approaches possible; in this chapter we evaluate four popular learning-to-rank strategies: Ranknet, MART, Random Forest and LambdaMART. We use an open source implementation of these methods in the RankLib toolkit. While we previously introduced LambdaMART, here we briefly introduce these three other variations of LExL. Ranknet is a pairwise learning to rank method, where each pair of the candidates is considered together to form a positive or a negative instance. The cost function of Ranknet aims to minimize the number of inversions in ranking. MART is based on the idea of boosting and it uses gradient boosted decision trees for prediction tasks. The prediction model is a linear combination of the outputs of a set of regression trees. Random Forest is an application of bagging, which can substantially improve the quality of probability estimates in almost all domains [145]. However bagging has two disadvantages – greater computation cost and loss of comprehensibility. Note that LambdaMART is a specific instance of MART that evolved out of a combination of Ranknet and MART. For each topic, we randomly partition the collected candidates together with their five categories of features into four equal-sized groups for training and testing. We use four-fold cross validation for reporting the results. We compare our proposed approach with two state of the art approaches for finding local experts:

- **Cognos+ [70].** The first baseline method is the Cognos expert ranking scheme.

Cognos was originally designed for identifying general topic experts, so the ranked lists from Cognos are independent of query location. Hence, we modify Cognos by incorporating a distance factor when calculating cover density ranking [146], where each label is weighted by a distance factor range in [0,1] in a similar way as in Candidate Proximity discussed in Section 5.3.4. We refer to this location-sensitive version of Cognos as Cognos+.

- **LocalRank [6].** The second baseline method is the LocalRank framework proposed in [6]. This framework ranks candidates by a linear combination of local authority and topical authority. We choose the best performing combination reported in that paper – spatial proximity plus direct labeled expertise (SP+DLE) – as the baseline to compare against.

Note that both of these alternative methods are unsupervised, whereas the learning-based approach proposed here integrates labeled training data to bootstrap the ranker. Naturally, we would expect the supervised approach to perform well; our goal here is to measure this improvement as well as investigate the key factors for this improvement.

5.4.2 Gathering Ground Truth

Since there is no publicly-available data that directly specifies a user's local expertise given a query (location + topic), we rely on an evaluation based on ground truth by employing human raters (turkers) on Amazon Mechanical Turk to rate the level of local expertise for candidates via human intelligent tasks (HITs).

5.4.3 Pooling Strategy

It is too expensive to manually label the local expertise for every candidate with each query pair (location + topic). Moreover, many candidates are irrelevant to the query location and do not possess expertise on the topic of interest. Hence, a pooling strategy is

adopted to improve the effectiveness of obtaining relevance judgments by reducing the number of irrelevant candidates presented to turkers to improve the effective utilization of turkers [147]. In order to build the pool of local expert candidates, the candidate set is sampled for each query pair, which only considers the candidates who appear at least one on on-topic list. 100 candidates are selected for each query pair, and then they are randomly assigned to different HITs.

5.4.4 HIT Design

Each HIT includes instructions and examples of local expertise along with twelve candidates to judge. Turkers can access the information about the query pair and a link to a candidate's Twitter page including account profile, recent tweets, lists and home location. Turkers are then asked to rate each candidate's local expertise on a five-point scale, corresponding to no local expertise (0), difficult to tell (1), a little local expertise (2), some local expertise (3), and extensive local expertise (4). The topic and location are kept the same within a single HIT, so the turkers can get familiar with the style of HITs with the task and make more consistent judgments. Several evaluation criteria [147] are adopted in order to collect high accuracy and reliable ground truth judgments. First, two out of the twelve candidate's are set as trap questions, where we have already judged the candidates as either clearly local experts (4) or obviously having no local expertise (0). These trap candidates are chosen to identify turkers who give random judgments or make judgments only by the candidate's home location (e.g., quickly assigning high scores to candidates whose locations are Houston in the map instead of looking at their Twitter information for a task seeking local experts on Houston healthcare). We also maintain a turker qualification type in AMT which only allows turkers whose results are consistently in good quality to continue working on our HITs. For each candidate, we collect five judgments from distinct turkers and the majority judgment is taken as the final local expertise rating;

Topic	**Accuracy**	κ **value**
food	0.6845	0.5320
sport	0.7889	0.4903
business	0.7119	0.4596
health	0.7639	0.4461
chef	0.6640	0.4679
football	0.7758	0.3834
entrepreneur	0.7128	0.2868
healthcare	0.7675	0.5952
Average	0.7337	0.4576

Table 5.3: Turker agreement for topics

if there is a tie in the vote, the ceiling of the average is taken as the final rating.

5.4.5 Turker Agreement

After run the HITs experiments, 16k judgments were collected in total across the eight topics and four locations based on the above settings. But are these assessments of local expertise reliable? To answer this, the *accuracy* and the *kappa statistic* [148] are calculated to explore the validity of turker judgments. The accuracy for a candidate c given a query pair q is defined as

$$Accuracy(c, q) = \frac{No.\ of\ majority\ judgments}{No.\ of\ judgments}$$

Accuracy ranges from 0 to 1, with 0 meaning every judgment for the candidate is unique and agrees with no other judgment, and 1 meaning all raters give a consistent judgment for the candidate. The kappa statistic also measures inter-rater reliability, ranging from 0 to 1, with larger values indicating more consistency in the judgments. In Table 5.3, we show the accuracy and kappa values for each topic, where we treat local expertise scores of 2, 3, and 4 as relevant, and scores of 0 and 1 as irrelevant. The average accuracy across all topics is 0.74, which indicates that around 3 out of 4 raters agree on whether one candidate is a local

expert. The accuracy is higher in some topics (e.g., football), indicating that assessing local expertise may be inherently easier in some cases. For kappa, an average of 0.46 means "moderate agreement." As in the case of accuracy, there is variability in the scores, with the topic entrepreneur being the most controversial topic to judge and healthcare being the easiest.

5.4.6 Evaluation Metrics

To evaluate the quality of local expertise approaches, three metrics are adopted across all experiments: Rating@k, Precision@k and NDCG@k.

Rating@k measures the average local expertise rating for a query pair to output the top-k experts for each approach, defined as:

$$Rating@k = \sum_{i=1}^{k} rating(c_i, q)/k$$

where c is candidate and q is the query pair. In our scenario, k=10. The Rating@10 here ranges from 0 to 4, where a value of 4 says the majority of the raters believe everyone of the top 10 experts found by the local expertise method has extensive local expertise. Since Recall will calculate the fraction of relevant ratings that are retrieved based on all Turkers' judgements across query topics and locations, the value of Recall won't change for different learning methods and different sets of features. Thus we utilize Rating@k instead of Recall.

Precision@k measures the percentage of the top-k suggested local experts that are actually local experts. Here, candidates with a rating 3 or 4 are considered as relevant; all others are irrelevant. Note that this is a more conservative approach than the one for inter-judge reliability; we want more distinguishing power deployed between approaches for

comparing local expertise methods.

$$Precision@k = \sum_{i=1}^{k} r_i/k$$

where $r_i = \begin{cases} 1 & \text{if rating}(c_i,\text{q}) \geq 3 \\ 0 & \text{else} \end{cases}$

NDCG@k compares how close each method's top-k ranking order of local experts is to the ideal top-k ranking order.

$$NDCG@10 = \frac{DCG@10}{IDCG@10}$$

$DCG@10 = \sum_{i=1}^{10} \frac{2^{rating_i}-1}{\log_2(i+1)}$ and $IDCG@10 = \sum_{i=1}^{10} \frac{2^{rating_i^{'}}-1}{\log_2(i+1)}$. $rating_i$ represents the actual rating of the candidate in position i, $rating_i^{'}$ represents the rating of the candidate in position i given the ideal decreasing ranking order of all candidates. $DCG@10$ is the discounted cumulative gain (DCG) of the learned ranking order until position 10 and IDCG@10 is the maximum possible DCG up to position 10.

5.4.7 Results

5.4.7.1 Comparison versus Baselines

We begin by comparing the proposed learning method (LExL) versus the two baselines. Figure 5.3 shows the Precision@10, Recall@10, and NDCG@10 of each method averaged over all queries.[2] We consider the LambdaMART version of LExL, in addition to methods using Ranknet, MART and Random Forest. First, we observe that three versions of LExL clearly outperform all alternatives, resulting in a Precision@10 of more

[2]Note that the results reported here for LocalRank differ from the results in [6] as the experimental setups are different. First, our rating has 5 scales, which is intended to capture more detailed expertise level. Second, [6] only considers ideal ranking order for the top 10 results from LocalRank when calculating IDCG@10, while we consider a much larger corpus.

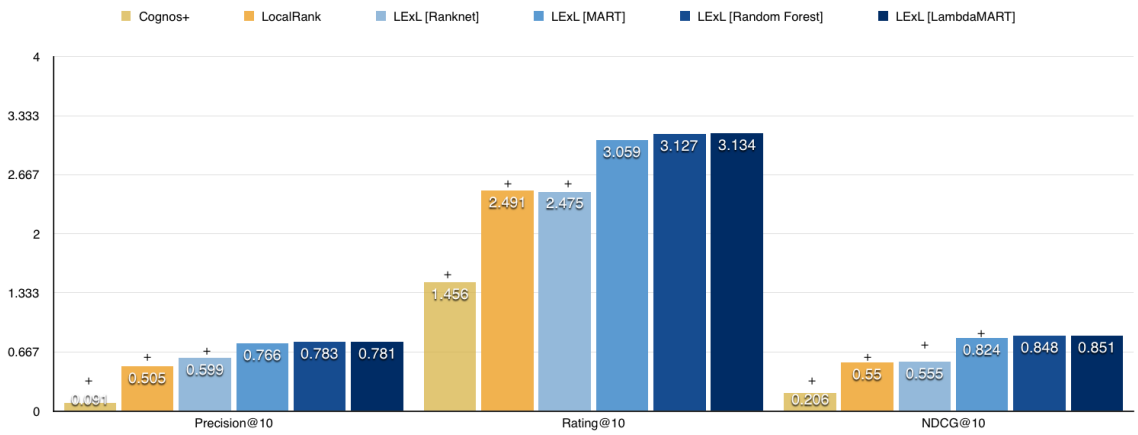


Figure 5.3: Evaluating the proposed learning-based local expertise approach versus two alternatives. '+' marks statistical significant difference with LExL [LambdaMART] according to paired t-test at significance level 0.05.

than 0.76, an average Rating@10 of around 3.1, and an NDCG of around 0.84.

Cognos has been shown to be effective at identifying general topic experts. However, we see here that even a modified version to include a distance factor is not compatible with local expert finding. For example, Cognos may identify a group of "healthcare" experts known nationwide, but it has difficulty uncovering local experts.

LocalRank has a much better Precision@10 of around 0.5 compared to Cognos+, which indicates that 50 percent of the candidates it identifies have at least "some local expertise" for the query. The average Rating@10 is 2.49, which means the candidates are generally rated between "a little expertise" and "some expertise". Since LocalRank explicitly builds on both topical and local signals (by exploiting the distance between a candidate's labelers and the query location), it performs much better than Cognos+. However, LocalRank is only a linear combination of these two factors, and so does not exploit either additional factors (like the random walk presented in this chapter) nor take advantage of a learning approach for optimizing the weighting of these factors.

For the four LExL approaches, Ranknet performs comparably to LocalRank, but the

remaining three all result in significantly better performance, with both Random Forest and LambaMART achieving comparably good results. These two methods have a Rating@10 of around 3.1, indicating that the local experts discovered have from "some local expertise" to "extensive local expertise". The Precision@10 and NDCG@10 also support the conclusion that these learning-based methods result in high-quality local experts. Since LambdaMART is significantly less computationally expensive ($\sim$1/6 of the computing time of Random Forest), we adopt it for the remainder of the chapter.

5.4.7.2 Effectiveness Across Topics and Locations

Given the good performance of LExL with LambaMART, we next turn to comparing the effectiveness of this approach across the four general topics and four narrower topics, before turning to a location comparison in the following discussion. Is the effectiveness of local expert finding consistent across topics? And does it vary by the specificity of the topic?

We observe in Table 5.4 that NDCG@10 is consistently high for the four general topics, with an average value of 0.8212. Precision@10 and Rating@10 are also consistent for general topics except for the topic of "health" which has relatively low values. We attribute this poor showing due to data sparsity: (i) First, through manual inspection we find that there are inherently only a limited number of candidates with high local expertise for the "health" topic in the training and testing datasets. (ii) Second, since we only consider candidates with "some local expertise" and "extensive local expertise" as good matches for a query, this additionally reduces the number of possible local experts. However, since the learning framework is effective at identifying even those few local experts in "health" we see a high NDCG@10.

We observe comparable results for the four narrower topics. The Precision@10 is lower than for the general topics (0.75 versus 0.81), but the NDCG@10 is higher (0.88

Topics	Precision@10	Rating@10	NDCG@10
food	0.8250	3.125	0.7442
sports	0.9152	3.225	0.9054
business	0.9237	3.368	0.8506
health	0.5873	3.059	0.7847
chefs	0.8233	3.163	0.9044
football	0.7283	2.933	0.9002
entrepreneurs	0.7377	3.040	0.7489
healthcare	0.7100	3.166	0.9673
General topic AVG	0.8128	3.193	0.8212
Subtopic AVG	0.7498	3.075	0.8802

Table 5.4: Quality of local expert rankings across topics

Locations	P@10	R@10	NDCG@10
Houston	0.7214	2.917	0.8473
Chicago	0.7788	3.244	0.8486
New York	0.7875	3.294	0.8501
San Francisco	0.7563	3.081	0.8580

Table 5.5: Quality of local expert ranking in different locations

versus 0.82). Part of the higher NDCG results may be attributed to the decrease in the denominator of NDCG for these narrower topics (the Ideal DCG), so the ranking method need only identify some of a pool of moderate local experts rather than identify a few superstar local experts.

In a similar fashion, we evaluate the quality of LExL across the four query locations, as shown in Table 5.5. For the most part, the Precision@10, Rating@10, and NDCG@10 show good consistency across these four locations, as Chicago, Houston, New York and San Francisco, suggesting the potential of a learning-based method to identify factors associated with each location for uncovering local experts.

Features	Precision@10	Rating@10	NDCG@10
User-based	0.6714†	2.748†	0.6955†
Tweet Content	0.6804†	2.730†	0.6973†
List-based	0.6839†	2.807†	0.6824†
Local Authority	0.7386†	3.002†	0.7662†
DistBRW	0.7431†	2.995†	0.7734†
All Features	0.7813	3.134	0.8507

Table 5.6: Quality of local expert ranking using different sets of features. '†' marks statistical significant difference with LExL[LambdaMART] according to paired t-test at significance level 0.05.

5.4.8 Evaluating Feature Importance

Given the strong performance of the learning approach for local experts, what is the significance of the different kinds of features used for learning? Recall that the learning model is built upon five kinds of features – user-based, tweet content, list-based, local authority, and distance-biased random walk (DistBRW) features. To assess the importance of these different features, we train four different LExL models, one for each feature type. For example, the model is trained only using user-based features and then evaluate the quality of local experts identified.

We can see from Table 5.6, the five feature classes result in varying levels of local expert quality. The user-based, list-based and tweet content features perform relatively well (especially when compared to LocalRank), though not as well as the local authority and DistBRW features. These results suggest that intelligent combinations of many features via a learning method can outperform a simple combination of two carefully selected features (as in LocalRank). The DistBRW features achieve the highest Precision@10 and NDCG@10 among all five kinds of features. We attribute the results to DistBRW integrating expertise propagationas well as distance bias factors into capturing local expertise. We can observe that the combination of all features performs the best of all. One more

interesting finding is that Tweet Content features didn't perform as well as we expected. This can be explained by that the keywords related to topic and location in tweet content didn't appear very often. For example, an NFL quarterback may share his feelings about life more often than his expertise topics of football. Thus, we can conclude that tweet content is not the determining factor of finding local experts.

But which specific features are most informative, regardless of feature category? Here, we adopt two different feature selection methods to identify the most informative features for local expert ranking.

- **Recursive Feature Elimination (RFE).** In this approach, a linear regression model is trained and weight is assigned to each feature. Then features with the smallest absolute weight are eliminated. For each topic, we keep eliminating the unimportant features until only required number of features are left.

- **Tree-Based Feature Selection.** In this approach, a number of randomized decision trees are built on various sub-samples of the dataset. The importance of a feature is determined by *Gini importance* or *Mean Decrease Impurity*, which is defined as the total decrease in node impurity of all trees in the ensemble [149]. Features that attach to a node with higher *Gini importance* are more informative in the model.

Table 5.7 shows the accumulated number of times that each feature is selected by the two different feature selection methods. For 39 features, most of the top features are from local authority features and DistBRW features.

The results is aggregated across all queries (topics + locations), and reported in Table 5.8 the top features for each feature importance method. There are seven common top features across both methods, which is highly consistent. Recall that d_{cq} and d_{ut} capture the distance from candidate to query location and the average distance form candidate to all on-topic labelers, respectively. $Prox_c$ shows the distance between the location of

Feature	REF	Tree-based	Feature	REF	Tree-based
$N_{follower}$	0	2	N_{listed}	1	2
N_{friend}	1	0	T_{listed}	4	2
N_{fav}	0	0	N_{list}	3	0
N_{status}	0	0	T_{list}	5	1
T_{create}	2	0	$list\_score_c$	1	6
			d_c	1	0
twP_c	0	0	d_{ct}	3	0
tw_H	0	0	d_u	3	1
$twB_{business}$	1	0	d_{ut}	4	7
$twB_{entrepreneur}$	0	0	d_{uq}	2	1
twB_{food}	1	1	d_{cq}	8	8
twB_{chef}	0	0	$Prox_c$	7	8
twB_{sport}	0	0	$Prox_{spread}$	6	6
twB_{food}	1	1	ha_0	2	5
twB_{health}	0	0	ha_1	5	4
$twB_{healthcare}$	0	0	ha_2	2	4
twB_{chi}	0	0	ha_3	2	2
twB_{hou}	0	0	ha_4	7	6
twB_{ny}	0	0	ha_5	2	6
twB_{sf}	0	0	ha_6	8	7

Table 5.7: Accumulated Times of Features Selected by Different Methods

Table 5.8: Individual feature importance

Method	**Top-10 Features**
RFE	d_{cq}, ha_6, $Prox_c$, ha_4, $Prox_{spread}$, T_{list}, ha_1, T_{listed}, d_{ut}, N_{list}
Tree-based	d_{cq}, $Prox_c$, ha_6, d_{ut}, ha_4, $Prox_{spread}$, ha_5, $list\_score_c$, ha_0, ha_1

Method	**Precision@10**	**Rating@10**	**NDCG@10**
RFE	0.7612	3.059	0.8357
Tree-based	0.7753	3.078	0.8444
All Features	0.7813	3.143	0.8507

Table 5.9: Performance using selected features

candidate and query location and ha_6 is the steady state local expertise authority score of candidate incorporating all three distance influences. Based on the selection results, we can say comparing to the tweet content, a candidate's list and location-related features are more decisive for finding local experts.

Ultimately, how explanatory are these features? We further train two additional LExL models – one using the top-10 features from the RFE feature importance method and one using the top-10 features using the tree-based method. Table 5.9 shows the evaluation metrics for these two approaches versus the full blown LExL model with all features. We can observe the difference of Precision@10 and Rating@10 is about 0.02 and NDCG@10 is about 0.1 for both methods compared to All Features case. Moreover, the difference between the value of three evaluation metrics and those of the All Features' case is not statistically significant. These results confirm the importance of the local authority features and the DistBRW features and further show that high-quality local expert models may be built using fairly compact features.

5.4.9 Generalizability of Local Expert Models

Finally, can a learnt model be reused to rank other topics? The generalizability of the local expert models is explored in this section. In many cases, we can imagine building a local expert model that is optimized for one type of topic (e.g., healthcare) but then we want to apply the model to a different topic (e.g., finance), for example in cases where training data is unavailable or expensive to collect. Are the models of local expertise generalizable enough to support high-quality local expert finding in these new topic areas? Or do the key features and feature weightings vary from topic to topic, so that a specialized model must be built for each topic?

The first experimental setup here is to train a model on each of four topics and then to apply this model to a different topic. Concretely, we train over the four general topics – health, food, business, and sports – and then rank candidates in each of the four narrower topics – healthcare, chefs, entrepreneurs, and football. Intuitively, a model trained over a related topic is expected to perform better than a model trained over a less similar topic (e.g., a health-based local expert model should do better for healthcare, but worse for football). But does this hold? And how well do the less related models perform?

The results of this experiment are shown in Table 5.10. For each of the four narrower topics, that indeed, the model corresponding to the most related general topic produces the best results. Perhaps surprisingly, these models perform on par with the models trained over the individual topics as in Table 5.4, or even better in Precision@10 and Rating@10. Since the general topic models build a more broader measure to define a local expert than the individual models of narrower topic, it can rank the more related candidates higher, which leads to high Precision@10 and Rating@10.

Even for models built on very different topics, we do see encouraging results. For example, the sports-based model for ranking chefs results in Precision@10 of 0.86, Rat-

Topic	Model	Precision@10	Rating@10	NDCG@10
chefs	health	0.8312	3.094	0.7363
	food	**0.8738**	**3.152**	**0.8012**
	business	0.8250	3.131	0.6644
	sports	0.8687	3.075	0.7963
	general	0.8633	3.125	0.8112
	original	0.8233	3.163	0.9044
football	health	0.6987	2.781	0.7910
	food	0.6125	2.618	0.5718
	business	0.6437	2.731	0.5934
	sports	**0.7083**	**2.875**	**0.8026**
	general	0.7014	2.925	0.8473
	original	0.7283	2.933	0.9002
entrepreneurs	health	0.7033	2.816	0.6247
	food	0.7166	2.825	0.6018
	business	**0.7511**	**2.950**	**0.7144**
	sports	0.7433	2.841	0.6432
	general	0.7866	2.966	0.6906
	original	0.7377	3.040	0.7489
healthcare	health	**0.7037**	**3.118**	**0.8844**
	food	0.6687	2.975	0.6443
	business	0.6875	3.002	0.7570
	sports	0.6650	3.090	0.8229
	general	0.6833	3.050	0.8847
	original	0.7100	3.166	0.9673

Table 5.10: Applying a model learned on one topic to rank local experts on a different topic.

ing@10 of 3.1, and NDCG@10 of 0.80 and the health-based model for ranking football results in Precision@10 of 0.69, Rating@10 of 2.8, and NDCG@10 of 0.79. These results indicate the potential of learning models that can be extended to new local expert topics.

In the second experiment, instead of having a model for each general topic, we train a single model for four general topics altogether, and then test this model on each subtopic. In Table 5.10 that the general model performs no worse than each individual model. This is attributed to more training data and avoidance of overfitting to one topic. It indicates we may find a common local expert model that is applicable regardless of the specific topic.

5.5 LExL System Design

We conclude with a discussion of issues impacting the deployment of LExL. The discovered local experts can be integrated into a variety of systems – including (i) recommender systems that can be biased to prefer the opinions and rating of the discovered local experts; (ii) local question-answer systems where local queries are routed to local experts; and (iii) locally-flavored information streams like the Facebook newsfeed and the Twitter stream, where posts can be re-ranked based on the local expertise scores of who is posting, among many other possible application scenarios. In the following, we raise some key points that can impact the success of deploying LExL into these scenarios, including efficiency, incentives, and some domain-specific concerns.

5.5.1 Efficiency

First, the complexity of learning a ranking model using the LambdaMART algorithm is $O(m \cdot N^2)$, where N is the number of training samples and m is the number of trees. As we have seen through experiments, a generalized ranking model can be built that performs competitively for other query locations and query topics, meaning that one strategy would be to build the models entirely offline. In this way, we could perform rigorous offline optimization (including identifying key sets of powerful features), such that the real-time

assessment of new candidates would require only the generation of features for that candidate and then ranking the candidates according to the model given features. For each query topic, we should keep track of query relevant users, together with their list features. To update the system with new expert candidates, we can collect users together with their features at regular time intervals to refresh the candidates. Most of the features can be acquired in constant time, as they can be pre-computed offline, for example, user-based features, the majority of the local authority features, and tweet content features. For the distance-biased random walk features, all seven of them can be updated simultaneously and we empirically set the number of iteration to a constant c that less than 100, for the algorithm to converge. Hence, the complexity for feature construction is thus $c \cdot O(n)$, where n represent the number of candidates. The ranking itself takes $O(n \cdot logn)$. Overall, the complexity is $O(n \cdot logn)$.

For application scenarios that require real-time local expert discovery, there are many ways to further decrease the response time required. For example, we could selectively lower the location accuracy, and report results for nearby cities. Separately, we could cache results for some frequent query locations so that query issuers can get immediate responses. Additionally, we can set a distance threshold such that distant candidates are not considered to decrease the computation cost for feature collection and ranking.

5.5.2 Incentives

A second key concern is incentives. In some application scenarios – for example re-ranking an information stream or recommending based on local experts – we can mine evidence of local experts without directly engaging with them. In other scenarios – for example, a local expert powered question-answer system – incentivizing local experts to participate is critical. Indeed, we have seen a wide variety of efforts that target incentive schemes in crowdsourcing systems. For example, some systems are built around tasks

that are inherently fun – like the ESP game and peekaboom – to encourage participation. Some sites like Quora, Stackoverflow and Wikipedia can attract and retain participants for several reasons: First, the inherent inclination towards human interactions, and community work on similar thing create emotional bonds. Second, on the one hand, users have a sense of accomplishment as their knowledge and experience are of immediate use and value to the individual participants and on the other hand, by supporting the system for others, the system will likely return the favor in an area that users need. Thirdly, the motivation for sites like Wikipedia mainly comes from one's academic interest and charity, and some people ask for no returns and simply for the sake of its success. Of course, the most direct and effective approach is offering rewards. For example, some systems provide a badging system or similar accumulation of reputation points that can serve to incentivize participation. Money is a great motivator for prominent crowdsourcing platforms like Amazon Mechanical Turk and Crowdflower.

In our continuing work, we are interested to combine several of these features for encouraging local experts to participate in a prototype question-answer system. Concretely, we are interested to make answering a combination of reward motivated and self-motivated. When people have some instant information needs, they are willing to pay some money and get precise and informative response quickly. Also the award money will encourage experts to answer. Meanwhile, we are also expecting to see a more natural eco-system where users are eager to help each other and seek and contribute knowledge.

5.5.3 Question-Answer Issues

For a local expert question-answer system, we will require additional system design. For example, how do we properly match the question with the appropriate level of expertise? How do we match similar questions so that previously answered questions can be reused to save system resources? Do the features we have identified – including list-based

and distance-biased random walk features – degrade as we consider ever more granular questions? We anticipate building on the large body of related work in general question-answer systems [150, 151, 152, 153] to explore the factors impacting local expert powered question-answer systems.

Query routing. For example, to avoid forwarding all questions to the top-ranked expert, we can use a round robin strategy or some other traffic scheduling scheme to divide the user queries to different experts. We can also add features that characterize the time required for each expert to reply and incorporating this urgency of the question into how the query is routed.

Rare queries. In some cases, a rare query may not match any local expert at all. In these cases, we can default to a broader set of local experts by leveraging a user-generated knowledge base (e.g., a folksonomy built over local experts). For example, if there are few experts for expert in "leveraged buyouts", we can relax the query to a broader category of "finance".

5.6 Summary

In this chapter, we have proposed and evaluated a geo-spatial learning-to-rank framework for identifying local experts that leverages the fine-grained GPS coordinates of millions of Twitter user and carefully curated Twitter list data. We introduced five categories of features for learning model, including a group of location-sensitive graph random walk features that captures both the dynamics of expertise propagation and physical distances. Through extensive experimental investigation, we find the proposed learning framework which intelligently assigns feature weights can produce significant improvement compared to previous methods. By investigating the features, we found high-quality local expert models can be built with fairly compact features. Additionally, we observed promising results for the reusability of learned models. In our future work, we seek to combine data

from different social media platforms to better deal with sparsity. Furthermore, we are interested in investigating what new features may provide complementary evidence of local expertise.

6. RECOMMENDATION: GEO-SOCIAL IMAGE RECOMMENDATION[1]

In this chapter we turn from location-sensitive retrieval to its flipside: location-sensitive recommendation. We seek to model and leverage geo-social characteristics in the broad category of multimedia recommendation by capturing individual user preferences. In particular, we tackle the problem of personalized image recommendation in geo-social systems.

6.1 Introduction

One of the foundations of many web and app-based communities is *image sharing*. For example, Pinterest, Facebook, Twitter, Flickr, Instagram, and Snapchat all enable communities to share, favorite, re-post, and curate images. And yet, these social actions are far outnumbered by the total number of images in the system; that is, there may be many valuable images undiscovered by each user. Hence, considerable research has focused on the challenge of *image recommendation* in these communities, e.g,. [80, 81, 82, 83, 84, 85, 86].

However, many of these works mainly leverage user profile and behavior patterns. Due to the extreme sparsity of users feedback in image sharing communities and a lack of proper representation, traditional recommendation including collaborative filtering and content-based methods face challenges. In contrast, Bayesian Personalized Ranking (BPR) has shown state-of-the-art performance for recommendation in implicit feedback datasets [79]. Yet, there exists some limitations: (i) First, user preferences in BPR are calculated as the inner product of user latent vector and image latent vector, which assigns equal weight

[1]Part of this chapter is reprinted with permission from "Neural Personalized Ranking for Image Recommendation" by Wei Niu, James Caverlee and Haokai Lu. 2018. The 11th ACM International Conference on Web Search and Data Mining (ACM WSDM '18). DOI: https://10.1145/3159652.3159728. Copyright 2018 by Association for Computing Machinery (ACM).

to each dimension of the latent feature space, meaning the variability of user preferences may not be adequately captured; (ii) Second, the matrix factorization component of BPR is linear in nature, which has limited expressiveness when compared to nonlinear methods; and (iii) existing effort for distributed BPR typically uses partially shared memory which may limit its scalability.

Recently, deep neural network based approaches have shown tremendous success in vision related problems and NLP problems, and we begin to see new advances applying deep neural network models for recommendation, yet most of the work focuses on modeling side information [154, 155] or building pointwise learning models by directly modeling user ratings [100, 102]. As far as we know, none has considered modeling pairwise ranking incorporating generalized matrix factorization. Additionally, contextual information are proven to be helpful for improving the performance of recommendation tasks. However, there is a research gap in understanding how users' preference to images are reflected on geo, topic and visual contextual information and how to incorporate such clues, if any, into a neural network based pairwise ranking model.

To overcome these challenges, we propose *Neural Personalized Ranking (NPR)* – a new neural network based personalized pairwise ranking model for implicit feedback, which incorporates the idea of generalized matrix factorization. Neural models promise potentially more flexibility in model design, added nonlinearity through activations, and ease of parallelization. While recent work in neural methods for recommendation has focused on modeling side information [154, 155] or building pointwise learning model by directly modeling user rating [100], a key feature of NPR is its careful modeling of users' implicit feedback via a relaxed assumption about unobserved item using pairwise ranking that builds on top of neural network based generalized matrix factorization components. Further, to alleviate the sparsity of user feedback and improve the quality of recommendation, we propose to leverage multiple categories of contextual information. We further

build an enhanced model by augmenting the baseline NPR model with multiple contextual preference clues and derive corresponding features for deriving *Contextual Neural Personalized Ranking (C-NPR)* to better uncover user preferences. In particular, these preference clues include user tags, geographic features, and visual factors.

In our experiments over the Flickr YFCC100M dataset, we demonstrate the proposed NPR model's effectiveness in comparison to several state-of-the-art approaches. Moreover, the contextual enhanced NPR model significantly outperforms the baseline model by 16.6% and a contextual-BPR model by 4.5% in precision and recall. We find that NPR is far more effective than BPR when there is only limited training data, as is often the case in real-world scenarios.

6.2 Preliminaries

Our goal is to provide *personalized image recommendation*, such that each user is recommended a personalized list of images.

Problem Statement. Formally, we assume a set of M users $\mathcal{U}$=$\{u_1, u_2, ..., u_M\}$ and a set of N images $\mathcal{I}$=$\{i_1, i_2, ..., i_N\}$. We further assume some users have explicitly expressed their interest for a subset of the images in $\mathcal{I}$. This preference may be in the form of a "like" or similar social sharing function. We aim to recommend for each user a personalized list of images from the set $\mathcal{I}$.

6.2.1 Matrix Factorization

Toward tackling the problem of *personalized image recommendation*, we begin with a straightforward adaptation of latent factor matrix factorization (MF) [156]. The standard formulation is: the preference r_{ui} of a user u towards an image i is predicted as:

$$r_{ui} = \boldsymbol{p}_u^T \boldsymbol{q}_i + b_u + b_i + \alpha \tag{6.1}$$

where $\boldsymbol{p}_u$ and $\boldsymbol{q}_i$ are the K-dimensional latent factors of user preference and image characteristics, respectively. The inner product $\boldsymbol{p}_u^T \boldsymbol{q}_i$ of the user latent vector and image latent vector represents a user's preference towards an image; it measures how well the user preferences align with the properties of the image. b_u and b_i correspond to user and image bias term while α is a global offset.

6.2.2 Bayesian Personalized Ranking

Since users only provide sparse one-class positive feedback (the "likes"), there is ambiguity in the interpretation of non-positive images since the negative examples and unlabeled positive examples are mixed together [77]. In this implicit feedback scenario, we may only assume users prefer the liked images to those that are not acted upon. To estimate the latent factors, instead of trying to model the matrix of "likes" directly in a pointwise setting with a least square regression formulation, we can construct the learning objective based on pairwise ranking between images. This idea is key to Bayesian Personalized Ranking [79], such that observed likes should be ranked higher than the unobserved ones. The model then tries to find latent factors that can be used to predict the expected preference of a user for an item.

Formally, we can adapt BPR to the *personalized image recommendation* task as follows. Suppose we have a user u_h and a pair of images i_j and i_k. User u_h's feedback for i_j is positive, and feedback for i_k is unobserved: we denote this relation as $j >_h k$. BPR aims to maximize the posterior probability $p(\Theta | j >_h k)$, where Θ is the set of parameters we try to estimate. According to Bayes' rule:

$$p(\Theta | j >_h k) \propto p(j >_h k | \Theta) P(\Theta) \tag{6.2}$$

and the likelihood function is defined as:

$$p(j >_h k|\Theta) = \delta(r_{hj} - r_{hk}) \tag{6.3}$$

where $\delta(\cdot)$ is the sigmoid function. To simplify notation, We will use the index of a user and an image. We assume a Gaussian prior $\Theta \sim N(0, \lambda_\Theta I)$, where λ_Θ is a set of model-specific parameters and I is the identity matrix. The prior provides regularization for the parameters to prevent overfitting.

Our objective is to find Θ that maximizes the log-likelihood for all users and all images:

$$\arg\max_{\Theta} \sum_{u_h \in \mathcal{U}, i_j (\in \mathcal{P}_h, i_k \in \mathcal{N}_h} \left(\ln \left(\delta(r_{hj} - r_{hk}) \right) - \lambda_\Theta \|\Theta_{hjk}\|^2 \right) \tag{6.4}$$

where $\mathcal{P}_h$ is the set of images for which u_h has provided positive feedback, and $\mathcal{N}_h$ is the set of images for which u_h's feedback is unobserved. Θ is $\{p_u, q_i, b_i\}$ for all users and images. With this pairwise setting, the user bias and global offset in equation 6.1 are canceled out.

6.3 Neural Personalized Ranking

In this section, we seek to complement existing matrix factorization and BPR-based approaches to personalized image recommendation through the exploration of a new neural network based personalized pairwise ranking model. Neural recommendation models promise some exciting characteristics in comparison with BPR: (i) First, user preferences in BPR are calculated as the inner product of user latent vector and image latent vector, which assigns equal weight to each dimension of the latent feature space. In contrast, neural methods may be able to capture the variability of user preferences by relaxing this equal weight requirement. (ii) Second, the matrix factorization component of BPR is linear in nature, which has limited expressiveness. In contrast, neural methods offer more flexibility by adding nonlinearity through activations. (iii) Finally, many neural methods

may be easily parallelized for scalable computation, whereas existing work on distributed BPR typically uses partially shared memory which may limit its scalability. In summary, neural models promise potentially more flexibility in model design, added nonlinearity through activations, and ease of parallelization.

6.3.1 Model Architecture

The NPR model structure is shown in Figure 6.1. There are three inputs to the model, the user and a pair of images, represented as tuple of index (h, j, k). Then user and image indexes are one-hot encoded as tuple of vectors $(\boldsymbol{u}_h, \boldsymbol{i}_j, \boldsymbol{i}_k)$. Since there are M users and N images, the dimensions of $\boldsymbol{u}_h, \boldsymbol{i}_j, \boldsymbol{i}_k$ are M, N, and N respectively. The output of the proposed model is the ground truth value which we train the model against:

$$y(h, j, k) = \begin{cases} 1 & \text{for } j >_h k \\ -1 & \text{for } j <_h k \end{cases} \tag{6.5}$$

where $j >_h k$ denotes that user u_h prefers image i_j to i_k. This definition transforms the ranking problem into a binary classification problem, which aims to check whether the pairwise preference relation holds. Following the input layer, each input is fully connected to the corresponding embedding layer for the sake of learning a compact representation of the users and images. The embedding dimension for both users and images are the same. We denote the embeddings as $\boldsymbol{p}_h$, $\boldsymbol{q}_j$, $\boldsymbol{q}_k$. Formally,

$$\boldsymbol{p}_h = \boldsymbol{W}_u \boldsymbol{u}_h, \quad \boldsymbol{q}_j = \boldsymbol{W}_i \boldsymbol{i}_j, \quad \boldsymbol{q}_k = \boldsymbol{W}_i' \boldsymbol{i}_k. \tag{6.6}$$

where $\boldsymbol{W}_u$,$\boldsymbol{W}_i$, $\boldsymbol{W}_i'$ are embedding matrices for users and images. As the model architecture is vertically symmetric, let's focus on the substructure marked inside the dotted triangle (see Figure 6.1). In the merge layer, user and image embedding vectors are multiplied element-wise, such that each dimension of the user preference vector and corresponding

image properties are in line. This step is analogous to the traditional matrix factorization. The resulting vector has the same dimension as the embeddings. More precisely:

$$\boldsymbol{m}_{hj} = \boldsymbol{p}_h \circ \boldsymbol{q}_j \tag{6.7}$$

where $\circ$ denotes the element-wise product. The merge layer is then connected to a single neuron dense layer, which computes the weighted sum of all dimensions and passes it through a ReLU nonlinear activation. Compared to traditional matrix factorization, such design allows each latent dimension to vary in importance and supports additional expressiveness through non-linearity. We adopt ReLU here based on our exploratory experiments, where we find that alternative activation functions like *sigmoid* and *tanh* suffer from saturation, which leads to overfitting. The output is preference score r_{hj}:

$$r_{hj} = a(\boldsymbol{w}^T \boldsymbol{m}_{hj} + b_1) \tag{6.8}$$

where $a(\cdot)$ is the activation function, $\boldsymbol{w}$ is the weight vector and b_1 is the bias term. This output r_{hj} characterizes the preference of u_h to i_j. We denote preference score from the mirror structure in Figure6.1 as r'_{hk}. Ultimately, the model prediction is $r_{hj} - r'_{hk}$.

6.3.2 Objective Function

We define the objective function to maximize as:

$$\frac{1}{n} \sum_{h \in \mathcal{U}, (i_j \in \mathcal{P}_h, i_k \in \mathcal{N}_h | i_j \in \mathcal{N}_h, i_k \in \mathcal{P}_h)} \ln \Big(\delta \big((r_{hj} - r'_{hk}) \cdot y(h,j,k) \big) \Big) - \lambda_\Theta \|\Theta\|^2 \tag{6.9}$$

where n is the number of training samples, $\delta(\cdot)$ is the sigmoid function, and $y(h,j,k)$ is the ground truth value. Since we only focus on whether the sign of the output is the same

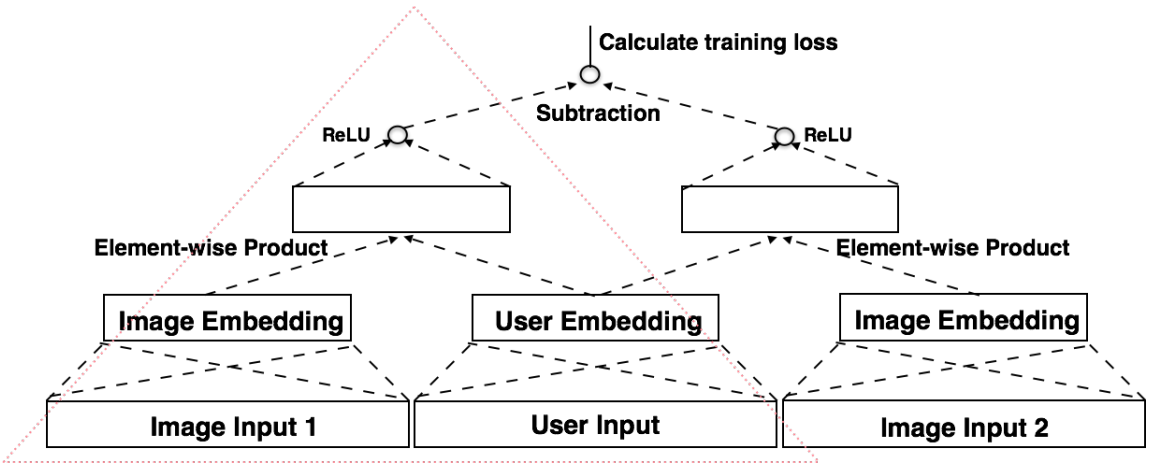


Figure 6.1: Neural Personalized Ranking (NPR) Structure

as $y(h, j, k)$, we employ the product between predicted value $r_{hj} - r'_{hk}$ and ground truth $p(h, j, k)$ as an indicator for how the predicted value is aligned with ground truth. A larger value is acquired if their signs are same. The regularization term is slightly different from that defined in the BPR-based model. We impose the L2-norm to the whole embedding matrix, instead of on each training sample for simpler implementation. If training samples are balanced for each user and image, such regularization will have the same effect as that mentioned in the BPR model.

6.3.3 Model Training and Inference

We initialize the weight matrices with random values uniformly distributed in [0,1]. To train the network, we transform the objective to equivalent minimization dual problem and adopt mini-batch gradient descent (MB-GD), which is a compromise between gradient descent (GD) and stochastic gradient descent (SGD). MB-GD converges faster than GD as it has frequent gradient updates while convergence is more stable than SGD. Besides, MB-GD allows utilization of vectorized operations from deep learning libraries, which typically results in a computational performance gain over SGD. Before each epoch, we

shuffle the training dataset. Then in each step, a batch of training tuples is served to the network. The error gradient is back propagated from output to input and parameters in each layer are updated. The batch size we used in experiments is 1,024. The optimization algorithm used for gradient update is Adam's[157]. The loss generally converges within 20 epochs given the amount of training data.

Given a user u, for every image $i \in \mathcal{N}_u$, her preference score r_{ui} is predicted from the neural network. In order to obtain the preference score, we feed the tuple (u, i, i) to the neural network, and get two values r_{ui} and r'_{ui} from the parallel branches. The final preference score is calculated as $r_{ui} = \frac{1}{2}(r_{ui}+r'_{ui})$. Then the set of images with unobserved feedback are sorted according to descending predicted preference score. We pick the top ranking images for recommendation.

6.3.4 Implementation Details

Neural network models can easily overfit. Thus we take a few measures to prevent overfitting. First, we apply dropout to the embedding weights during training. The dropout rate is fine-tuned for each dataset. Second, early stopping is adopted to terminate training if there is no decrease on validation loss for 3 epochs. Additionally, we impose L2-regularization to the user contextual preference vectors, such that the preference score is not overwhelmed by large contextual feature values. If validation loss does not decrease, we reduce the learning rate to 20% of its current value, allowing for finer adjustment to gradient update. Furthermore, we tune the regularization coefficients through grid search to optimal.

6.4 Contextual NPR

Although the neural personalized ranking model is promising, it faces two key challenges. The first is *sparsity* – very few images have been liked, so it is difficult to make recommendations for users who have little feedback as well as to recommend newly posted

images. The second is *preference complexity* – images are diverse and there are many reasons for a user to like an image. Hence, we propose to improve NPR with an enhanced model – *contextual neural personalized ranking (C-NPR)* – by leveraging multiple categories of auxiliary information that may help overcome the sparsity issue while also providing clues to user preferences.

In this section, we begin by showing evidence about user's preference reflected through contextual domains, then we formally define the contextual NPR model and explain how this model is capable of modeling multiple categories of contextual preference jointly. Finally, we shed light on how we derive each category of image contextual feature vector.

6.4.1 Geo, Topical, Visual Preference

Based on the Flickr YFCC100M dataset [158] (see Section 6.7.1), we begin here by highlighting evidence for the impact of three sources of contextual information on image preference, before formally defining the contextual NPR model.

Evidence of Spatial Preference. Figure 6.2 shows the percentage distribution of "liked" image in decreasing order across the regions where these images were taken. Here we aggregate each user's top-10 regions where their liked image comes from. The k^{th} boxplot is generated from all users who have liked images from at least k regions. We observe that the median percentage of liked images from the top region is above 33%; that is, at least half of all users have 33% of their liked images come from a single region (though not necessarily the same region for each user). Suppose a user has no preference of regions, a single region would at most contain 9% of her liked images (as the largest region contains 9% of the images). Thus we conclude there is a strong tendency for a user to favor images from certain regions, especially from a few of them as the percentage decrease sharply as the region number increases.

Evidence of Topic Preference. We consider each unique user tag as a potential topic.

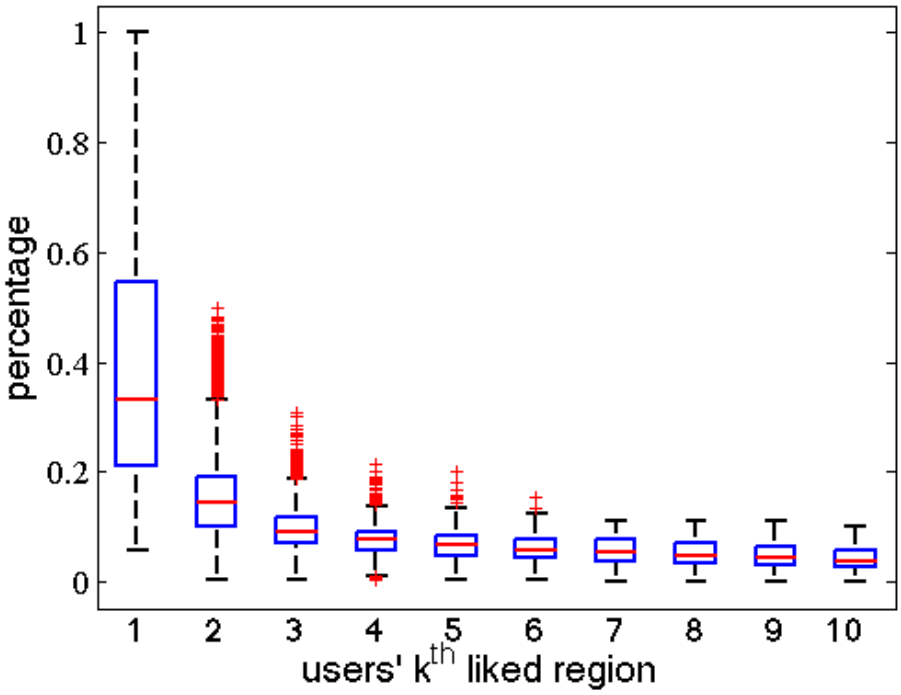


Figure 6.2: Geo preferences: Users tend to "like" images from only a few regions.

Figure 6.3 shows users' liked image distribution over the tags that have been applied to those images. We list the results for top-10 tags of each user (not necessarily the same set of tags for each user). The k^{th} boxplot summarizes users that have more than k tags labeled to the set of liked images. We observe that $\sim$75% of users have at least one common tag shared among more than $\sim$35% of their liked images. Even the median ratio for the 10^{th} tag attached to liked images is much higher than the percentage of most frequent tags in the whole dataset. Thus we conclude that users have topic preferences for the images they like. As the percentage decreases slowly with k, we ascribe this to user may have multiple favorable tags.

Evidence of Visual Preference. Finally, we explore clues for user's visual preference by comparing image similarity across three sampled sets, with each containing 100,000

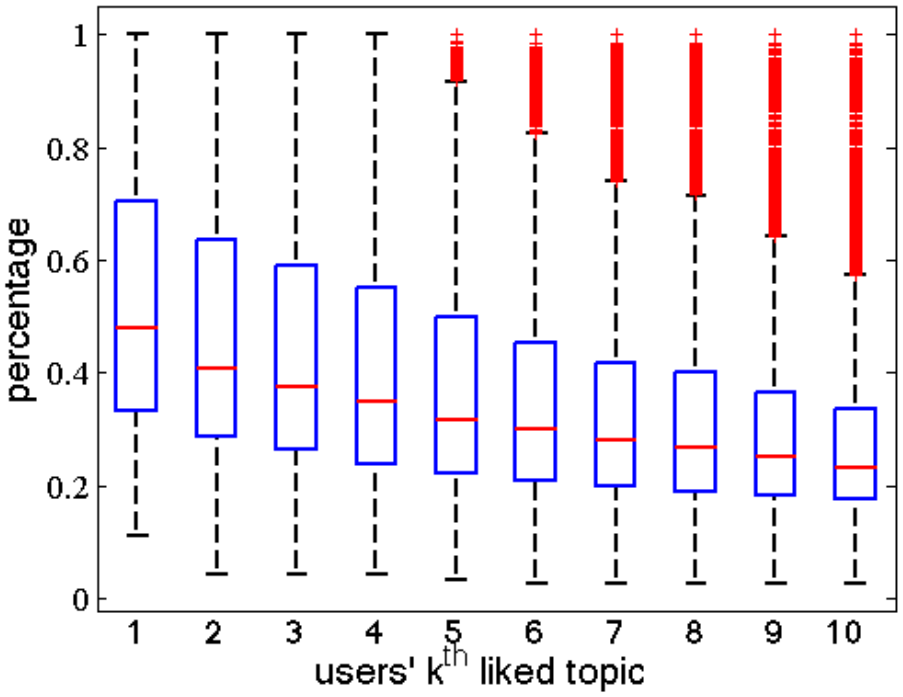


Figure 6.3: Topic preferences: Users tend to "like" images with similar tags.

image pairs. The sets are constructed in the following manner: (i) Randomly sample image pairs; (ii) Randomly sample a user, then sample a pair of image from her liked images; and (iii) For each image, pick its most socially alike images[2]. Next, we calculate the cosine similarity of aforementioned image pairs based on their visual feature vectors. The similarity distribution for three sets are shown in Figure 6.4. We observe that image pairs liked by a user tend to be more similar in visual appearance than randomly picked image pairs, with a median similarity around 0.25 vs.0.20. For image pairs that are liked by similar groups of users, the pairwise visual similarity is even higher, reaching 0.30. All three findings are statistically significant with p-value less than 1e-8. Hence, we conclude

[2]We represent each image as a vector of user's who like it, then identify similar images with high cosine similarity score.

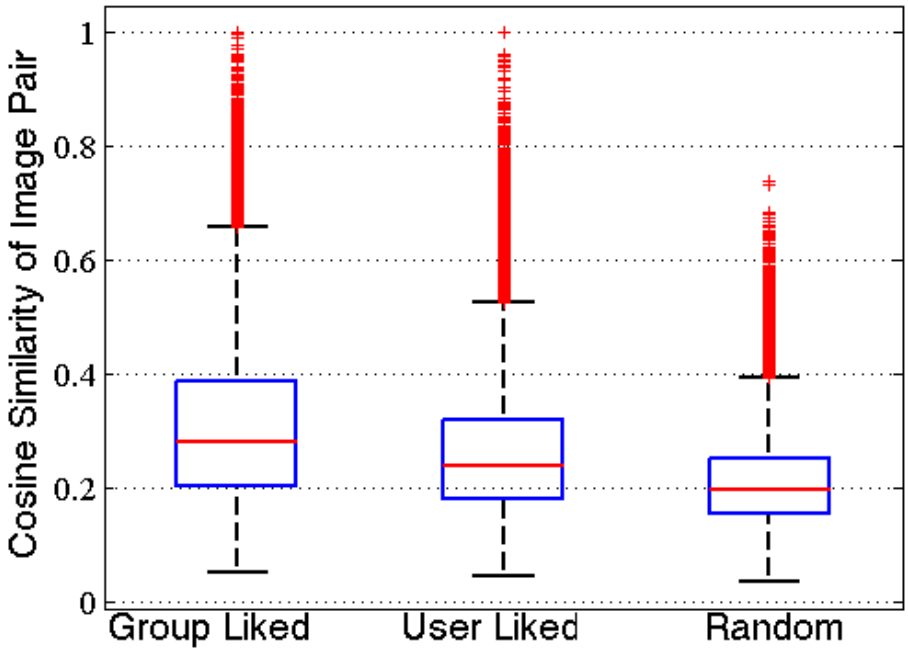


Figure 6.4: Visual preferences: Pairs of "liked" images tend to be more visually alike than random pairs.

that users have visual preference for images that they like, and that there exists group of users that share similar preferences.

Evidence of Social Preference. We compare the preference similarity among socially close user samples and random user samples, with each containing 100,000 user pairs. The samples are constructed as follows:

1. Randomly sample user pairs.

2. For each user, pick top users that are socially close by considering both user contacts and group information.

Figure 6.5 shows the preference similarity distribution for each set of user pairs. Again

we adopt cosine similarity. We observe most user pairs in these two sets have zero similarity, due to the sparse nature of the dataset. However, we notice the socially close user pairs are denser for similarity greater than 0, with the 75 percentile equals to 0.03. We further verify the distribution difference through two sampled t-test, which rejects the hypotheses that two samples are drawn from the same distribution. Thus we conclude socially close users are prone to have similar preference than random users.

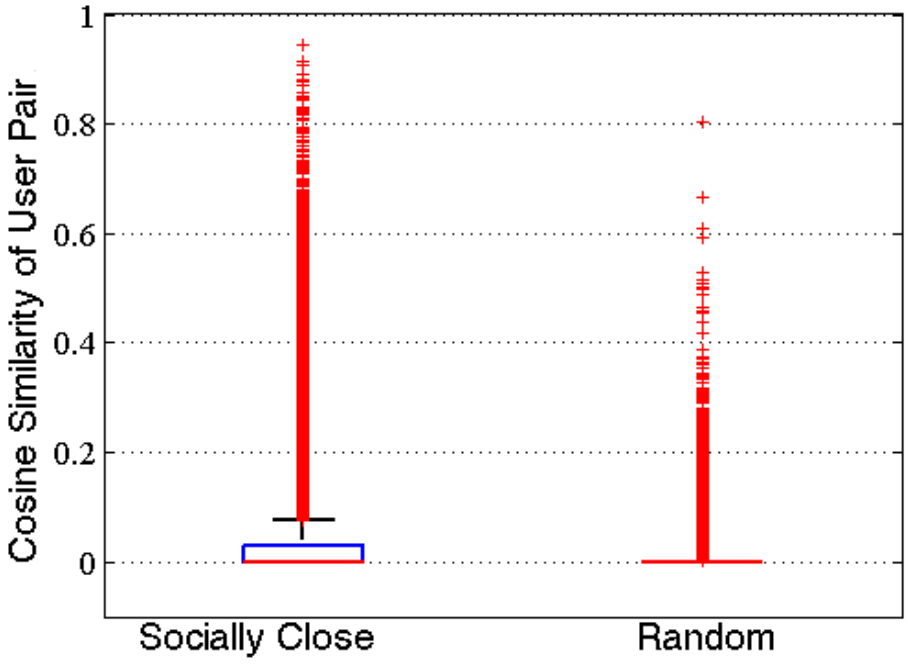


Figure 6.5: Compare Socially Close and Random Users

6.4.2 Modeling Geo, Topical, and Visual and Social

Given the evidence of user preferences, we turn here to model these features for integration into the basic NPR model.

Deriving Spatial Feature. We noticed that each users' preference for an image is not evenly distributed across regions. A user may prefer images from certain locations or multiple locations. Thus it might be worthy to take into account geo-preferences in recommendation because it not only help to understand how to recommend in the presence of geo-information but helps to explain why modeling spatial clustering may cope with the sparsity issue and boost overall performance. To this end, we propose a new model for jointly modeling geographical preference in image recommendation.

We assume the area is geographically partitioned into K regions and each image is taken from one of the regions. Instead of griding into blocks of equal area which has been used previously [97, 159], we propose to partition area into regions according to the image density, where the shape and size of region doesn't have to be consistent and could be irregular. The reason is images are not distributed homogeneously across the continent (generally, dense around cities and tourist attractions and sparse elsewhere). Focusing on density helps to reduce the irrelevant areas and the size of each region we drill down into, which allows for more precise modeling. We apply mean shift clustering algorithm, which builds upon the concept of kernel density estimation (KDE), to identify geographical clusters of images. It works by placing a Gaussian kernel on each image coordinate in the dataset. Then by iteratively shifting each point in the data set until they reach the top of their nearest KDE surface peak. The only parameter to set is the bandwidth, with which it attempts to generate a reasonable number of clusters based on the density. The cluster result is shown in Figure 6.7, where each dot represents an image and the cluster of points represents a region. In total, there are 217 regions with bandwidth of 100km.

We assume the probability that a user likes an image in one region is influenced by her likes status in other regions. If a user has liked images from region p, then she has a larger probability of favoring an image in a region closer to p. Previous work in POI recommendation assumes the influence distance of a POI is fixed according to a normal distribution

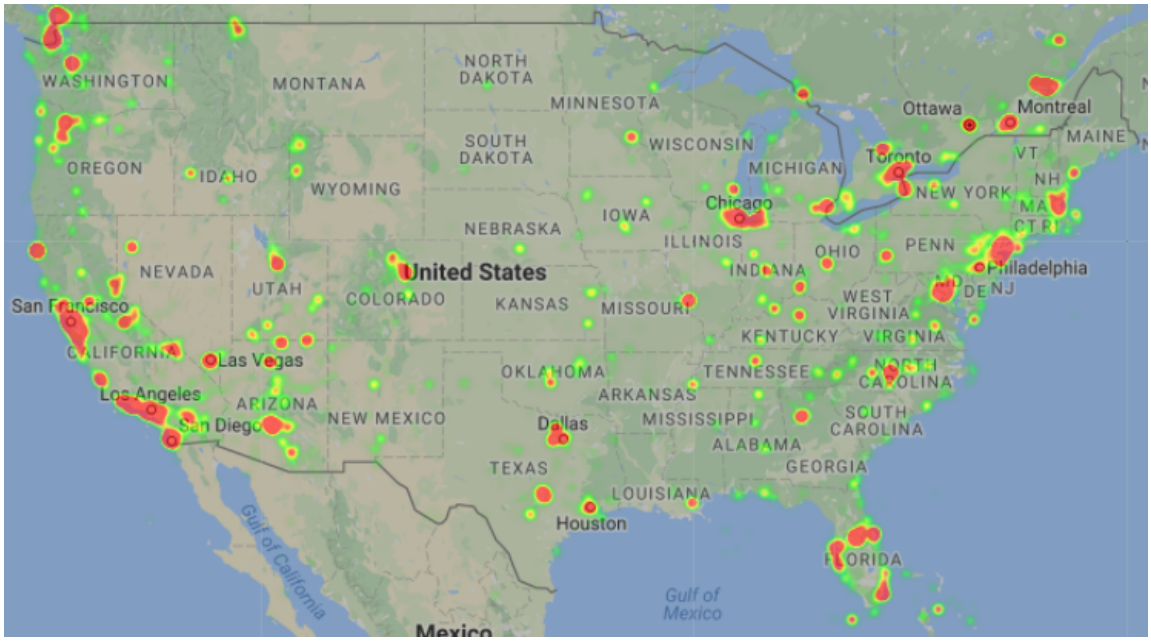


Figure 6.6: Image Heatmap

$\mathcal{N}(0, \sigma^2)$ [97]. However, it is commonly perceived that influence for regions of the same size should be different, not to mention the diverse shape and size in our scenario. Thus we assume each region p has an influence according to a normal distribution $\mathcal{N}(0, \sigma_p^2)$, where σ_p is the standard deviation of distance from each image coordinate to the cluster center. To this end, the influence from p_i to p_j is defined as:

$$f_{ij} = \frac{1}{\sigma_{p_i}} K(\frac{d(i,j)}{\sigma_{p_i}})$$

, where p_i and p_j are the regions that image i and j belong to and the relation between image and region is many to one. $d(\cdot)$ is the Haversine distance between the center of two regions, $K(\cdot)$ is the standard normal distribution and σ_{p_i} is the standard deviation which we adopt as the bandwidth of kernel function. Thus the influence from each region to all other regions is represented as a row vector. The advantage is it encodes the idea

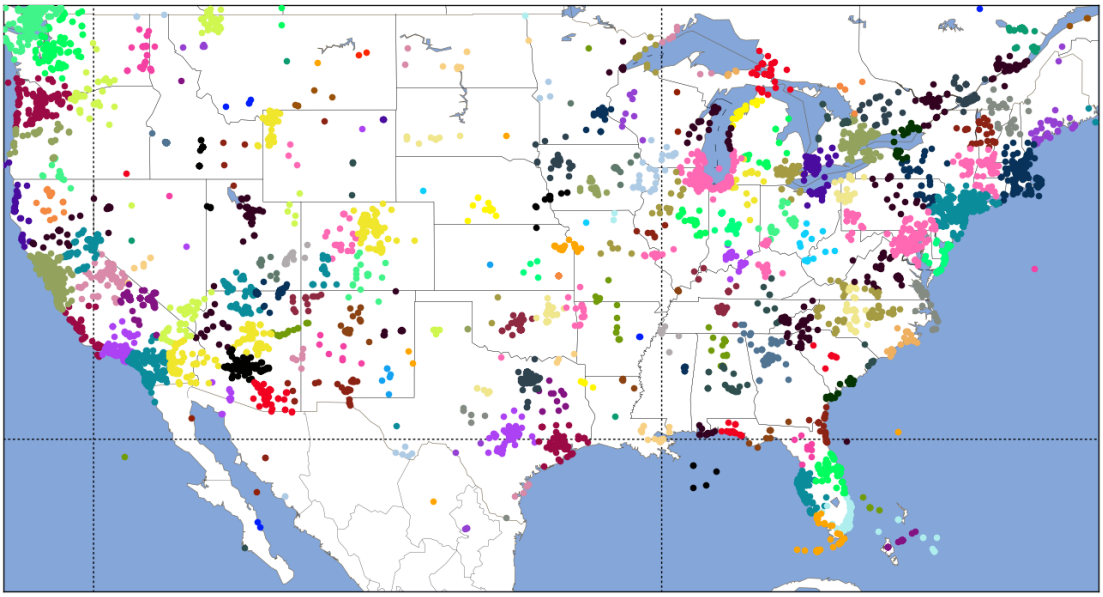


Figure 6.7: Geographic Regions by Meanshift Clustering

of kernel density estimation where the estimated geographical density of u's liked image distribution at p_j is:

$$d_u^j = \sum\nolimits_{p_i \in P_u} \frac{n_i}{\sigma_{p_i} |P_u|} K(\frac{d(i,j)}{\sigma_{p_i}})$$

, where n_i is u's number of likes within p_i, P_u is the set of regions that u has likes. It can be written as the dot product of two vectors. However, different from the kernel density estimation, a user's preference vector is learned.

Deriving Topic Features. Text based search is mainstream nowadays and is indispensable channel for user to discover image. Tags not only act as summarization of concrete things like recognizable objects, scene, and weather, but also shed light on abstract and hidden knowledge about the image like emotion and background theme, which supplements the visual appearance. Each user has her own flavor and interest. Thus it is important that the topic preference of the user is aligned with the topic of the image when making recommendation. Thus it is meaningful to research on how to incorporate social tags, as an

indicator of topical interest, into image recommendation, because it helps understanding how to boost recommendation and cope with sparsity issue with the presence of topic information. each image has a topic feature vector $\boldsymbol{t}_i$, representing topic properties of the image.

To extract the topical theme associated with each image, we aggregate the user-generated tags, title, and description (if any) for each image. This text not only acts as a descriptor of concrete objects, scene, and weather, but also sheds light on abstract and hidden knowledge about the images like emotion and background theme, which supplements the visual appearance. We ignore tags which have occurred fewer than d times in the dataset and apply dimensionality reduction over 58k unique tags. PCA and topic modeling method Latent Dirichlet Allocation (LDA) are compared for carrying out this task. The detailed comparison is shown in the experiments section.

Deriving Visual Features. Recently, high-level visual features extracted from deep convolutional neural networks have revolutionized the state-of-the-art performance in image recognition [160] and image captioning [161]. Here, the output of fc6 layer of the Places Hybrid-CNN is adopted as the image feature [162], which contains 4,096 dimensions. This CNN was trained on 1,183 categories which includes 205 scene categories from Places Databases and 978 object categories from ImageNet (ILSVRC2012) images. Dimension reduction is further applied for reducing computation complexity. The existing approach for visual BPR [94], which learns an embedding kernel for visual dimension reduction while training the recommendation model, turns out to be less efficient than directly utilizing the full set of 4,096 features. Hence, we propose to reduce visual feature dimension separately from model training. We compare the performance of using principle component analysis (PCA) and stacked auto-encoder for image feature dimension reduction with harnessing the whole set of features in training the model. For PCA, we retain the first v_d components in decreasing order of explained variance as new visual features. The

stacked auto-encoder has 3-layer encoding network and 3-layer decoding network which the structure is mirror of each other. The AE is trained in layer-wise fashion.

Figure 6.8 shows the structure of stacked auto-encoder we use, the v_d dimensional output of the encoder network is treated as new visual features. To train the stacked auto-encoder, we adopt layer-wise training strategy. First, we train an auto-encoder with one hidden layer, by applying original visual feature as both input and output. Next, We feed the images to this encoder network and use the output of hidden layer (or we say encoder network) as input and output for training the second auto-encoder, so on and so forth. Finally, we can acquire the feature vector.

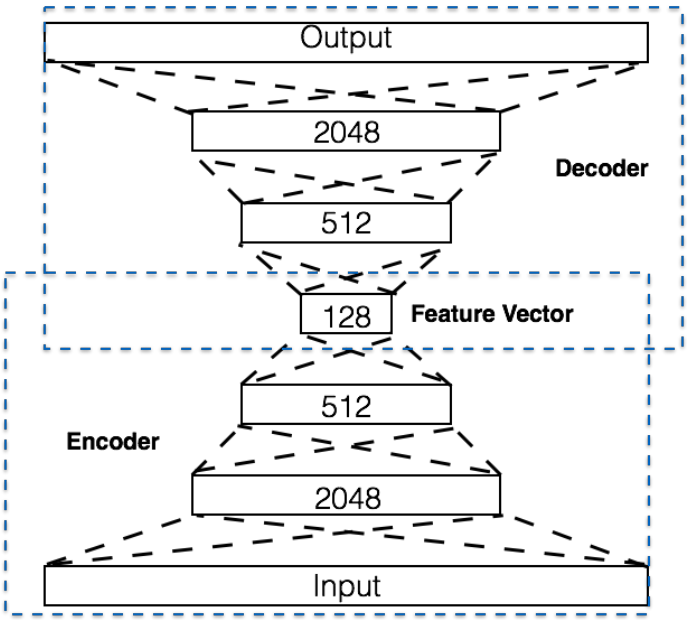


Figure 6.8: Auto-encoder

Modeling Social Influence. Social closeness has been proven to contribute to the effectiveness of various prediction and recommendation tasks [89, 90]. Apart from modeling visual, topic and geo preference, we are also interested to explore how we will be informed

by preference of socially similar user. We crawled each user's contact and groups through Flickr API. We consider group and contact as two forms of social connection. Additionally, we assume images that are liked by same group of users to be more similar than otherwise.

User-user relation derived from group. When two user have similar membership of the groups, we assume their preference for images are more alike than those who are not. Similarly, if two users have similar contact friends, we consider their preferences to be more similar than those who are not. Since groups and contacts are all unique for each user, we adopt Jaccard similarity to measure the user closeness where users are represented as a vector of groups that she joined or the contacts she has, respectively. And the user similarity is defined as the sum of similarity based on group sim_g and similarity based on contact sim_c:

$$sim(u_i, u_j) = sim_g(u_i, u_j) + sim_c(u_i, u_j)$$

We incorporate these relations as a regularization term to objective function:

$$\sum_{i=1}^{N} \sum_{l \in U^i_{social}} sim(u_i, u_l) \|\boldsymbol{p}_i - \boldsymbol{p}_l\|^2 \tag{6.10}$$

$\boldsymbol{p}_i$, $\boldsymbol{p}_l$ are the latent user factor, and U^i_{social} is the set of her similar users. When two users are socially similar, the regularization forces the difference between the latent factor of two users to be closer. We use the 25 closest neighbors for computational efficiency.

Image-image relationship. When two images are favored by the same group of users, we assume these images are more similar than those that are not. With this assumption, we

have the following regularization term to objective function:

$$\sum_{j=1}^{N} \sum_{l \in I_{sim}^{j}} sim(i_j, i_l) \|\boldsymbol{q}_j - \boldsymbol{q}_l\|^2 \tag{6.11}$$

where we use Jaccard similarity for image similarity $sim(\cdot)$. When $sim(i_j, i_l)$ is large, the difference between latent image factor $\boldsymbol{q}_j$ and $\boldsymbol{q}_l$ is forced to be small. We further denote the sum of equation 6.10 and Equation 6.11 as S.

6.5 From NPR to C-NPR

This evidence of clear variation in user preference motivates our need to augment NPR. Formally, with the contextual feature vector $\bar{\boldsymbol{f}}_{\boldsymbol{i}}$ for image i, we then seek to uncover user's preference latent vector $\boldsymbol{f}_{\boldsymbol{u}}$ to $\bar{\boldsymbol{f}}_{\boldsymbol{i}}$ such that the vector product $\boldsymbol{f}_{\boldsymbol{u}} \circ \bar{\boldsymbol{f}}_{\boldsymbol{i}}$ captures how user preference is aligned with the image's contextual features. Concretely, we explore the impact of three features – geo-location, topics, and visual features – on these preferences. We modify the neural network structure of each branch in Figure 6.1 to accommodate for

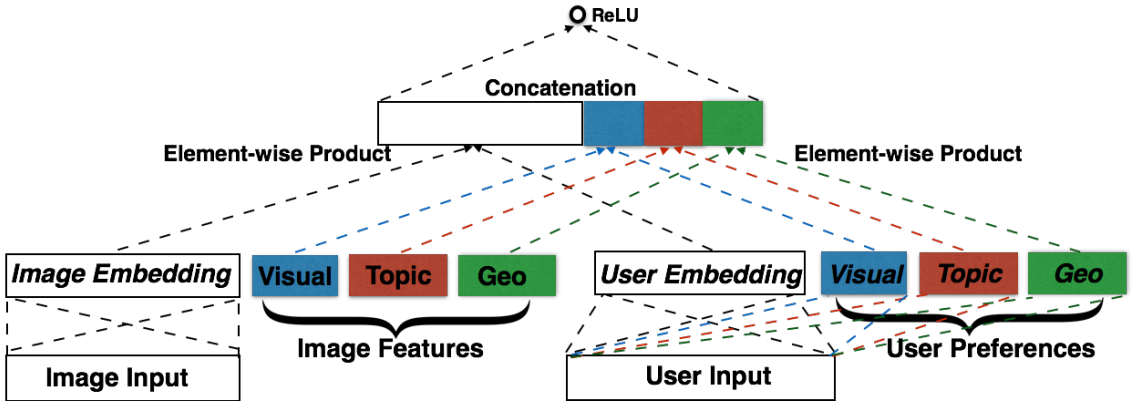


Figure 6.9: NPR with Contextual Information

modeling contextual preference. The new architecture for a branch incorporating visual,

geo, and topic contextual features and preferences is shown in Figure 6.9. Aside from the user and image input, each category of contextual features of image $\bar{\boldsymbol{v}}_i$, $\bar{\boldsymbol{t}}_i$, $\bar{\boldsymbol{g}}_i$ is served as an extra input. Each corresponding contextual preference hidden layer is fully connected above user input and to be learned. Then the user's preference to the contextual feature is calculated with element-wise product to measure how feature and preference are aligned. Specifically, visual topic, geo latent vector of user u_h are calculated as

$$\boldsymbol{v}_h = \boldsymbol{W}_v \boldsymbol{u}_h, \quad \boldsymbol{t}_h = \boldsymbol{W}_t \boldsymbol{u}_h, \quad \boldsymbol{g}_h = \boldsymbol{W}_g \boldsymbol{u}_h \tag{6.12}$$

where $\boldsymbol{W}_v$, $\boldsymbol{W}_t$, $\boldsymbol{W}_g$ are the weight matrices. The visual, topic, geo preference of user u_h to image i_j are

$$\begin{aligned} \boldsymbol{e}_{hj}^v &= \boldsymbol{v}_h \circ \bar{\boldsymbol{v}}_j \\ \boldsymbol{e}_{hj}^t &= \boldsymbol{t}_h \circ \bar{\boldsymbol{t}}_j \\ \boldsymbol{e}_{hj}^g &= \boldsymbol{g}_h \circ \bar{\boldsymbol{g}}_j \end{aligned} \tag{6.13}$$

Then the general preference $\boldsymbol{m}_{hj}$ and contextual preferences are concatenated in the merge layer. Formally:

$$\boldsymbol{m'}_{hj} = \begin{bmatrix} \boldsymbol{m}_{hj} & \boldsymbol{e}_{hj}^v & \boldsymbol{e}_{hj}^t & \boldsymbol{e}_{hj}^g \end{bmatrix}^T \tag{6.14}$$

Finally, the merge layer is further connect to a single neuron dense layer as before. The updated preference score is

$$r_{hj} = a(\boldsymbol{W'}\boldsymbol{m'}_{hj} + b_1) \tag{6.15}$$

Additional contextual information about image can be incorporated following the same step as stated above. In summary, each new feature vector is served as an extra input to the neural network, and a corresponding preference embedding layer is augmented on top of user input. Then element-wise product is adopted to model consistency between

preference and intrinsic property, followed by concatenation of all preference components and weighted sum.

6.6 From BPR to C-BPR

In addition to updating NPR with contextual information, we can also integrate these new factors into BPR. Indeed, a visual preference-enhanced version of BPR has been previously introduced by He et al.[94], where the basic matrix factorization model in equation 6.1 is extended to a visual-enhanced model:

$$r_{ui} = \boldsymbol{p}_u^T \boldsymbol{q}_i + \boldsymbol{v}_u^T \bar{\boldsymbol{v}}_i + b_i \tag{6.16}$$

where $\boldsymbol{v}_u$ is user's visual preference latent vector, and $\bar{\boldsymbol{v}}_i$ is the visual feature of the image. The inner product $\boldsymbol{v}_u^T \bar{\boldsymbol{v}}_i$ indicates how user's visual preference align with the visual appearance of the image. If the two components align well, the resulting rating score would be higher, and vice versa.

Similarly, we can define a topic-enhanced model as:

$$r_{ui} = \boldsymbol{p}_u^T \boldsymbol{q}_i + \boldsymbol{t}_u^T \bar{\boldsymbol{t}}_i + b_i \tag{6.17}$$

where we assume each user has topic preference latent vector $\boldsymbol{t}_u$ where each dimension represents the preference to some topic keywords, and each image has a topic feature vector $\bar{\boldsymbol{t}}_i$, representing topic properties of the image. The inner product $\boldsymbol{t}_u^T \bar{\boldsymbol{t}}_i$ captures the coherence between topic interest of user and topic theme of the image. The topic enhanced model put emphasize on topic matching.

We can define a geo-enhanced model as:

$$r_{ui} = \boldsymbol{p}_u^T \boldsymbol{q}_i + \boldsymbol{g}_u^T \bar{\boldsymbol{g}}_i + b_i \tag{6.18}$$

where $\boldsymbol{g}_u$ is a user's latent geographic preference vector with each element corresponding to u's preference for a region, and $\bar{\boldsymbol{g}}_i$ is the geographic characteristic vector of p_i, encoding the influence from other locations to p_i. To note, all images from the same region as i share the same $\bar{\boldsymbol{g}}_i$. The geo-enhanced model use the inner product $\boldsymbol{g}_u^T \bar{\boldsymbol{g}}_i$ to account for the user's preference for image belonging to a region. In this way we can recommend more images from unpopular regions which alleviates sparsity.

In total, we additionally integrate topic and geo factors describing user u's preference for image i as:

$$r_{ui} = \underbrace{\boldsymbol{p}_u^T \boldsymbol{q}_i}_{\text{general}} + \underbrace{\boldsymbol{v}_u^T \bar{\boldsymbol{v}}_i}_{\text{visual}} + \underbrace{\boldsymbol{t}_u^T \bar{\boldsymbol{t}}_i}_{\text{topical}} + \underbrace{\boldsymbol{g}_u^T \bar{\boldsymbol{g}}_i}_{\text{geo}} + \underbrace{b_i}_{\text{bias}} \tag{6.19}$$

One reason for such explicit modeling of the three side information is there exists no evidence to show that they are included in the basic latent space. It guarantees more diverse recommendation by incorporating visually, geographically, and topically related images even though they are potentially appealing, thus providing an offset to the effect of sparsity.

6.6.1 Learning the Model

The objective function L to maximize is:

$$\sum_{u_h \in \mathcal{U}, i_j \in \mathcal{P}_h, i_k \in \mathcal{N}_h} \Big(\ln \big(\delta(r_{hj} - r_{hk}) \big) - \lambda_\Theta \|\Theta_{hjk}\|^2 - \lambda_s S_{hjk} \Big) \tag{6.20}$$

The set of parameter Θ to be learned is $\{\boldsymbol{p}_u, \boldsymbol{q}_i, \boldsymbol{v}_u, \boldsymbol{t}_u, \boldsymbol{g}_u, b_i\}$ for all users and images. We apply stochastic gradient ascent for maximizing the objective. In each iteration, for every tuple (u_h, i_j, i_k) in training dataset, we update each parameter by taking a step along the ascending gradient direction in the following fashion:

$$\Theta^{t+1} = \Theta^t + \eta \frac{\partial L_{hjk}}{\partial \Theta}$$

where L_{hjk} is the component in L corresponding to (u_h, i_j, i_k). In more details, we list partial derivative for each parameters:

$$\text{as } \frac{\partial \ln(\delta(x))}{\partial x} = \frac{1}{1+e^x}, \quad \text{let } \varphi = \frac{1}{1+e^{r_{hj}-r_{hk}}}$$

Update for user latent factor $\boldsymbol{p}_h$

$$\frac{\partial L_{hjk}}{\partial \boldsymbol{p}_h} = \varphi(\boldsymbol{q}_j - \boldsymbol{q}_k) + \lambda_u \boldsymbol{p}_h + \lambda_s \sum_{l \in U^h_{social}} sim(u_h - u_l)(\boldsymbol{p}_h - \boldsymbol{p}_l)$$

Update for image latent factors $\boldsymbol{q}_j$ and $\boldsymbol{q}_k$

$$\frac{\partial L_{hjk}}{\partial \boldsymbol{q}_j} = \varphi \boldsymbol{p}_h + \lambda_i \boldsymbol{q}_j + \lambda_s \sum_{l \in I^j_{sim}} sim(i_j - i_l)(\boldsymbol{q}_j - \boldsymbol{q}_l)$$

q_k is updated similarly.

Update for visual, topic and geo preference latent factors $\boldsymbol{v}_u$, $\boldsymbol{t}_u$, $\boldsymbol{g}_u$

$$\frac{\partial L_{hjk}}{\partial \boldsymbol{v}_j} = \varphi(\boldsymbol{v}_j - \boldsymbol{v}_k) + \lambda_v \boldsymbol{v}_j$$

$\boldsymbol{t}_u$ and $\boldsymbol{g}_u$ are updated similarly, with regularization coefficient λ_t and λ_g.

Update for image bias term b_j and b_k

$$\frac{\partial L_{hjk}}{\partial b_j} = \varphi - \lambda_b b_j$$

b_k is updated similarity.

6.7 Experiments

In this section, we conduct a set of experiments to evaluate neural personalized image recommendation. Specifically, we first introduce the data preparation workflow and basic experimental setup. Then we compare NPR with baseline models, followed by reporting performance of contextual enhanced models. We drill down to discover the impact of each category of contextual information. We further look into the performance of the proposed model in the typical cold start scenario. Finally, we discuss the characteristics of NPR and BPR in terms of amount of training data required and convergence rate.

6.7.1 Data

The dataset we use for evaluation is based on the Flickr YFCC100M dataset [158]. We select images with geo-coordinates and that are located in the US mainland. We then circumscribe the image with machine tags outdoor, nature, landscape, art. We further crawl the image "likes" from the Flickr API and we select images with greater than 30 likes overall and users with more than 10 liked images. We also crawled social information includes groups and contacts for each user.

The resulting datasets for experiments are listed in Table 6.1, where the sparsity for the small dataset and large dataset is 0.96% and 0.16%, respectively, which means only 0.96% and 0.16% of the possible user-image relation is available. These two datasets represent two different levels of feedback scarcity. And effective sparsity for training data is half of the reported value after train/test split. The geographical heatmap of the large dataset is shown in Figure 6.6; we notice the majority of images come from populated areas or famous tourist sites, as shown in red.

Dataset	#Users	#Images	#Feedback	Sparsity
Small	1,891	2,013	36,827	0.96%
Large	27,782	21,720	961,506	0.16%

Table 6.1: Post-processed Datasets Statistics

6.7.2 Experimental Setup

All experiments of BPR-based models were performed on a desktop machine with 60GB memory and 8 core Intel i7-4820k3.7GHz. NPR-based models are trained using Nvidia GeForce GTX Titan X GPU with 12 GB memory and 3,072 cores.

Constructing the Training Set. We randomly partition the liked images of each user into 50% for training and validation and 50% for testing. The validation set split ratio is 0.3. The loss on validation set is used for tracking training progress. The training set consists of tuples (h, j, k) where h, j, k correspond to user index, positive image index, and negative image index, respectively. Including every pair of positive and negative combination for each user in training would be costly. Yet practically, evaluation metrics saturate even with a much smaller set of training tuples. Thus we propose to use a sampling method for generating training tuples.

To generate each training tuple, we first randomly select a user u from user set $\mathcal{U}$, then randomly select a positive image i_j from $\mathcal{P}_u$, and finally randomly select k negative image i_k from $\mathcal{N}_u$ to pair with the i_j. We will discuss the influence of k on performance below. We repeat this process until generating the expected number of training data tuples. All reported results are based on training over a set that sampling users to 5 times of the number of positive feedbacks, with 10 negative images to pair for each positive sample.

Although it is very likely that we end up leaving part of the positive samples unused, the model based on this sampling strategy exhibits better overall performance and requires less training data to converge compared with sampling negatives for each positive sample.

The model is trained in a balanced way among every user and not biased towards users that have more likes.

Evaluation Metrics. We adopt precision@k, recall@k and F1-score@k for evaluating personalized ranking. Precision measures the fraction of correctly predicted images among the retrieved images. Recall measures the fraction of relevant images that have been picked over the total relevant images. F1@k is a weighed average of Prec@k and Rec@k. All measures are averaged across all users.

$$Prec@k = \frac{1}{N}\sum_{i=1}^{N}\frac{|GT(u_i) \cap Pred(u_i)@k|}{k}$$

where $GT(u_i)$ is the ground truth liked image for u_i in test data, and $Pred(u_i)@k$ is the top k recommended images for u_i.

$$Rec@k = \frac{1}{N}\sum_{i=1}^{N}\frac{|GT(u_i) \cap Pred(u_i)@k|}{|GT(u_i)|}$$

$$F1@k = \frac{2 \cdot Prec@k \cdot Rec@k}{Prec@k + Rec@k}$$

Baselines.

- NCF. Neural collaborative filtering is a pointwise model that composed of multi-layer perceptron and generalized matrix factorization components [100]. All the configuration adopted is similar according to original paper including 4 hidden layer, 64 hidden unit and pre-training. We sample 5 negative for each positive which shows to be optimal in the paper.

- Multi-layer perceptron based pairwise ranking model. A personalized pairwise ranking model based on multi-layer perceptron was introduced [101]. We adopt a setting with 3

hidden layers, with each layer containing 200,100,100 units, respectively.

- BPR and its variants. We consider the basic pairwise ranking for matrix factorization model shown in Equation (6.1). In addition, we can also integrate the contextual factors into traditional BPR. Indeed, a visual preference-enhanced version of BPR model has been previously introduced by He et al.[94]. Hence, we also consider a visual (VBPR), topic (TBPR), geo (GBPR), and combined version of BPR (C-BPR).

- NPR. This is the neural network based model for personalized pairwise ranking as shown in Figure 6.1.

- NPR-noact. This is the NPR model without nonlinear activation.

- Contextual NPR. This includes NPR considering only visual (VNPR), topic (TNPR), and geo (GNPR) contextual information.

Reproducibility. For all models, the user and image latent factor dimension are set to 100 empirically for a trade-off between performance and computation complexity as well as for fair comparison. The number of visual feature dimensions is 128, the number of topic dimensions is 100 and 500 for small dataset and large dataset respectively, the number of geographic dimensions is the same as the number of geo clusters which is 155 and 217 for small and large dataset respectively.

For the NPR-based approach, we adopt mini-batch gradient descent where the batch size is set to 1,024. The dropout rate for small dataset was set to 0.6 and for large dataset was set to 0.45. The regularization parameters are fine-tuned. For example, on large dataset λ_u=λ_i=$1\mathrm{e}{-7}$, λ_v=λ_g=$1\mathrm{e}{-6}$, and λ_t=$1\mathrm{e}{-5}$. For BPR-based approaches, we initialize the learning rate to 0.02 and decrease it to 97% its current value in each consecutive iteration, which has been shown to be effective to help convergence in fewer iterations [49]. And generally, training converges within 80 iterations. The regularization parameters

are fine-tuned and shared among all BPR baselines, concretely, λ_u=λ_i=λ_b=0.02, λ_v=λ_g= λ_s=0.01 and λ_t=0.1.

6.7.3 NPR vs. Alternatives

We begin by investigating the quality of NPR versus each of the baselines for personalized recommendation without contextual information. We report the average precision@k, recall@k for k at 5, 10, 15 in Figure 6.10 for small dataset and 6.11 for large dataset. We observe that NPR and BPR are neck and neck, with BPR slightly superior (less than 1%) in precision and recall. This indicates BPR-MF is a strong baseline. Although MF component is linear, the logistic objective function brings in nonlinearity. Both approaches consistently substantially outperform other baseline approaches in precision and recall. Moreover, the pairwise method generally yields better results. For example, NPR improves the precision and recall over the pointwise model NCF by 50% for the large dataset and improves the precision and recall. This illustrates the relaxed assumption for unobserved samples helps to reduce the recommendation bias. The nonlinear activation function lead to an average of 3.8% increase in precision and 3.3% increase in recall on small dataset, and even larger 11.5% and 12.5% increase in precision and recall on the large dataset. By bringing in nonlinearity, the representativeness of the model is enriched. We observe the performance metrics are generally lower on the large dataset; the reason is that recommendation becomes more difficult given more images and increasing sparsity. However, the performance gap between approaches expands with increasing sparsity, indicating the great opportunity for the proposed approach when feedback is lacking.

6.7.4 Comparing Contextual Enhanced Models

To evaluate the effect for incorporating each category of contextual information in recommendation, we present precision and recall at k for each contextual enhanced NPR and

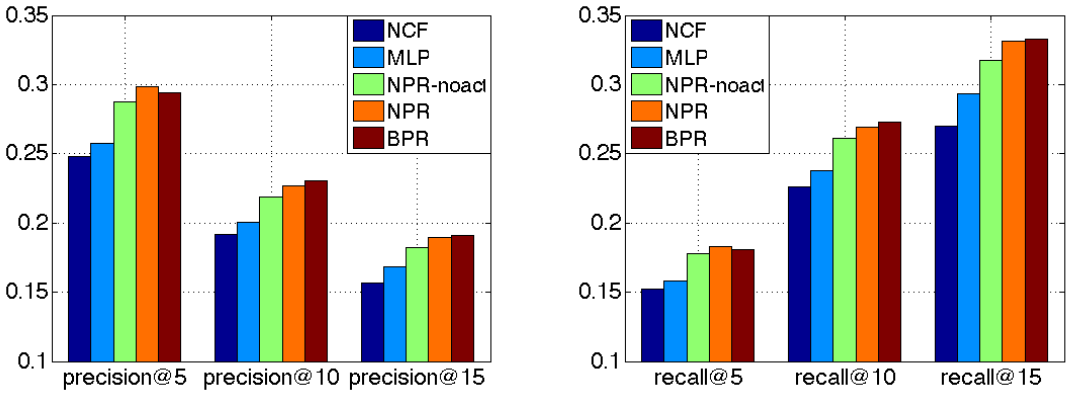


Figure 6.10: Average Precision and Recall for Base Models on Small Dataset

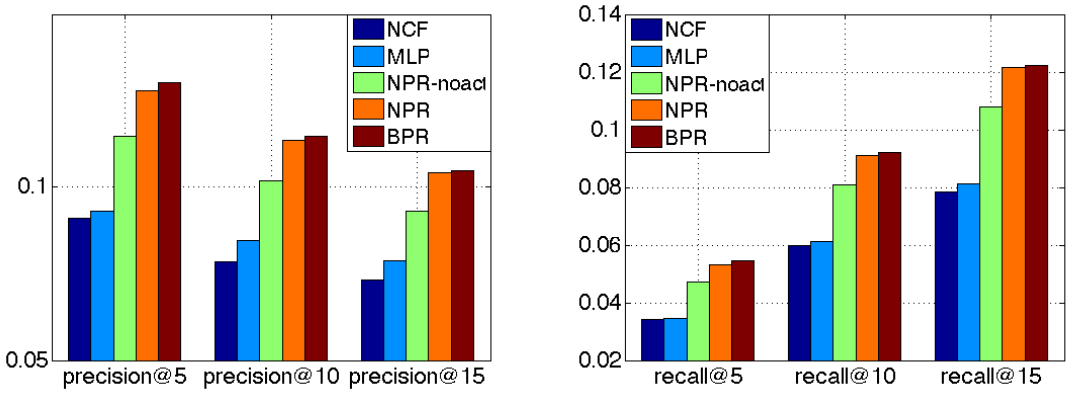


Figure 6.11: Average Precision and Recall for Base Models on Large Dataset

Method	**p@5**	**p@10**	avg Δ	**r@5**	**r@10**	avg Δ
NPR	0.1280	0.1137	-	0.0531	0.0909	-
VNPR	0.1354	0.1177	+4.6%	0.0563	0.0952	+5.4%
TNPR	0.1411	0.1250	+10.1%	0.0599	0.1021	+12.6%
GNPR	0.1326	0.1178	+3.6%	0.0564	0.0953	+5.5%
C-NPR	0.1504	0.1317	+16.6%	0.0644	0.1081	+16.6%

Table 6.2: Integrate Contextual Information in NPR

Method	**p@5**	**p@10**	avg Δ	**r@5**	**r@10**	avg Δ
BPR	0.1302	0.1148	-	0.0544	0.0920	-
VBPR	0.1366	0.1188	+4.2%	0.0577	0.0961	+5.3%
TBPR	0.1384	0.1217	+8.5%	0.0588	0.0992	+8.0%
GBPR	0.1331	0.1171	+2.1%	0.0562	0.0950	+3.3%
C-BPR	0.1445	0.1255	+10.2%	0.0619	0.1034	+13.1%

Table 6.3: Integrate Contextual Information in BPR

BPR models over large dataset in Table 6.2 and 6.3 [3]. We observe that modeling additional contextual factors improves over the basic NPR and BPR method. Concretely, TNPR gives an average improvement of 10.1% in precision and 12.6% in recall over NPR baseline on the large dataset. This indicates that rich textual side knowledge acts as an effective filter for sifting relevant images. VNPR performs slightly better than NPR, with an improvement of 4.6% and 5.4% in precision and recall. The lesson here is learning personal visual preference does help to connect users with images that have appearance agreement. Furthermore, GNPR gives an average of 3.6% percent and 5.5% percent increase in precision and recall. This implies modeling user's geographical region which is consistent with our observation in Section 6.4.1, where we notice the user's strong geographical preference. While on social related image sharing sites, users do have connections focusing around their home location and places they are familiar with. Images in these regions may be more likely to be related with the user. Finally, we observe that C-NPR experiences an average of more than 16% increase in precision and recall. This implies that the proposed model is effective in integrating various categories of contextual information jointly to make better recommendation. We observe similar trends in C-BPR models.

For social enhanced model, we incorporated user group and user contact social information to the model as regularization for user latent factor. Social information been

[3] We delay reporting and discussing in here for models incorporating social signal due to its imperceptible contribution.

proven to contribute to the effectiveness of various prediction and recommendation tasks [89, 90]. However, we observe from Table 6.4 that, for example in BPR based models, it only slightly improves over the base model by an average of 0.22% in precision and 0.87% in recall. And for the overall model, incorporating social only has 0.2% and 0.3% average improvement over its counterpart in precision and recall. However, the computation cost for incorporating social information is very high. From our experience in actual experiment, it almost tripled the time cost of training the model without social incorporated. We thus omit discussion of social enhanced models further in this scenario and we attribute such behavior to that the social ties in Flickr are weak.

Method	**p@5**	**p@10**	avg Δ	**r@5**	**r@10**	avg Δ
BPR	0.1302	0.1148	-	0.0544	0.0920	-
SBPR	0.1302	0.1153	+0.22%	0.0547	0.0931	+0.87%
TVGBPR	0.1445	0.1255	+10.2%	0.0619	0.1034	+13.1%
TVGSBPR	0.1450	0.1256	+10.4%	0.0623	0.1034	+13.4%

Table 6.4: Integrate Social Information in BPR

6.7.5 NPR and BPR with Contextual Information

First, even though the NPR base model performs similarly with BPR, we observe that C-NPR leads by an average of $\sim$4.5% higher precision and recall over C-BPR on the large dataset and $\sim$1.5% increase on the small dataset. We ascribe this improvement to the neural network based model flexibly adjusting weights for each feature dimension and nonlinear activation enriching the expressiveness. Second, the higher increase for the large dataset indicates the C-NPR model could be more beneficial than the C-BPR model for recommendation under the real-world scenario of extreme feedback sparsity.

Method	p@5	p@10	r@15	r@5	r@10	r@15
C-NPR(S)	0.2987	0.2371	0.1977	0.1866	0.2842	0.3471
C-BPR(S)	0.3034	0.2335	0.1945	0.1874	0.2801	0.3419
C-NPR(L)	0.1504	0.1317	0.1192	0.0644	0.1081	0.1430
C-BPR(L)	0.1445	0.1255	0.1141	0.0619	0.1034	0.1363

Table 6.5: Compare Contextual NPR and Contextual BPR

6.7.6 Cold Start

In this experiment, we focus on the cold-start scenario which is commonly encountered in recommendation where we have a limited number of positive user feedbacks for training the model. Here we select users who have fewer than seven liked images to examine the performance of the proposed model on large dataset in cold-start setting. Interestingly we observe in Table 6.6 that the proposed C-NPR model outperforms base NPR model by average $\sim$21% in precision and recall. Additionally, each of contextual models exhibit better performance than the NPR baseline, with TNPR taking the lead showing an average improvement of $\sim$13% in precision and recall. This implies these contextual factors help to alleviate the sparsity in the cold-start setting. Moreover, the lager improvement compared with ordinary setting again validates our claim that the contextual information is especially helpful when feedback is rare.

Method	p@5	p@10	p@15	r@5	r@10	r@15
NPR	0.0723	0.0598	0.0518	0.0643	0.1063	0.1381
VNPR	0.0775	0.0628	0.0554	0.0683	0.1131	0.1455
TNPR	0.0820	0.0678	0.0584	0.0731	0.1206	0.1558
GNPR	0.0775	0.0644	0.0563	0.0685	0.1120	0.1455
C-NPR	0.0893	0.0721	0.0626	0.0769	0.1282	0.1668

Table 6.6: NPR Cold-start Performance

Similarly, for BPR based alternatives, we notice an average of 16% increase in precision@10 and recall@10 by incorporating contextual information in Table 6.7. Topic information contribute most to base model with an average improvement of 8.5% in precision and 6.7% in recall.

Method	**p@5**	**p@10**	**p@15**	**r@5**	**r@10**	**r@15**
BPR	0.0741	0.0602	0.0552	0.0659	0.1069	0.1392
VBPR	0.0800	0.0635	0.0548	0.0712	0.1129	0.1460
TBPR	0.0832	0.0653	0.0562	0.0731	0.1141	0.1468
GBPR	0.0779	0.0630	0.0544	0.0786	0.1043	0.1449
TVGBPR	0.0863	0.0696	0.0591	0.0768	0.1240	0.1577

Table 6.7: BPR Cold-start Performance

6.7.7 Number of training samples

In this section, we explore how performance of different models is influenced by the amount of training data used as well as by the number of negative samples for each positive. As mentioned in Section 6.7.2, the training data generation procedure is as follows: for NPR-1 and BPR-1, we first randomly sample a user, then sample one positive (liked) image for the user and followed by one negative (unobserved/ disliked) image of the user. For NPR-10 and BPR-10 instead, we randomly sample ten negative image for each positive image while keeping other steps the same. The total number of training tuples generated is measured in terms of the number of positive feedbacks in original dataset. In Figure 6.12, the horizontal axis represents the number of times (of positive feedback) to sample and the vertical axis is the F-1 score@10. We observe that BPR-1 and NPR-1 achieve increasing F-1 score with gradual increase in training data. However, the performance of NPR-1 and BPR-1 models have disparate properties. First, the increase for

NPR-based model is relatively gentle, while steeper for BPR based model. Furthermore, the NPR model performs much better, for example, it gains 0.23 for F-1 score@10 at 5 times of sampling while the BPR-based model only reaches 0.17. The difference in performance is more severe when training data is lacking. Interestingly, we also notice that neural network based model generally achieves better performance with inadequate training data. We attribute this to linear model has less powerful expressiveness, hence incurring overfitting more easily and vice versa for nonlinear models. The performance gap doesn't decrease even after we adjust the regularization parameters to their optimal setting. To note, the same phenomenon was observed on the large dataset. After we decoupled the negative samples sampled for training, we notice better F1 score for both approaches, yet the performance curve gradually saturates as we continue serving more training data. This indicates the models would stop improving as size of training data is no longer the bottleneck. To note, all aforementioned results are based on training by sampling 5 times of positive samples and 10 negative samples per positive. To our knowledge, this is the first effort to compare such differences in behavior of these two categories of models, and we hope this finding will provide some reference for further research.

6.7.8 Feature Dimension Reduction

We compare PCA and stacked AE for visual dimension reduction, which are further compared to the model trained with the original 4,096 features. To note, all results reported in this section are based on the BPR-based approach. From Table 6.8, we observe that PCA and the auto-encoder perform competitively to each other, with PCA slightly better on the large dataset and AE better on the small dataset. This implies that the nonlinear nature of AE doesn't make a big difference with linear nature of PCA in this scenario as the original features are already generated with nonlinear CNN. An important finding is that adopting the 128 dimension visual vector achieves remarkably good performance

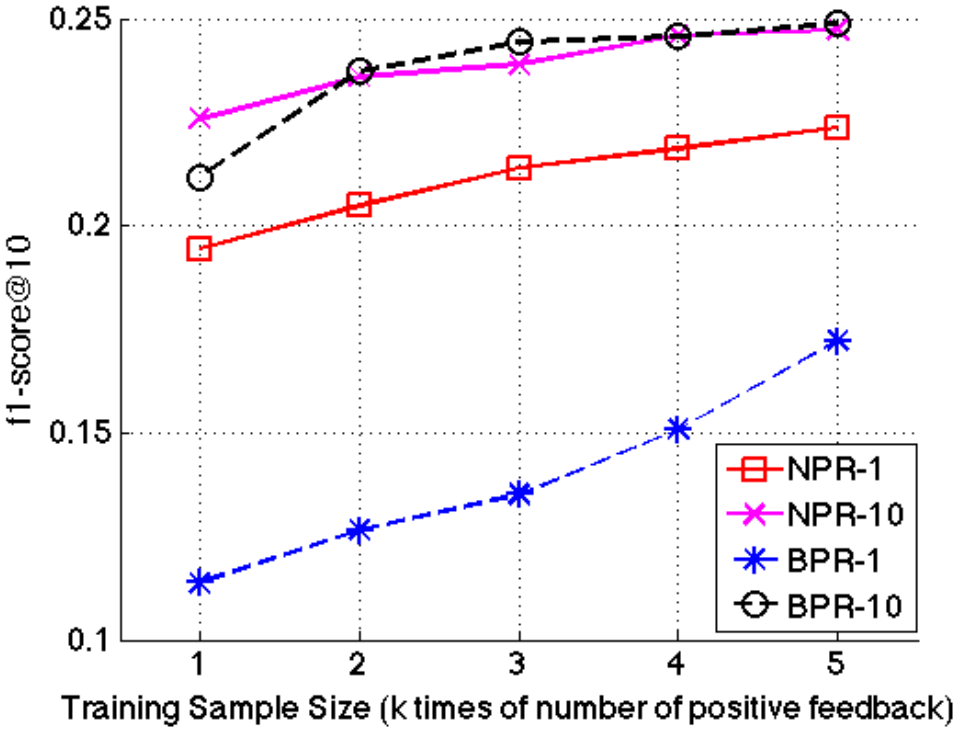


Figure 6.12: Performance w.r.t Training Sample Size

as compared with directly using the 4,096 dimension visual features, although the 128 dimensions only account for 53% of the variance according to PCA.

Approach	Dataset	**F-1@5**	**F-1@10**	**F-1@15**
PCA 128	Large	0.0812	0.1063	0.1171
	Small	0.2281	0.2508	0.2439
Auto Encoder 128	Large	0.0808	0.1062	0.1166
	Small	0.2290	0.2501	0.2441
Original CNN 4096	Large	0.8410	0.1064	0.1173
	Small	0.2279	0.2497	0.2438

Table 6.8: Visual Dimension Reduction

Similarly, we compare LDA[163] and PCA as a form for reducing the dimensionality of tags. LDA discovers latent topic composition of tag set associated with each image. Each identified topic is considered as a new dimension in the new feature space. Table 6.9 shows the results. We observe that feature generated by PCA helps the model performs better on both the large and small datasets. Specifically, we consider 100 dimensions for the small dataset since there are fewer distinct user tags, while we compare 200 and 500 dimensions for the large dataset since tag set is more diverse. We observe LDA generates sparse feature vectors, and the topics it identifies sometimes overlap or difficult to interpret. The reason could be it's challenging to set a proper number of topics and the tag set is noisy. Additionally, we notice with different dimensions of feature vectors generated from PCA, the corresponding model yields different results. For example, the 500 dimensions feature helps reach 0.1193 for F-1@10, better than the 0.1075 achieved by the model using 200 dimensions feature. This is dissimilar to the case for visual features, where reducing the dimension doesn't harm performance much. The reason could be the number of tags are far more than the dimension of visual features, and tags are unorganized and stochastic.

Dataset	**Approach**	**F-1@5**	**F-1@10**	**F-1@15**
Small	PCA 100	0.2271	0.2543	0.2477
	LDA 100	0.2280	0.2514	0.2426
Large	PCA 200	0.8070	0.1075	0.1148
	LDA 200	0.0767	0.1023	0.1130
	PCA 500	0.0825	0.1093	0.1198
	LDA 500	0.0798	0.1069	0.1172

Table 6.9: Topic Dimension Reduction

6.8 Summary

In this chapter, we tackle the problem of personalized image recommendation. We propose *Neural Personalized Ranking (NPR)* – a new neural network based personalized pairwise ranking model for implicit feedback, which incorporates the idea of generalized matrix factorization. We further build an enhanced model by augmenting the basic NPR model with users' multiple contextual preference clues and derive corresponding features that can be incorporated into both the NPR and BPR frameworks to better uncover user preferences. Through extensive experimental validation, we demonstrate the proposed NPR model significantly outperforms several state-of-the-art approaches. Moreover, we observe the superiority of contextual enhanced NPR model over the baseline model.

7. SUMMARY AND FUTURE WORK

In this section, we present a summary of this dissertation and potential future research directions.

7.1 Summary

In this dissertation research, we have focused on the emerging class of geo-social systems, that are fundamentally shaped by the intrinsic mutual reinforcement relationship between geography (geo) and user behaviors (social). These systems are noted for their massive amounts of geo-tagged user activities – including user checkins, user-contributed multimedia (like text, images, and videos), and user tagging of other users and resources. While the capabilities of these geo-social systems are beginning to grow, there is significant research gap in understanding, modeling and leveraging geo-tagged user activities and in identifying the key factors that influence the success of new geo-social systems. Hence, this dissertation has explored three key scenarios – user profiling, ranking and recommendation. In summary, this dissertation has made three unique contributions:

- First, we proposed two frameworks for profiling location and user in geo-social system. Specifically, we proposed and evaluated a community-based framework for tackling the problem of geospatial tag distribution estimation, which is a key component of many new location-augmented search, retrieval, and mining applications. Through experimental investigation, we found that geo-locations have a tendency of sharing similar “ideas" and forming geo-spatial communities. Meanwhile, we demonstrated how our community discovery approach and smoothing strategy leads to high-quality hashtag distribution estimation. Additionally, we introduced a location-sensitive folksonomy generation framework toward the goal of improved user profile estimation. Our key motivating intuition is that spatial variation of user tagging manifests in how users or-

ganize and apply tags, and is critical for building more robust folksonomies. Through extensive experiments, we have demonstrated the impact of such a location-sensitive folksonomy on finding relevant tags for user profiling. We have seen that the location-sensitive folksonomy-informed user profiling is more effective in finding relevant tags, and learning to rank strategy is helpful for optimizing weights of each feature and leads to high quality user profile tags.

- Second, we have proposed and evaluated a geo-spatial learning-to-rank framework for identifying local experts that leverages the fine-grained GPS coordinates of millions of Twitter user and carefully curated Twitter list data. We introduced five categories of features for the proposed learning model, including user-based features, local authority features, tweet content features, list-based features and a group of location-sensitive graph random walk features that capture both the dynamics of expertise propagation and physical distances. We adapt these features into the state-of-the-art learning to rank approach LambdaMart. Through extensive experimental investigation, we find the proposed learning framework which intelligently assigns feature weights can produce significant improvement compared to previous methods. By investigating the features, we found high-quality local expert models can be built with fairly compact features. Additionally, we observed promising results for the reusability of learned models.

- Third, we tackled the problem of personalized image recommendation in geo-social system like Flickr and Instgram as part of our effort toward enhancing geo-social systems. We proposed *Neural Personalized Ranking (NPR)* – a new neural network based personalized pairwise ranking model for implicit feedback, which incorporates the idea of generalized matrix factorization. We further build an enhanced model by augmenting the basic NPR model with users' multiple contextual preference clues and derive corresponding features. These contextual signals – including geographical features, user

tags, and high level image features – can be incorporated into both the NPR and alternative frameworks, such as Bayesian Personalized Ranking, to better uncover user preferences. Through extensive experimental validation, we demonstrated the proposed NPR model significantly outperforms several state-of-the-art approaches. Moreover, we observe the superiority of contextual enhanced NPR model over the base model.

7.2 Further Study

- **profiling, ranking and recommendation.** In chapters 3, 4, 5 and 6, we discussed frameworks for location-aware profiling, ranking and recommendation. There we focus on a single domain, each with an inherently focused source of features for simplicity and compactness of modeling. Future research can look into incorporating alternative source of information into the current frameworks. First, for location profiling, we can study geo-spatial crowds or community formation in alternative social media platforms (e.g., Pinterest) and to incorporate alternative signals of community formation, including activity patterns, temporal changes of idea flow, and topic-sensitive signals (e.g., considering only political hashtags). Second, future work can combine data from different social media platforms to better deal with sparsity in user profiling and expertise retrieval. For example, now there are increasing number of services that accept authorized login with a Twitter account, providing an opportunity for acquiring diverse cross-domain information about each user and complementary evidence of local expertise. Third, we believe recommendation performance will be improved, especially for new users, by incorporating user demographic information (e.g., age, education) and behavior patterns (e.g., views, shares and comments). Additionally, we would like to extend the current model with additional contextual information, for example, modeling the temporal evolution of preferences by revising certain model components with LSTMs.

- **Other location-aware applications.** In this dissertation research, we look at enhancing geo-social systems from three important use cases. Aside from these efforts, there exist many impactful application scenarios where we can potentially leverage the unique characteristics of geo-social systems to further enhance over their existing functionality. For example, real-time location-sensitive emergency detection and response. For example, during Hurricane Harvey, many victims post their locations and images on social media in hope to be rescued. It will be very helpful if we can synthesize some of the key information from these multi-modal data including time, location, text, and image or video and the geo-crowd formation and their behavior to evaluate the extent of such destruction from the overwhelming noise in real-time systems, providing guideline for rescue coordination and organization. This will require algorithms that combine both social media stream analysis capabilities with models of the dynamics of crowd behavior.

REFERENCES

[1] J. Constine, "Facebook now has 2 billion monthly users and responsibility." https://techcrunch.com/2017/06/27/facebook-2-billion-users/, November 2017.

[2] L. Matney, "Youtube has 1.5 billion logged-in monthly users watching a ton of mobile video." https://techcrunch.com/2017/06/22/youtube-has-1-5-billion-logged-in-monthly-users-watching-a-ton-of-mobile-video/, November 2017.

[3] C. Smith, "Youtube has 1.5 billion logged-in monthly users watching a ton of mobile video." http://expandedramblings.com/index.php/by-the-numbers-interesting-foursquare-user-stats/, November 2017.

[4] J. Liu, Z. Li, J. Tang, Y. Jiang, and H. Lu, "Personalized geo-specific tag recommendation for photos on social websites," *IEEE Transactions on Multimedia*, vol. 16, no. 3, pp. 588–600, 2014.

[5] H. Lu and J. Caverlee, "Exploiting geo-spatial preference for personalized expert recommendation," in *Proceedings of the 9th ACM Conference on Recommender Systems*, pp. 67–74, ACM, 2015.

[6] Z. Cheng, J. Caverlee, H. Barthwal, and V. Bachani, "Who is the barbecue king of texas?: a geo-spatial approach to finding local experts on twitter," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pp. 335–344, ACM, 2014.

[7] W. Niu, Z. Liu, and J. Caverlee, "Lexl: A learning approach for local expert discovery on twitter," in *Proceedings of the 35th European Conference on Information*

Retrieval, Springer, 2016.

[8] W. Niu, Z. Liu, and J. Caverlee, "On local expert discovery via geo-located crowds, queries, and candidates," *ACM Trans. Spatial Algorithms Syst.*, vol. 2, pp. 14:1–14:24, Nov. 2016.

[9] S. McClendon and A. C. Robinson, "Leveraging geospatially-oriented social media communications in disaster response," *International Journal of Information Systems for Crisis Response and Management (IJISCRAM)*, vol. 5, no. 1, pp. 22–40, 2013.

[10] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in *Proceedings of the 19th international conference on World wide web*, pp. 851–860, ACM, 2010.

[11] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing: concepts, methodologies, and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, no. 3, p. 38, 2014.

[12] J.-A. Yang, M.-H. Tsou, C.-T. Jung, C. Allen, B. H. Spitzberg, J. M. Gawron, and S.-Y. Han, "Social media analytics and research testbed (smart): Exploring spatiotemporal patterns of human dynamics with geo-targeted social media messages," *Big Data & Society*, vol. 3, no. 1, 2016.

[13] Y. Li, Y. Zheng, H. Zhang, and L. Chen, "Traffic prediction in a bike-sharing system," in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, p. 33, ACM, 2015.

[14] X. Liu, X. Kong, and Y. Li, "Collective traffic prediction with partially observed traffic history using location-based social media," in *Proceedings of the 25th*

ACM International on Conference on Information and Knowledge Management, pp. 2179–2184, ACM, 2016.

[15] Y. Zheng, "Tutorial on location-based social networks," in *Proceedings of the 21st international conference on World wide web, WWW*, vol. 12, 2012.

[16] S. Scellato, A. Noulas, R. Lambiotte, and C. Mascolo, "Socio-spatial properties of online location-based social networks.," in *ICWSM*, vol. 11, pp. 329–336, 2011.

[17] P. Serdyukov, V. Murdock, and R. Van Zwol, "Placing flickr photos on a map," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pp. 484–491, ACM, 2009.

[18] A. X. Zhang, A. Noulas, S. Scellato, and C. Mascolo, "Hoodsquare: Modeling and recommending neighborhoods in location-based social networks," in *Social Computing (SocialCom), 2013 International Conference on*, pp. 69–74, IEEE, 2013.

[19] D. Preoţiuc-Pietro, J. Cranshaw, and T. Yano, "Exploring venue-based city-to-city similarity measures," in *Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing*, p. 16, ACM, 2013.

[20] J. Cranshaw, R. Schwartz, *et al.*, "The livehoods project: Utilizing social media to understand the dynamics of a city," in *Proceedings of the ICWSM*, 2012.

[21] A. Noulas, S. Scellato, C. Mascolo, and M. Pontil, "Exploiting semantic annotations for clustering geographic areas and users in location-based social networks.," *The social mobile web*, vol. 11, no. 2, 2011.

[22] M. Kafsi, H. Cramer, B. Thomee, and D. A. Shamma, "Describing and understanding neighborhood characteristics through online social media," in *Proceedings of the 24th International Conference on World Wide Web*, pp. 549–559, ACM, 2015.

[23] F. Qiu and J. Cho, "Automatic identification of user interest for personalized search," in *Proceedings of the 15th international conference on World Wide Web*, pp. 727–736, ACM, 2006.

[24] A. Ahmed, Y. Low, M. Aly, V. Josifovski, and A. J. Smola, "Scalable distributed inference of dynamic user interests for behavioral targeting," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 114–122, ACM, 2011.

[25] Z. Zhao, Z. Cheng, L. Hong, and E. H. Chi, "Improving user topic interest profiles by behavior factorization," in *Proceedings of the 24th International Conference on World Wide Web*, pp. 1406–1416, ACM, 2015.

[26] I. S. Ribeiro, R. L. Santos, M. A. Gonçalves, and A. H. Laender, "On tag recommendation for expertise profiling: A case study in the scientific domain," in *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pp. 189–198, ACM, 2015.

[27] K. Ikeda, G. Hattori, C. Ono, H. Asoh, and T. Higashino, "Twitter user profiling based on text and community mining for market analysis," *Knowledge-Based Systems*, vol. 51, pp. 35–47, 2013.

[28] J. Li, A. Ritter, and E. H. Hovy, "Weakly supervised user profile extraction from twitter.," in *ACL (1)*, pp. 165–174, 2014.

[29] R. Li, C. Wang, and K. C.-C. Chang, "User profiling in an ego network: co-profiling attributes and relationships," in *Proceedings of the 23rd international conference on World wide web*, pp. 819–830, ACM, 2014.

[30] W. Niu, J. Caverlee, and H. Lu, "Location-sensitive user profiling using crowdsourced labels," in *AAAI*, 2018.

[31] H. Wang, B. Chen, and W.-J. Li, "Collaborative topic regression with social regularization for tag recommendation.," in *IJCAI*, pp. 2719–2725, 2013.

[32] D. Liu, X.-S. Hua, L. Yang, M. Wang, and H.-J. Zhang, "Tag ranking," in *Proceedings of the 18th international conference on World wide web*, pp. 351–360, ACM, 2009.

[33] B. Sigurbjörnsson and R. Van Zwol, "Flickr tag recommendation based on collective knowledge," in *Proceedings of the 17th international conference on World Wide Web*, pp. 327–336, ACM, 2008.

[34] S. Tuarob, L. C. Pouchard, and C. L. Giles, "Automatic tag recommendation for metadata annotation using probabilistic topic modeling," in *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries*, pp. 239–248, ACM, 2013.

[35] X. Xia, D. Lo, X. Wang, and B. Zhou, "Tag recommendation in software information sites," in *Mining Software Repositories (MSR), 2013 10th IEEE Working Conference on*, pp. 287–296, IEEE, 2013.

[36] R. Krestel, P. Fankhauser, and W. Nejdl, "Latent dirichlet allocation for tag recommendation," in *Proceedings of the third ACM conference on Recommender systems*, pp. 61–68, ACM, 2009.

[37] P. Heymann, D. Ramage, and H. Garcia-Molina, "Social tag prediction," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 531–538, ACM, 2008.

[38] Y. Song, B. Qiu, and U. Farooq, "Hierarchical tag visualization and application for tag recommendations," in *Proceedings of the 20th ACM international conference on Information and knowledge management*, pp. 1331–1340, ACM, 2011.

[39] C. Verma, V. Mahadevan, N. Rasiwasia, G. Aggarwal, R. Kant, A. Jaimes, and S. Dey, "Construction and evaluation of ontological tag trees," *Expert Systems with Applications*, vol. 42, no. 24, pp. 9587–9602, 2015.

[40] E. Zangerle, W. Gassler, and G. Specht, "Recommending#-tags in twitter," in *Proceedings of the Workshop on Semantic Adaptive Social Web (SASWeb 2011). CEUR Workshop Proceedings*, vol. 730, pp. 67–78, 2011.

[41] F. Godin, V. Slavkovikj, W. De Neve, B. Schrauwen, and R. Van de Walle, "Using topic models for twitter hashtag recommendation," in *Proceedings of the 22nd International Conference on World Wide Web*, pp. 593–596, ACM, 2013.

[42] K. Y. Kamath, J. Caverlee, K. Lee, and Z. Cheng, "Spatio-temporal dynamics of online memes: a study of geo-tagged tweets," in *Proceedings of the 22nd international conference on World Wide Web*, pp. 667–678, ACM, 2013.

[43] Z. Ma, A. Sun, and G. Cong, "Will this# hashtag be popular tomorrow?," in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pp. 1173–1174, ACM, 2012.

[44] A. Kaltenbrunner, S. Scellato, Y. Volkovich, D. Laniado, D. Currie, E. J. Jutemar, and C. Mascolo, "Far from the eyes, close on the web: impact of geographic distance on online social interactions," in *Proceedings of the 2012 ACM workshop on Workshop on online social networks*, pp. 19–24, ACM, 2012.

[45] L. Backstrom, J. Kleinberg, R. Kumar, and J. Novak, "Spatial variation in search engine queries," in *Proceedings of the 17th international conference on World Wide Web*, pp. 357–366, ACM, 2008.

[46] H. Zhang, M. Korayem, E. You, and D. J. Crandall, "Beyond co-occurrence: discovering and visualizing tag relationships from geo-spatial and temporal similarities,"

in *Proceedings of the fifth ACM international conference on Web search and data mining*, pp. 33–42, ACM, 2012.

[47] Z. Cheng, J. Caverlee, and K. Lee, "You are where you tweet: a content-based approach to geo-locating twitter users," in *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp. 759–768, ACM, 2010.

[48] A. Brodersen, S. Scellato, and M. Wattenhofer, "Youtube around the world: geographic popularity of videos," in *Proceedings of the 21st international conference on World Wide Web*, pp. 241–250, ACM, 2012.

[49] L. Hu, A. Sun, and Y. Liu, "Your neighbors affect your ratings: on geographical neighborhood influence to rating prediction," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pp. 345–354, ACM, 2014.

[50] L. Backstrom, E. Sun, and C. Marlow, "Find me if you can: improving geographical prediction with social and spatial proximity," in *Proceedings of the 19th international conference on World wide web*, pp. 61–70, ACM, 2010.

[51] D. Jurgens, T. Finethy, J. McCorriston, Y. T. Xu, and D. Ruths, "Geolocation prediction in twitter using social networks: A critical analysis and review of current practice.," in *ICWSM*, vol. 15, pp. 188–197, 2015.

[52] H. Saif, Y. He, and H. Alani, "Alleviating data sparsity for twitter sentiment analysis," in *CEUR Workshop Proceedings (CEUR-WS. org)*, 2012.

[53] J. Lin, R. Snow, and W. Morgan, "Smoothing techniques for adaptive online language models: topic tracking in tweet streams," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 422–429, ACM, 2011.

[54] J. Lin, R. Snow, and W. Morgan, "Smoothing techniques for adaptive online language models: topic tracking in tweet streams," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 422–429, ACM, 2011.

[55] C. Zhai and J. Lafferty, "A study of smoothing methods for language models applied to information retrieval," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 2, pp. 179–214, 2004.

[56] K. Balog, Y. Fang, M. de Rijke, P. Serdyukov, L. Si, *et al.*, "Expertise retrieval," *Foundations and Trends® in Information Retrieval*, vol. 6, no. 2–3, pp. 127–256, 2012.

[57] N. Craswell, A. P. de Vries, and I. Soboroff, "Overview of the trec 2005 enterprise track.," in *Trec*, vol. 5, pp. 199–205, 2005.

[58] M. Bouguessa, B. Dumoulin, and S. Wang, "Identifying authoritative actors in question-answering forums: the case of yahoo! answers," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 866–874, ACM, 2008.

[59] J. Guo, S. Xu, S. Bao, and Y. Yu, "Tapping on the potential of q&a community by recommending answer providers," in *Proceedings of the 17th ACM conference on Information and knowledge management*, pp. 921–930, ACM, 2008.

[60] A. Pal, F. M. Harper, and J. A. Konstan, "Exploring question selection bias to identify experts and potential experts in community question answering," *ACM Transactions on Information Systems (TOIS)*, 2012.

[61] A. Alkouz, E. W. De Luca, and S. Albayrak, "Latent semantic social graph model for expert discovery in facebook.," in *IICS*, pp. 128–138, 2011.

[62] K. Balog, L. Azzopardi, and M. De Rijke, "Formal models for expert finding in enterprise corpora," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 43–50, ACM, 2006.

[63] B. Dom, I. Eiron, A. Cozzi, and Y. Zhang, "Graph-based ranking algorithms for e-mail expertise analysis," in *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pp. 42–48, ACM, 2003.

[64] R. Yeniterzi and J. Callan, "Constructing effective and efficient topic-specific authority networks for expert finding in social media," in *Proceedings of the first international workshop on Social media retrieval and analysis*, pp. 45–50, ACM, 2014.

[65] J. Zhang, M. S. Ackerman, and L. Adamic, "Expertise networks in online communities: structure and algorithms," in *Proceedings of the 16th international conference on World Wide Web*, pp. 221–230, ACM, 2007.

[66] C. S. Campbell, P. P. Maglio, A. Cozzi, and B. Dom, "Expertise identification using email communications," in *Proceedings of the twelfth international conference on Information and knowledge management*, pp. 528–531, ACM, 2003.

[67] A. Bozzon, M. Brambilla, S. Ceri, M. Silvestri, and G. Vesci, "Choosing the right crowd: expert finding in social networks," in *Proceedings of the 16th International Conference on Extending Database Technology*, pp. 637–648, ACM, 2013.

[68] G. A. Wang, J. Jiao, A. S. Abrahams, W. Fan, and Z. Zhang, "Expertrank: A topic-aware expert finding algorithm for online knowledge communities," *Decision Support Systems*, vol. 54, no. 3, pp. 1442–1451, 2013.

[69] J. Weng, E.-P. Lim, J. Jiang, and Q. He, "Twitterrank: finding topic-sensitive influential twitterers," in *Proceedings of the third ACM international conference on Web search and data mining*, pp. 261–270, ACM, 2010.

[70] S. Ghosh, N. Sharma, F. Benevenuto, N. Ganguly, and K. Gummadi, "Cognos: crowdsourcing search for topic experts in microblogs," in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pp. 575–590, ACM, 2012.

[71] W. Li, C. Eickhoff, and A. P. de Vries, "Geo-spatial domain expertise in microblogs.," in *ECIR*, pp. 487–492, Springer, 2014.

[72] G. Cong, C. S. Jensen, and D. Wu, "Efficient retrieval of the top-k most relevant spatial web objects," *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 337–348, 2009.

[73] T.-Y. Liu *et al.*, "Learning to rank for information retrieval," *Foundations and Trends® in Information Retrieval*, vol. 3, no. 3, pp. 225–331, 2009.

[74] L. Hang, "A short introduction to learning to rank," *IEICE Transactions on Information and Systems*, 2011.

[75] Z. Yang, J. Tang, B. Wang, J. Guo, J. Li, and S. Chen, "Expert2bole: From expert finding to bole search," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,(KDDâĂŹ09)*, pp. 1–4, 2009.

[76] C. Moreira, P. Calado, and B. Martins, "Learning to rank for expert search in digital libraries of academic publications," in *Progress in Artificial Intelligence: 15th Portuguese Conference on Artificial Intelligence*, Springer, 2011.

[77] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang, "One-class collaborative filtering," in *Data Mining, 2008. ICDM'08. Eighth IEEE International*

Conference on, pp. 502–511, IEEE, 2008.

[78] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pp. 263–272, Ieee, 2008.

[79] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pp. 452–461, AUAI Press, 2009.

[80] J. Fan, D. A. Keim, Y. Gao, H. Luo, and Z. Li, "Justclick: Personalized image recommendation via exploratory search from large-scale flickr images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 2, pp. 273–288, 2009.

[81] Y. Jing, X. Zhang, L. Wu, J. Wang, Z. Feng, and D. Wang, "Recommendation on flickr by combining community user ratings and item importance," in *Multimedia and Expo (ICME), 2014 IEEE International Conference on*, pp. 1–6, IEEE, 2014.

[82] Y. Li, J. Luo, and T. Mei, "Personalized image recommendation for web search engine users," in *Multimedia and Expo (ICME), 2014 IEEE International Conference on*, pp. 1–6, IEEE, 2014.

[83] J. Sang and C. Xu, "Right buddy makes the difference: An early exploration of social relation analysis in multimedia applications," in *Proceedings of the 20th ACM international conference on Multimedia*, pp. 19–28, ACM, 2012.

[84] Y. Li, T. Mei, Y. Cong, and J. Luo, "User-curated image collections: Modeling and recommendation," in *Big Data (Big Data), 2015 IEEE International Conference on*, pp. 591–600, IEEE, 2015.

[85] S. Liu, P. Cui, W. Zhu, S. Yang, and Q. Tian, "Social embedding image distance learning," in *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 617–626, ACM, 2014.

[86] X. Liu, M.-H. Tsai, and T. Huang, "Analyzing user preference for social image recommendation," *arXiv preprint arXiv:1604.07044*, 2016.

[87] C. Lei, D. Liu, W. Li, Z.-J. Zha, and H. Li, "Comparative deep learning of hybrid representations for image recommendations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2545–2553, 2016.

[88] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in *Recommender systems handbook*, pp. 217–253, Springer, 2011.

[89] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proceedings of the fourth ACM conference on Recommender systems*, pp. 135–142, ACM, 2010.

[90] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proceedings of the fourth ACM international conference on Web search and data mining*, pp. 287–296, ACM, 2011.

[91] A. Q. Macedo, L. B. Marinho, and R. L. Santos, "Context-aware event recommendation in event-based social networks," in *Proceedings of the 9th ACM Conference on Recommender Systems*, pp. 123–130, ACM, 2015.

[92] A. Van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," in *Advances in neural information processing systems*, pp. 2643–2651, 2013.

[93] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel, "Image-based recommendations on styles and substitutes," in *Proceedings of the 38th International ACM*

SIGIR Conference on Research and Development in Information Retrieval, pp. 43–52, ACM, 2015.

[94] R. He and J. McAuley, "Vbpr: Visual bayesian personalized ranking from implicit feedback.," in *AAAI*, pp. 144–150, 2016.

[95] J.-D. Zhang and C.-Y. Chow, "Geosoca: Exploiting geographical, social and categorical correlations for point-of-interest recommendations," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 443–452, ACM, 2015.

[96] C. Cheng, H. Yang, I. King, and M. R. Lyu, "Fused matrix factorization with geographical and social influence in location-based social networks.," in *AAAI*, vol. 12, pp. 17–23, 2012.

[97] D. Lian, C. Zhao, X. Xie, G. Sun, E. Chen, and Y. Rui, "Geomf: joint geographical modeling and matrix factorization for point-of-interest recommendation," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 831–840, ACM, 2014.

[98] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pp. 325–334, ACM, 2011.

[99] B. Liu, Y. Fu, Z. Yao, and H. Xiong, "Learning geographical preferences for point-of-interest recommendation," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1043–1051, ACM, 2013.

[100] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *WWW*, pp. 173–182, ACM, 2017.

[101] M. Trofimov, S. Sidana, O. Horodnitskii, C. Laclau, Y. Maximov, and M.-R. Amini, "Representation learning and pairwise ranking for implicit and explicit feedback in recommendation systems," *arXiv preprint arXiv:1705.00105*, 2017.

[102] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *RecSys*, pp. 191–198, ACM, 2016.

[103] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, *et al.*, "Wide & deep learning for recommender systems," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pp. 7–10, ACM, 2016.

[104] A. M. Elkahky, Y. Song, and X. He, "A multi-view deep learning approach for cross domain user modeling in recommendation systems," in *WWW*, ACM, 2015.

[105] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, "Collaborative denoising auto-encoders for top-n recommender systems," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pp. 153–162, ACM, 2016.

[106] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," *arXiv preprint arXiv:1511.06939*, 2015.

[107] W. Niu, J. Caverlee, and H. Lu, "Neural personalized ranking for image recommendation," in *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*, ACM, 2018.

[108] B. Shaw, J. Shea, S. Sinha, and A. Hogue, "Learning to rank for spatiotemporal search," in *Proceedings of the sixth ACM international conference on Web search and data mining*, pp. 717–726, ACM, 2013.

[109] J. Teevan, A. Karlson, S. Amini, A. Brush, and J. Krumm, "Understanding the importance of location, time, and people in mobile local search behavior," in *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, pp. 77–80, ACM, 2011.

[110] L. B. Baltussen, M. Büchi, *et al.*, "One Percent of Twitter, Part II: Geotags, Text Analysis, and Event Profiling." `https://www.researchgate.net/publication/307934068_One_Percent_of_Twitter_Part_II_Geotags_Text_Analysis_and_Event_Profiling`, November 2017.

[111] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annual review of sociology*, vol. 27, no. 1, pp. 415–444, 2001.

[112] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[113] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.

[114] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *science*, vol. 315, no. 5814, pp. 972–976, 2007.

[115] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027–1035, Society for Industrial and Applied Mathematics, 2007.

[116] W. R. Tobler, "A computer movie simulating urban growth in the detroit region," *Economic geography*, vol. 46, no. sup1, pp. 234–240, 1970.

[117] R. Sinnott, "Virtues of the haversine.," 1984.

[118] J. W. Hager, J. F. Behensky, and B. W. Drew, "The universal grids: Universal transverse mercator (utm) and universal polar stereographic (ups)," tech. rep.,

DEFENSE MAPPING AGENCY HYDROGRAPHIC/TOPOGRAPHIC CENTER WASHINGTON DC, 1989.

[119] J. Lin, "Divergence measures based on the shannon entropy," *IEEE Transactions on Information theory*, vol. 37, no. 1, pp. 145–151, 1991.

[120] K. Y. Kamath and J. Caverlee, "Spatio-temporal meme prediction: learning what hashtags will be popular where," in *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pp. 1341–1350, ACM, 2013.

[121] D. Pelleg, A. W. Moore, *et al.*, "X-means: Extending k-means with efficient estimation of the number of clusters.," in *ICML*, vol. 1, pp. 727–734, 2000.

[122] Q. Liu, E. Chen, H. Xiong, C. H. Ding, and J. Chen, "Enhancing collaborative filtering by user interest expansion via personalized ranking," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 1, pp. 218–233, 2012.

[123] A. Majumder and N. Shrivastava, "Know your personalization: learning topic level personalization in online services," in *Proceedings of the 22nd international conference on World Wide Web*, pp. 873–884, ACM, 2013.

[124] L. Hong, A. S. Doumith, and B. D. Davison, "Co-factorization machines: modeling user interests and predicting individual decisions in twitter," in *Proceedings of the sixth ACM international conference on Web search and data mining*, pp. 557–566, ACM, 2013.

[125] M. Jiang, P. Cui, R. Liu, Q. Yang, F. Wang, W. Zhu, and S. Yang, "Social contextual recommendation," in *Proceedings of the 21st ACM international conference on Information and knowledge management*, pp. 45–54, ACM, 2012.

[126] H. Yin, B. Cui, L. Chen, Z. Hu, and Z. Huang, "A temporal context-aware model for user behavior modeling in social media systems," in *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pp. 1543–1554, ACM, 2014.

[127] E. Zhong, N. Liu, Y. Shi, and S. Rajan, "Building discriminative user profiles for large-scale content recommendation," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2277–2286, ACM, 2015.

[128] P. Bhattacharya, S. Ghosh, J. Kulshrestha, M. Mondal, M. B. Zafar, N. Ganguly, and K. P. Gummadi, "Deep twitter diving: Exploring topical groups in microblogs at scale," in *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pp. 197–210, ACM, 2014.

[129] V. Rakesh, D. Singh, B. Vinzamuri, and C. K. Reddy, "Personalized recommendation of twitter lists using content and network information.," in *ICWSM*, 2014.

[130] C. H. Brooks and N. Montanez, "Improved annotation of the blogosphere via autotagging and hierarchical clustering," in *Proceedings of the 15th international conference on World Wide Web*, pp. 625–632, ACM, 2006.

[131] X. Zhu, Z. Ming, Y. Hao, and X. Zhu, "Tackling data sparseness in recommendation using social media based topic hierarchy modeling.," in *IJCAI*, pp. 2415–2423, 2015.

[132] S. Wang, J. Tang, Y. Wang, and H. Liu, "Exploring implicit hierarchical structures for recommender systems.," in *IJCAI*, pp. 1813–1819, 2015.

[133] P. Heymann and H. Garcia-Molina, "Collaborative creation of communal hierarchical taxonomies in social tagging systems," tech. rep., Stanford, 2006.

[134] Y.-J. Chu, "On the shortest arborescence of a directed graph," *Science Sinica*, vol. 14, pp. 1396–1400, 1965.

[135] T. Joachims, "Optimizing search engines using clickthrough data," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 133–142, ACM, 2002.

[136] M. Lui and T. Baldwin, "langid. py: An off-the-shelf language identification tool," in *Proceedings of the ACL 2012 system demonstrations*, pp. 25–30, Association for Computational Linguistics, 2012.

[137] E. H. Chi, "Who knows?: searching for expertise on the social web: technical perspective," *Communications of the ACM*, vol. 55, no. 4, pp. 110–110, 2012.

[138] X. Liu, W. B. Croft, and M. Koll, "Finding experts in community-based question-answering services," in *Proceedings of the 14th ACM international conference on Information and knowledge management*, pp. 315–316, ACM, 2005.

[139] A. Pal and S. Counts, "Identifying topical authorities in microblogs," in *Proceedings of the fourth ACM international conference on Web search and data mining*, pp. 45–54, ACM, 2011.

[140] J. Zhang, J. Tang, and J. Li, "Expert finding in a social network," in *International Conference on Database Systems for Advanced Applications*, pp. 1066–1069, Springer, 2007.

[141] K. Zickuhr, "Location-based services," *Pew Internet and American Life Project*, 2013.

[142] C. Burges, K. Svore, P. Bennett, A. Pastusiak, and Q. Wu, "Learning to rank using an ensemble of lambda-gradient models," in *Proceedings of the Learning to Rank Challenge*, pp. 25–35, 2011.

[143] Q. Wu, C. J. Burges, K. M. Svore, and J. Gao, "Ranking, boosting, and model adaptation," *Tecnical Report, MSR-TR-2008-109*, 2008.

[144] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM (JACM)*, vol. 46, no. 5, pp. 604–632, 1999.

[145] F. Provost and P. Domingos, "Tree induction for probability-based ranking," *Machine learning*, vol. 52, no. 3, pp. 199–215, 2003.

[146] C. L. Clarke, G. V. Cormack, and E. A. Tudhope, "Relevance ranking for one to three term queries," *Information processing & management*, vol. 36, no. 2, pp. 291–311, 2000.

[147] G. Kazai, J. Kamps, M. Koolen, and N. Milic-Frayling, "Crowdsourcing for book search evaluation: impact of hit design on comparative system ranking," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pp. 205–214, ACM, 2011.

[148] J. L. Fleiss, J. Cohen, and B. Everitt, "Large sample standard errors of kappa and weighted kappa.," *Psychological Bulletin*, vol. 72, no. 5, p. 323, 1969.

[149] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.

[150] L. A. Adamic, J. Zhang, E. Bakshy, and M. S. Ackerman, "Knowledge sharing and yahoo answers: everyone knows something," in *Proceedings of the 17th international conference on World Wide Web*, pp. 665–674, ACM, 2008.

[151] L. Mamykina, B. Manoim, M. Mittal, G. Hripcsak, and B. Hartmann, "Design lessons from the fastest q&a site in the west," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 2857–2866, ACM, 2011.

[152] G. Wang, K. Gill, M. Mohanlal, H. Zheng, and B. Y. Zhao, "Wisdom in the social crowd: an analysis of quora," in *Proceedings of the 22nd international conference on World Wide Web*, pp. 1341–1352, ACM, 2013.

[153] D. Coetzee, A. Fox, M. A. Hearst, and B. Hartmann, "Should your mooc forum use a reputation system?," in *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pp. 1176–1187, ACM, 2014.

[154] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1235–1244, ACM, 2015.

[155] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 353–362, ACM, 2016.

[156] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, 2009.

[157] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[158] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, "Yfcc100m: The new data in multimedia research," *Communications of the ACM*, vol. 59, no. 2, pp. 64–73, 2016.

[159] W. Niu, J. Caverlee, H. Lu, and K. Kamath, "Community-based geospatial tag estimation," in *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on*, pp. 279–286, IEEE, 2016.

[160] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[161] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International Conference on Machine Learning*, pp. 2048–2057, 2015.

[162] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in neural information processing systems*, pp. 487–495, 2014.

[163] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.