Social Network based Travel Analysis using User Trust Behavior Model

D.Ravi[1], T.K.P. Rajagopal[2], V. Gowri Shankar[3], B.Vijay[4], S.Vinoth Kumar[5]
[1]*Associate Professor, Department of CSE, Kathir College of Engineering, Coimbatore, India*
[2]*Associate Professor & HOD, Department of CSE, Kathir College of Engineering, Coimbatore, India*
[3,4,5]*B. E. Student, Department of CSE, Kathir College of Engineering, Coimbatore, India*

Abstract—**With the popularity of social media (e.g., Facebook and Flicker), users can easily share their check-in records and photos during their trips. In view of the huge number of user historical mobility records in social media, we aim to discover travel experiences to facilitate trip planning. When planning a trip, users always have specific preferences regarding their trips. Instead of restricting users to limited query options such as locations, activities, or time periods, we consider arbitrary text descriptions as keywords about personalized requirements. Advanced works have extended on mining and ranking routes that exist from checked-in data. To meet the need for automatic trip organization, we claim that more features of Places of Interest (POIs) should be extracted. A framework that use collection of data from travellers records of mobility. We have designed a keyword extraction module to classify the POI-related tags, for effective matchings. We have further designed a route reconstruction algorithm to construct route candidates that fulfill the requirements. To provide befitting query results, we explore Representative Skyline concepts, that is, the Skyline routes which best describe the trade-offs among different POI features. To evaluate the effectiveness and efficiency of the proposed algorithms, we have conducted extensive experiments on real location-based social network datasets, and the experiment results show that our methods do indeed demonstrate good performance compared to state-of-the-art works.**

Index Terms— route reconstruction algorithm, social network

I. Introduction

LOCATION-BASED social network (LBSN) services allow users to perform check-in and share their check-in data with their friends. In particular, when a user is traveling, the check-in data are in fact a travel route with some photos and tag information. As a result, a massive number of routes are generated, which play an essential role in many well-established research areas, such as mobility prediction, urban planning and traffic management. If we focus on trip planning and intend to discover travel experiences from shared data in location-based social media. To facilitate trip planning, the advanced works in provide an interface in which a user could submit the block and the total travel time. In contrast, we consider a scenario where users specify their preferences with keywords. However, the query results of existing travel routes usually rank the routes simply by the importance or the number of uploads of routes. For such ranking, the existing works derive a scoring function, where each route will have one score according to its features (e.g., the number of Places of Interest, the popularity of places). Usually, the query results will have similar routes. Recently, aimed to retrieve a greater diversity of routes based on the travel factors considered. As high scoring routes are often too similar to each other, this work considers the diversity of results by exploiting Skyline query. We develop a Keyword-aware Representative Travel Route framework to retrieve several recommended routes where keyword means the personalized requirements that users have for the trip. The route dataset could be built from the collection of low-sampling check-in records.

1) Existing system:

Text Mining, Preprocessing, And Tagging. A data collection, analysis, and processing tree was developed in Rapid miner (www.rapidminer.com) to discover the most frequent words (positive, negative, and side effects) to find their term-frequency-inverse document frequency (TF-IDF) scores within each post. The dataset was uploaded (“Read Excel”), processed (“Process Documents to Data”) using subcomponents (“Extract Content,” “Tokenize,” “Transform Cases,” “Filter Stop words,” “Filter Tokens,” respectively) that filtered excess noise (misspelled words, common stop words, etc.) to ensure measurable variable uniformity. The result (“Processed Data”) contained the final word list, with each word containing a specific TF-IDF score.

2) Semantic similarity of user postings:

The TF-IDF scores in each post were built based on a representative word set present throughout the forum and reflects the posts’ semantic content. Therefore, we viewed a TF-IDF vector as the semantic profile of each post. Consequently, various measures of similarities can be derived to reflect how close the semantic profiles of two posts are, e.g., Euclidian distance or correlation. Additionally, clustering analysis can be performed to identify groups of similar semantic profiles. We used k-means clustering to roughly group the semantic profiles of all posts from our forum, as a preprocessing step necessary for the network-based modelling.

3) Network-based modelling of forum postings:

Forum posting activity consisting of threads containing thousands of postings and replies were modelled into a large user-centric network. The modelling approach aimed at reflecting user interactions while simultaneously considering the posts’ semantic content. The users in our network connect directed edges to two different types of interactions: direct and context interactions. Direct interactions correspond to direct user-to-user replies using the forum's "Reply" option. These interactions were modelled with bidirectional edges connecting the two corresponding nodes. This allows the exchange of information between a poster and a direct replier. Con-text interactions reflect users posting within a specific thread (threads are topic-specific, and thread semantic content is homogeneous). Thus, one directional edges were used to connect starters to all other users within a specific thread. This provides the information transfer from thread initiators to users posting within the respective thread.

II. Proposed System

1) Text categorization:

The major attempt to detect the overall polarity of a sentence, paragraph, or text span, regardless of the entities mentioned (e.g., hotels, restaurants) and their aspects (e.g., food, service). By contrast, this task is concerned with aspect based opinion analysis (ABSA), where the goal is to identify the aspects of given target entities and the opinion expressed towards each aspect. Datasets having customer reviews with human annotations identifying the aspects of the target entities and the opinion polarity of each aspect will be provided.

2) ANN – Artificial Neural Network:

This training method contains bag of words which can be update by admin using Artificial Neural Network this process in called as Data training. This is the basic input for the text data training purpose. Also here all data will be uploaded in to a centralized server for data analysis purpose. Centralized data distribution systems defined here as systems that allow distributed end-user applications, databases and data providers to be integrated with dedicated data sources. Even thou this project will be implemented using ASP.NET, basically it will satisfies all the web based procedures and applications. In case of the organization may having more than two branches in various locations, (i.e.) in different states.

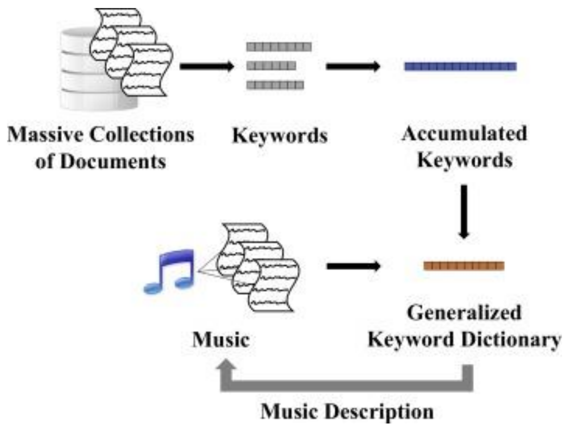
Massive Collections of Documents
Keywords
Accumulated Keywords
Music
Generalized Keyword Dictionary
Music Description

Fig. 1. Efficient keyword search

In statistical modelling, a training set is used to fit a model that can be used to predict a "response value" from one or more "predictors."

Statistical models used for prediction are often called regression models, of which linear regression and logistic regression are two examples. In these fields, a major emphasis is placed on avoiding over fitting, so as to achieve the best possible performance on an independent **test set** that follows the same probability distribution as the training set.

III. Framework Overview

In this section, the proposed framework KSTR is presented. KSTR is comprised of two modules: the offline pattern discovery and scoring module and the online travel routes exploration module. Offline Pattern Discovery and Scoring Module. Given an LBSN dataset, we first analyze the tags of each POI to determine the semantic meaning of the keywords, which are classified into (i) Geo-specific keywords, (ii) Temporal keywords, and (iii) Attribute keywords according to their characteristics. Furthermore, we derive the feature scores of the POIs and generate proper candidate travel routes. Online Travel Routes Exploration Modules. In this module, we aim to provide an interface for users to specify query ranges and preference-related keywords. Once the system receives a specified range and time, the online module will retrieve those travel routes that overlap the query range and the stay time period.

This section describes an offline process of pattern discovery from trajectory histories, which includes (1) the scoring mechanism for keywords and POIs; (2) a review of feature scoring methods that quantify the goodness of the routes; and (3) the candidate route generation algorithm.

1) Keyword Extraction: In this section, we present how we extract the semantic meaning of the keywords and propose a matched score to describe the degree of connection between keywords and trajectories. The keyword extraction module first computes the spatial, temporal and attribute scores for every keyword win the corpus. At query time, each query keyword will be matched to the pre-computed score of matching w.

A) Geo-Specific Keywords:

Some tags are specific to a location, which represents its spatial nature. To quantify the geo-specificity of a tag, an external database identifies geo-terms in the overall tag set and then the tag distribution on the map rates the identified geo-terms. Specifically, to identify name tags, we leverage an external geo-database. In Microsoft Bing services, Geo-code Dataflow API (GDA) can query large numbers of geo-terms to get their representative locations and addresses. For a tag w, using GDA, if its location (latitude; longitude) is returned, and 0 otherwise. Then, using the geographic distribution of the tags, we can find place-level geo-terms like 'Taipei101' in noisy geo-terms. Country-level geo-terms like 'USA' and city-level geo-terms like 'Seattle' are far more widely distributed on the globe than place-level geo-terms.

B) Temporal Keywords:

Some tags are specific to a time interval, which represents its temporal nature. To quantify the temporal-specificity of a tag, time distribution on a tag rates the identified temporal terms. Using the time distribution of tags, we can find tags associated with a specific time interval like 'sunset'. Tags independent of time like 'Taipei' are far more widely distributed in time than time-specific tags.

C) Attribute Keywords:

To find attribute keywords, we consider tags frequently associated with a POI (TF), while not with so many other POIs (IDF). To quantify the relevance between a tag and a POI, we define a “document” as an estimated check-in set I_p of p. Using this POI-driven knowledge, our scoring conveys the POI semantic information in both TF and IDF.

2) Passive Check-Ins:

In previous sections, we worked with check-ins generated by users manually recording their whereabouts, such as foursquare check-ins of visiting Taipei 101. However, some such whereabouts are only passively recorded, such as photos of Taipei 101. Considering that six billion public photos have been uploaded in Facebook and more than 3 percent of photos have geo data, the volume of geo-tagged photos is 2.5 times larger than that of active check-ins. In addition, they capture locations that cannot be covered by active check-ins, such as new restaurants yet to be registered at Foursquare DB. We study how such passive check-ins can be leveraged, by extending our framework KSTR.

3) Feature Scoring Methods:

With a set of travel route records, feature scoring should be considered to find proper recommendations. In this paper, we also explore three travel factors: “Where: people tend to visit popular POIs”, “When: each POI has its proper visiting time”, and “Who: people might follow social-connected friends’ footsteps”. To achieve the “Where, When, Who” consideration issue of user demands, the pattern discovery and scoring module defines the ranking mechanism for each POI with global attractiveness, proper visiting time and geo-social influence.

4) Candidate Route Generation:

In the previous sections, we have proposed the methods for matching raw texts to POI features and mining preference patterns in existing travel routes. However, the route data-set sometimes may not include all the query criteria, and may have bad connections to the query keywords. Thus, we propose the Candidate Route Generation algorithm to com-bine different routes to increase the amount and diversity. The new candidate routes are constructed by combining the subsequences of trajectories. Here we introduce the pre-processing method first.

Pre-Processing: With the information that a trajectory T_i consists of sequence of Combined Points Should be ordered by Time. Since tail.time must be larger than head. Time, p_k. time should be larger than p_i. Time in order to replace p_j with p_k. DFS-Based Route Enumeration. In order to generate all possible routes from their original trajectories. We consider all the POIs in the headset as the source, and explore as far as possible along each link before backtracking.

Algorithm-1: Candidate Route Generation

Input: Raw trajectory set T;
Output: New candidate trajectory set T_c.

```
1:  Initialize a stack S;
2:  Split each route r2T into (head,tail) subsequences;
3:  Reconstruct(headSet).
4:  Procedure Reconstruct(Set):
5:  foreach (head,tail) 2 Set do
6:  endFlag = False;
7:  if S is empty or tail.time>S.pop().time then
8:  Push head in S;
9:  Push tail in S;
10: else
11: Push head in S;
12: endFlag = True;
13: if endFlag is False then
14: Reconstruct(tailSet)
15: Insert S in $T_c$;
16: Procedure End
```

TABLE I
RAW TRAJECTORY DATASET

Tid	(head, tail) subsequence	
T_1	p_1(10:00)! p_3(12:00)	p_3(12:00)! p_5(15:30)
	p_5 (15:30)!p_8 (17:30)	p_8 (17:30)!p_{10}(19:00)
T_2	p_2 (10:30)!p_3 (12:30)	p_3 (12:30)!p_4 (17:00)
	p_4 (17:00)!p_5 (19:00)	p_5 (19:00)!p_6 (19:30)
T_3	p_7 (18:30)!p_8 (19:30)	p_8 (19:30)!p_9 (20:00)

For example, the three existing travel routes T_1, T_2 and T_3 can be reinterpreted as (head,tail) pairs, then we have the headSet {p_1; p_2; p_3;p_4; p_5; p_7; p_8}. Starting from p_1, {p_1(10:00) ! p_3(12:00)} is found first. p_3 is the combined point to {p_3 (12:30) !p_4 (17:00)} since the visiting time order is correct. Finally, a can-didate route T_4^0 is generated as {p_1 (10:00) !p_3 (12:30) !p_4 (17:00) !p_5 (19:00) !p_6 (19:30)}. T_1-T_3 are the original routes and T_4^0-$T_6^0$are three of the reconstructed routes.

IV. TRAVEL ROUTES EXPLORATION

With the featured trajectory dataset, our final goal is to recommend a set of travel routes that connect to all or partial user-specific keywords. We first explain the matching function to process the user query. Which is suitable for the travel route recommendation applications, and present the algorithm of the distance-based representative skyline search for the online recommendation system. The Travel Route Exploration procedure is presented as Algorithm-2.

Algorithm-2: Travel Routes Exploration

Input: User u, query range Q, a set of keywords K;

To process the user queries, we first describe how to match query keywords with the characteristic scores assigned to tags. In the offline model, we have built a tag corpus for POIs with characteristic scores and metadata. Given a keyword set K and arbitrary POI p at query time, we define a keyword matching mea-sure KM with the pre-computed information.

1) Greedy Scoring Using Multidimensional Index:

Since computing the optimal representative skyline problem is NP-hard in high dimensional space, a multidimensional index is helpful to efficiently return the results for real-time applications. The DFS-based approach to generate the candidate routes is to enumerate all subsequences. In the procedure of generation, we can simultaneously build an R-tree index while adding each entry into the dataset T_c (at Line 15 of Algorithm 1).

I-greedy is a progressive algorithm that continuously returns 2-approximate guaranteed representative solutions. Instead of retrieving the entire skyline until it is fully computed, I-greedy able to access only a fraction the skyline, which saves a considerable cost. The fundamental of I-greedy is the best-first farthest neighbor search. Specifically, given an MBR M in the R-tree, its max representative distance, max rep dist (M;R), is a value which upper bounds the representative distance of any potential skyline point p in the sub tree of M. Furthermore, to eliminate redundant computations, the greedy algorithm first maintains a conservative skyline based on the intermediate and leaf entries already encountered. Second, it adopts a different access order with fewer empty tests which checks if an arbitrary point is a skyline point.

TABLE II
SUBSET OF CANDIDATE ROUTES

Tid	POI sequence
T1	p1 (10:00)!p3 (12:00)!p5 (15:30)!p8 (17:30)!p10(19:00)
T2	p2 (10:30)!p3 (12:30)!p4 (17:00)!p5 (19:00)!p6 (19:30)
T3	p7 (18:30)!p8 (19:30)!p9 (20:00) ! s ! 6
T^0	p1 (10:00) ! p (12:30) !p5 (17:00) !p (19:00) p (19:30)
T^0	p1 (10:00) ! p (12:00) !p5 (19:00) !p (19:30) s
T^0	p (10:00) p (12:00) p (15:30) p (19:30) p (20:00)

A) Complexity:

Assume that the number of routes in the dataset is N, and the average length of the routes is l. The time complexity of our Travel Route Exploration algorithm depends on three parts: (i) scan the whole database to find the candidate routes in the query range, (ii) calculate feature scores and extract an arbitrary skyline search on all candidate routes, and (iii) derive the representative skyline travel routes. In the case of extensive routes returned from a large-scale query region, it leads to excessive computational time and is not applicable for an interactive online system. We optimize the implementation by parallelizing the score comparison in step (ii), which involves independent computations of each route. For step (iii), when allowed to run continuously.

V. EXPERIMENTS

In this section, we empirically evaluate the effectiveness and efficiency of the proposed algorithms. First, we describe the baseline approaches and evaluation methodology of the experiments. We use two real-world LBSN datasets. The FB dataset is collected by Facebook API.5 We have taken 96 volunteers' Facebook accounts as user seeds (most of the users live in Taiwan) and crawled all their and their friends' location records (i.e., check-ins and geo-tagged photos) over the period of Jan. 2012-Dec. 2014. CA is another four square dataset with an undirected friendship network.

We implemented the system on an x86_64 Linux server with 16 cores and 8 GB memory. All the scores mentioned in Section 3 are computed offline and stored in a Post-greSQL 9.3 database with GIS extension.

1) Keyword Matching Accuracy:

In this section, we evaluate the quality of the extracted key-words. Since our check-in datasets do not have sufficient text descriptions, i.e., tags, we collected an additional photo dataset consisting of 165,057 photos with 958,441 tags. For that, the tags are regarded as input keywords. We used Flickr API to collect photos with photo ID, image, location (lat. and lon.), user ID, photographed time, and textual tags (only if they existed) as attributes. We collected GPS-tagged photos in the same local area, i.e., the Taipei area, amounting to 165,057 photos. Geo-Social Influenced Routes (GSI). Only consider the geo-social influence score. The route consists of POIs visited by geo-social influential users in the social network. Keyword-Aware Skyline Travel Route (KSTR). KSTR out-puts full Skyline routes based on both POI and user factors. Keyword-Aware Representative Travel Route. Our KRTR outputs optimal representative Skyline routes.

VI. RESULTS AND DISCUSSION

3 types of techniques have been used to fetch the relevant information, namely ANN, Text categorization and opinion mining.

- A news channel has been involved to avoid rumors.
- Any people can post the information about the traffic and placed they went.
- All positive and negative information can be shared to the public.
- User can compare the public information with the user information to avoid rumors.
- In combination with micro blogging and opinion mining user will get the most relevant information for their search.
- Information can be shown in charts, graphs and Grid.

VII. CONCLUSION

In this paper, we study the travel route recommendation problem. We have developed a KRTR framework to suggest travel routes with a specific range and a set of user preference keywords. These travel routes are related to all or partial user preference keywords, and are recommended based on (i) the attractiveness of the POIs it passes, (ii) visiting the POIs at their corresponding proper arrival times, and (iii) the routes generated by influential users. We propose a novel keyword extraction module to identify the semantic meaning and match the measurement of routes, and have designed a route reconstruction algorithm to aggregate route segments into travel routes in accordance with query range and time period. We leverage score functions for the three aforementioned features and adapt the representative Skyline search instead of the traditional top-k recommendation system. The experiment results demonstrate that KRTR is able to retrieve travel routes that are interesting for users, and outperforms the baseline algorithms in terms of effectiveness and efficiency. Due to the real-time requirements for online systems, we aim to reduce the computation cost by recording repeated queries and to learn the approximate parameters automatically in the future.

References

[1] Z. Chen, H. T. Shen, X. Zhou, Y. Zheng, and X. Xie, "Searching trajectories by locations: An efficiency study," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2010, pp. 255–266.
[2] H.-P. Hsieh and C.-T. Li, "Mining and planning time-aware routes from check-in data," in Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manage. 2014, pp. 481–490.
[3] V. S. Tseng, E. H.-C. Lu, and C.-H. Huang, "Mining temporal mobile sequential patterns in location-based service environments," in Proc. Int. Conf. Parallel Distrib. Syst., 2007, pp. 1–8.
[4] W. T. Hsu, Y. T. Wen, L. Y. Wei, and W. C. Peng, "Skyline travel routes: Exploring skyline for trip planning," in Proc. IEEE 15th Int. Conf. Mobile Data Manage., 2014, pp. 31–36.
[5] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from GPS trajectories," in Proc. 18th Int. Conf. World Wide Web, 2009, pp. 791–800.
[6] Q. Yuan, G. Cong, and A. Sun, "Graph-based point-of-interest recommendation with geographical and temporal influences," in Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manage. 2014, pp. 659– 668.
[7] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation," in Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2011, pp. 325–334.