LSBNGNN

Chinmaey Shende and Soumyashree Sahu

I. Introduction

Location based social networks have recently gained much attention.As millions of users, social services like Facebook and Twitter have some of the most popular Internet applications, the vast knowledge that all these social networking sites accumulates, it enables impact of mobility and social relationships in between users. Various traditional approaches generate mobility and social features that can help with friendship and location prediction functionalities but these may require tedious human efforts and also difficult to generalize.

Location Based Social Networks (LBSNs), such as Foursquare, have attracted millions of users and generated a considerable amount of digital footprints from their daily life.This can be considered as a data source of studying impact of mobility and social relationships on each other.In this specific type of LSBNs, the users can share their presence at a real time with their friend by checking in at a point of interest(POI), such as any famous attraction or restaurant.

Various previous findings have discussed about correlation between user mobility characteristics and social networking[8][28].Two main approaches have been emphasized in these papers and those are - friendship prediction (or link prediction) and location prediction. Friendship prediction emphasizes on recommending social relationships that are likely to be established in future[28] and location prediction, predicts to which POI a user will go to at a given context(at a certain time). Previous works have illustrated that, considering correlation between user mobility and social relationships can improve the performance of both friendship and location prediction. Generally, all such approaches usually select a set of hand-crafted features from user mobility data or social network and show impact of one on the other.But such manual feature engineering process require tedious effort from experts and also show less generalization.To overcome the limitation of hands-crafted features, automatic feature learning[3] is proposed with application of networks or graphs, known as graph(or network) embedding.

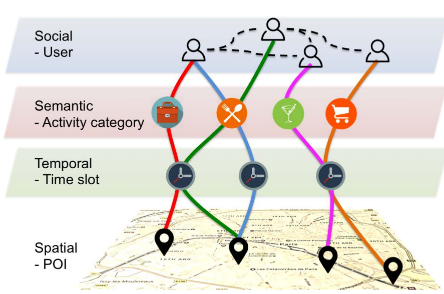

Fig. 1. LBSN-System Diagram

The figure presents LBSN data in form of a hypergraph, that has four key data domains,i.e., spatial,temporal,semantic and social.This graph contains edges(black dotted line- friendships) between two users in the social network, hyper-edges(check-ins) linking four nodes, one from each domain, representing user's presence at a POI at a specific time along with the semantic information representing user's activity at location.

Existing graph embedding techniques can't fully grasp the complex structure of LBSNs[6].Against this background, human mobility and social relationships were revisited in LBSNs and new approach LBSN2Vec is proposed, which is a hypergraph embedding approach designed specifically for LBSN data with including both friendship edges and check-in hyper-edges at same instant.

A. *background*

LBSN2Vec designs hyper-graph embedding approach that involves 3 stages.

1) Data Collection:-
   Large scale data collected over long-time period and analysis is done to reveal correlations between user check-ins and corresponding social network. The publicly available LBSN datasets from Foursquare[12],Gowalla[8] are collected. These contain user check-in data over a period of time and one snapshot of corresponding user social network before and at the end of two-years.
2) LBSN2Vec
   LBSN2Vec, a hypergraph embedding approach is proposed, that efficiently learn node embeddings of friendship edges and check-in hyperedges in a LBSN hypergraph. LBSC2Vec, not only uses social network, check-in data on LBSNs, semantic and temporal information of check-ins , but also address the impact of mobility patterns and social relationships on each other.
3) Evaluation of dataset
   After thorough evaluation on dataset, it's proved that LBSN2Vec outperforms state-of-the-art graph embedding techniques on both friendship and location prediction tasks. The improvement achived in friendship prediction and location prediction is 32.95 percent and 25.32 percent, respectively.

B. *Graph Embeddings*

Existing graph embedding approaches focus on preserving pairwise node proximity in a classical graph, which can be further classified into two categories according to the embedding learning process.

- Factorization based approach: In [6] , the factorization based approach is introduced, that measure pairwise node proximity as a matrix using a certain network proximity metric, such as common neighbor , and then factorize this proximity matrix using matrix factorization techniques to learn the node embeddings.But they may face scalability issues due to quadratic complexity of matrix factorization algorithms.
- Graph sampling approach: Graph-sampling approaches[13] sample node pairs (directly or via random walks) from a graph, and then design specific models to learn node embeddings from the sampled node pairs via stochastic optimization. These graph-sampling based approaches are able to scale up to large datasets, as their complexity mainly depends on the number of the sampled node pairs.

II. Related work

In order to compare and improve up on the approach of calculating the hypergraph embedding following papers were studied.

A. Human Mobility and Social Relationships

1) LBSN2VEC++:
LBSN2VEC++ [2] is improvement on LBSN2VEC. Heterogeneous graph embedding problems have been studied LBSN2VEC. This approaches mainly focuses on capturing the meta-structures of a heterogeneous graph (using meta-paths, for example) and project nodes into a unified embedding space.However, simply projecting nodes into a unified space fails to fully capture the complex structural characteristics of the LBSN heterogeneous hypergraph, resulting in unsatisfied results. LBSN2VEC++ learn node embeddings from both sampled friendship homogeneous edges and check-in heterogeneous hyperedges by not only capturing the n-wise node proximity encoded by the hyperedges, but also considering the embedding space transformation between node domains to fully grasp the structural characteristics of the LBSN heterogeneous hypergraph. LBSN2VEC++ additionally preserves triple-wise proximity of activity node, the transformed user and POI nodes.
2) GNN-FiLM In GNN-FiLM [3], the representation of the target node of an edge is used to compute a transformation that can be applied to all incoming messages, allowing feature wise modulation of the passed information. In GNN-FiLM, the representation of the target node of an edge is used to compute a transformation that can be applied to all incoming messages, allowing feature wise modulation of the passed information.
This is the latest GNN implementation supported with Tensor flow. It receives a graph as an input. It uses neural network based technique to learn and update node embeddings.
3) Hypergraph Convolution and Hypergraph Attention [4] In many real applications, the relationships between objects are in higher-order, beyond a pairwise formulation. To efficiently learn deep embeddings on the high-order graph-structured data, this paper proposes two end-to-end trainable operators to the family of graph neural networks, i.e., hypergraph convolution and hypergraph attention. Whilst hypergraph convolution defines the basic formulation of performing convolution on a hypergraph, hypergraph attention further enhances the capacity of representation learning by leveraging an attention module. This paper could provide directions to possible techniques to handle hypergraphs with GNN.
4) Hypergraph neural networks [5] Hypergraph neural networks (HGNN) [3] framework is for data representation learning, which can encode high-order data correlation in a hypergraph structure. A hyperedge convolution operation is designed to handle the data correlation during representation learning. In this way, traditional hypergraph learning procedure can be conducted using hyperedge convolution operations efficiently. HGNN is able to learn the hidden layer representation considering the high-order data structure, which is a general framework considering the complex data correlations.
5) Hin2Vec [6] HIN2Vec neural network model, designed to capture the rich semantics embedded in HINs by exploiting different types of relationships among nodes. Given a set of relationships specified in forms of meta-paths in an HIN, HIN2Vec carries out multiple prediction training tasks jointly based on a target set of relationships to learn latent vectors of nodes and meta-paths in the HIN. This paper is compared with LBSN2VEC++. This can provide guidance to understand the issues in handling the Hypergraphs with GNN.

III. Project Proposal

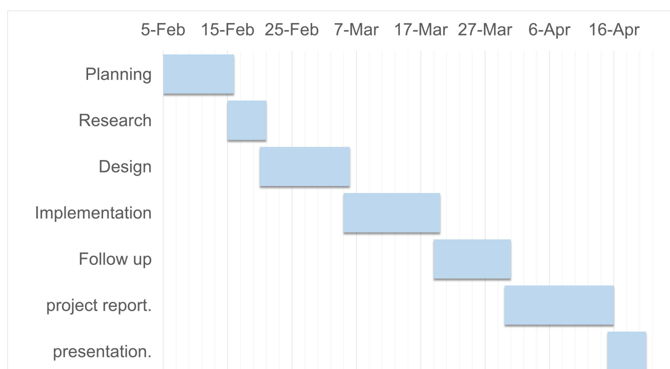

Fig. 2. Plan

A. Responsible contents

1) Introduction + background: Soumya
2) Related work (or survey): Chinmaey and Soumya
3) Proposed directions of technical components: Chinmaey
4) Dataset preparation + plan of experimental studies: Soumya

5) References: Soumya

B. Proposed work

1) LBSN2Vec :
It has considered user check-in and social network snapshots to study impact of friendship and mobility pattern of users on each other. It uses random-walk-with-stay scheme to sample friendship and check-in hyperedges from a LBSN hypergraph, but learn node embeddings without considering the heterogeneous nature of the LBSN hypergraph. Our goal is to reproduce algorithm to learn the hypergraph node embeddings. We plan to reproduce the results using the source code shared by the authors [7].
2) LBSN2Vec++ :
This Uses the same random-walk-with-stay scheme as LBSN2Vec to sample friendship and check-in hyperedges from a LBSN hypergraph, but learn node embeddings considering the heterogeneous nature of the LBSN hypergraph. Specifically, it learns to preserve the n-wise node proximity using the general technique, with involving the embedding space transformation in learning from check-in hyperedges. LBSN2Vec++ significantly and consistently outperforms both state-of-the-art graph embedding techniques. We plan to reproduce the results of the LBSN2Vec in order to compare the results.
3) New approach LBSN2GNN: GNN supports learning the node embeddings effectively. However it would be interesting to use the GNN to learn embeddings with Hypergraphs. Using GNN we will attempt find correlation between friendship and mobility pattern taking user check-ins data and social network snapshot data. As GNN does not specify it's application over hypergraphs we need to understand issues with Hypergraph implementation using Neural network. Also we need to understand existing neural network based implementation HIN2Vec to compare the performance.

IV. Empirical Data Analysis

Data analysis section will include or dataset collection, it's characteristics, followed by empirical analysis on impact of mobility on social network and vice versa.

A. Dataset collection

- Dataset of Foursquare is used which contains global-scale checkins over about two years (from Apr. 2012 to Jan. 2014), [1]
- Dataset of Foursquare is used which contains two snapshots of the corresponding user social network collected before (in Mar. 2012) and after (in May 2014) the check-in data collection period.

To summarize, the dataset contains 22,809,624 check-ins from 114,324 users on 3,820,891 POIs, and two snapshots of the corresponding user social network before (363,704 friendships) and after (607,333 friendships) the check-in data collection period. Fig 3 shows CCDF plot for user check-in data.

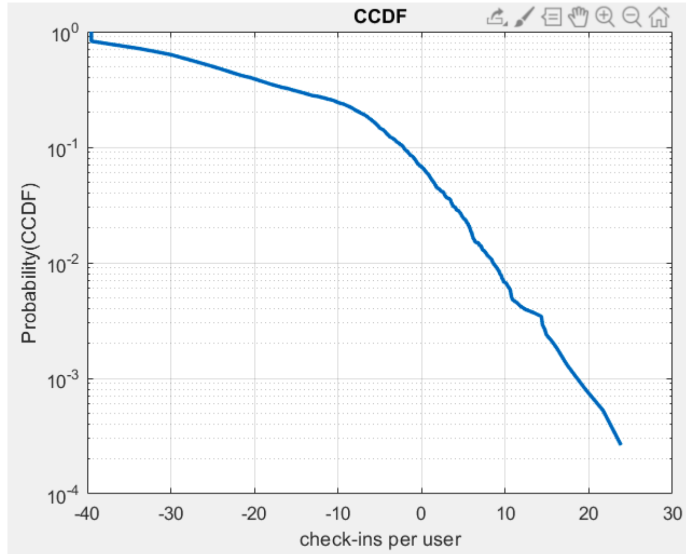

Fig. 3. CCDF Check-ins per user

References

[1] Dataset: https://sites.google.com/site/yangdingqi/home/foursquare-dataset
[2] Dingqi Yang, Bingqing Qu, Jie Yang, and Philippe Cudré-Mauroux "LBSN2Vec++: Heterogeneous Hypergraph Embedding for Location-Based Social Networks"
[3] Marc Brockschmidt "GNN-FiLM: Graph Neural Networks with Feature-wise Linear Modulation."
[4] Song Baia, Feihu Zhanga, Philip H.S. Torra "Hypergraph Convolution and Hypergraph Attention"
[5] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph neural networks," AAAI 2019, 2018.
[6] Tao-yang Fu et. al. "HIN2Vec: Explore Meta-paths in Heterogeneous Information Networks for Representation Learning"
[7] Source code LBSN2VEC https://github.com/eXascaleInfolab/LBSN2Vec