Yelp Matching and Prediction in Bipartite Networks– Milestone

Sidi Chang
schang03@email.wm.edu

April 13, 2016

1 Introduction

For the last five years, social network has become an important part of our daily life. As smart phones become increasingly popularized, Facebook, Google and Uber have changed our behaviors, as well as Yelp. Matching and predicting novel links has many applications in different areas, such as Ads, recommendation system and network analysis [4] [3]. Yelp Business rating prediction challenge [7] in 2013 gave us great datasets to analyze. In this paper, my motivation, prior related work, data collection and aggregation will be explained. Methodology and algorithms for this research plan will be discussed and concluded clearly and thoughtfully.

2 Motivation

My research interest was initially triggered by debating over whether Uber will worse NYC traffic [2]. The debate made me curious about how Uber can affect NYC traffic in different areas and how Uber can recommend drivers to customers in peak hours. After searching and analysis of open source data from Github, the incompleteness of Uber data, lack of open API from Uber was insufficient to finish the challenge. Later, I saw Yelp competition in Kaggle which is similar to the task with Uber Data. Yelp Data Challenge [7] gives me an opportunity to conduct research on this field, evaluating on Link Prediction in different areas and recommendation of businesses to each customer.

3 Prior Related Work

The big data and its techniques is discussed in the work of H.V. Jagadish et al. [3]. In this Review Article, the authors talked about key insights of big data. They states that most researchers only focus on part of creating value from Big Data rather than following all the steps. The methods surveyed show detailed steps on the big data life cycle, from data acquisition to interpretation. The conclusion shows that even we have entered an era of Big data with not only key technical challenges but also social and political challenges that need to be addressed. In my paper, I will follow the steps appeared in this paper and try to avoid social and political challenges such as make business name encrypted.

The visual exploration concerning visualization methodology proposed by Nivan Ferreira's group [1]. In the article, the authors not only focus on methodology of methods and challenges to address, but they also propose a new visual query model that supports complex queries over origin-destination(OD) date. With this implementation, people can access and visualize data much easily. This article also provides us new strategy to generate clutter-free visualization for large results with interactive response times. In my paper, I will try to use visualization tools such as Tableau, D3.js and hold a server through Heroku.

The stock aware recommendation algorithm is a link prediction method proposed by Zhong's group [8]. Unlike the relatively simple methods surveyed in previous paper which only cares about the accuracy of matching and prediction, stock aware recommendation algorithm is a hybrid approach that combines bipartite matching and predicting with inventory size. The algorithm can be considered in two phases, in which recommendations for all users based on both user preference and inventory of different items are made as first step, then a dual method that can reduce the number of variables from n^2 to n is applied to improve significantly computing time complexity. In my paper, I will learn from his methods and figure out some optimization methods to decrease my algorithm complexity.

Previous work for a different Yelp Data Challenge Dataset is proposed as a final project for Stanford CS224W class [4]. Various methods such as preference attachment, Jaccard Coefficient are used in this paper. The authors address this problem in detail and tryed different methodology and algorithms to implement. Then the authors also compare experimental results and conclude their decision based on those results. Since this paper addresses several methodology and algorithm which is central to my work, I will describe it detailed in later section. In my paper, I will implement some of his algorithms and compare with other algorithms and toolbox/functions in Python and MATLAB, then evaluate performance of each methodology.

Lastly, Personalized PageRank is an extension to PageRank proposed by Page, Brin et al. in their original Page Rank algorithm [5]. Whereas the teleport vector is uniform over all nodes in basic PageRank, one can instead choose to specify a different probability distribution over the set of nodes, yielding personalized results for different individuals. In my paper, I will apply the algorithm to do the recommendation matching.

Above all, in my project the difference between my work and other papers has two phases. The first phase is that I would follow some methodology and algorithms of those papers. [3] [1] give me some basic insight of big data procedure. [4] [5] give me methodology and algorithms I can use to make prediction and recommendation. I will combine those ideas and choose the most fit one on my project. The second phase is that I would follow some paper such as [8] and come up with some new idea with recommendation, my recommendation will mainly base on Near-Neighbor Search algorithm.

4 Problem Definition

Part of the problem is a previous challenge called RecSys Challenge 2013 which focuses on Yelp business rating prediction [7]. The source of my dataset is also from there. To transform this into an easily analyzable form, I will first make the following definitions:

- We assume there exists two bipartite sets: set of customers, set of business. Because two sets are bipartite, there does not exist any connection inside each set

- We say that there is an edge between customer i and business j if customer i gave business j a rating before

My problem is specified in two parts: Link prediction on a business rating, and business recommendation for customers.

Link prediction on a business rating. Given a set of businesses B, a set of customers C, I construct the B2C graph $G = (V, E)$ as follow: Let V be the set of $|B| + |C|$. For each customer i and business j, if customer has ever rated business, then edge $value(i, j) = rating($1 Star to 5 Star). For the graph G, we consider the following features: a) The number of customers who have rated to several businesses b) the number of customers who have interacted the same business. The link prediction task is then to predict the rating for each business in the future.

Recommendation Based on the previous graph, we need to make recommendations for each customers with their highest potential businesses they may go.

5 Data Collection and Processing

I collected Yelp Dataset Challenge [7] to serve as base data for my project.

5.1 Dataset Information

The Yelp data contains two main datasets: Training set and Test set. In training set, 11,537 businesses, 8,282 checkin sets, 43,873 users, 229,907 reviews; in test set, 1,205 businesses, 734 checkin sets, 5,105 users, 22,956 reviews.

5.2 Preprocessing

Training data and test data have the same format. The first part of my project is to do the prediction which involves the rating info, the second part is about recommendation which mainly focus on stars, reviews and categories. Dataset is stored in JSON object format. Each object type is in a separate file, one JSON object per line. Four kinds of data can be obtained from the dataset. I will clean some of the data with Python and some of the JSON data will be transformed to TSV file.

- Business

```
{
  ’type’: ’business’,
  ’business_id’: (encrypted business id),
  ’name’: (business name),
  ’neighborhoods’: [(hood names)],
  ’full_address’: (localized address),
  ’city’: (city),
  ’state’: (state),
  ’latitude’: latitude,
  ’longitude’: longitude,
  ’stars’: (star rating, rounded to half-stars),
  ’review_count’: review count,
  ’categories’: [(localized category names)]
  ’open’: True / False (corresponds to permanently closed, not business hours),
}
```

- Review

  ```
  {
    ’type’: ’review’,
    ’business_id’: (encrypted business id),
    ’user_id’: (encrypted user id),
    ’stars’: (star rating),
    ’text’: (review text),
    ’date’: (date, formatted like ’2012-03-14’, %Y-%m-%d in strptime notation),
    ’votes’: {’useful’: (count), ’funny’: (count), ’cool’: (count)}
  }
  ```

- User
  Some user profiles are omitted from the data because they have elected not to have public profiles. Their reviews may still be in the data set if they are still visible on Yelp.

  ```
  {
    ’type’: ’user’,
    ’user_id’: (encrypted user id),
    ’name’: (first name),
    ’review_count’: (review count),
    ’average_stars’: (floating point average, like 4.31),
    ’votes’: {’useful’: (count), ’funny’: (count), ’cool’: (count)}
  }
  ```

- Checkin If there are no checkins for a business, the entire record will be omitted.

  ```
  {
    ’type’: ’checkin’,
    ’business_id’: (encrypted business id),
    ’checkin_info’: {
          ’0-0’: (number of checkins from 00:00 to 01:00 on all Sundays),
          ’1-0’: (number of checkins from 01:00 to 02:00 on all Sundays),
          ...
          ’14-4’: (number of checkins from 14:00 to 15:00 on all Thursdays),
          ...
          ’23-6’: (number of checkins from 23:00 to 00:00 on all Saturdays)
    } # if there was no checkin for an hour-day block it will not be in the dict
  }
  ```

After cleaning the data, I draw the degree distribution of Yelp dataset, majority of the businesses are located in AZ, so I call it AZ Degree distribution and is shown in Figure [1].

6 Methodology and Algorithms for research

As I mentioned before, the paper proposed by Kevin (Junhui) Mao, Xiao Cai and Ya Wang [4] discussed methodology and algorithms clearly and thoughtfully. Insight from their paper, I would use three classes of algorithms in Link prediction and one class of algorithm in Recommendation:

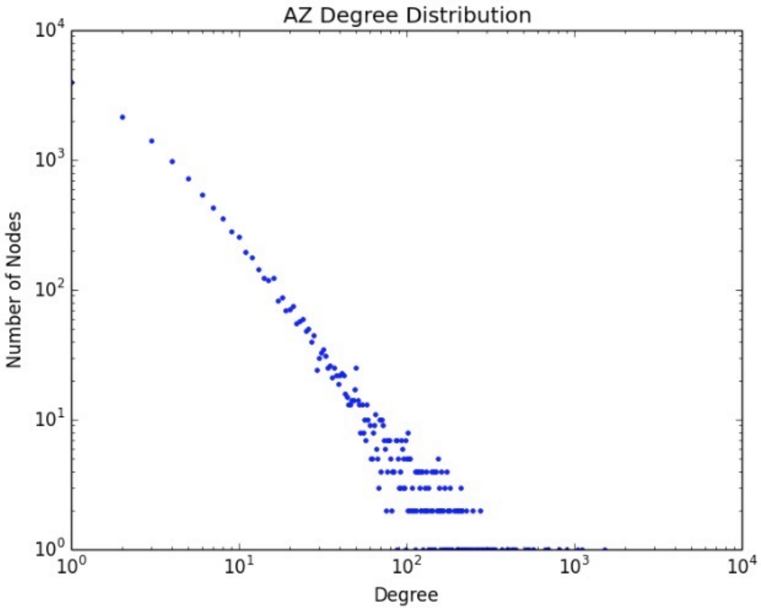

Figure 1: Degree distribution of Businesses in AZ

- scores based on proximity metrics using node neighbor features(Both prediction and recommendation)
- scores based on random network that serves as benchmark

Node Neighborhood Definition In a user-business bipartite network, if two users review the same business, we assume they are similar to each other. The more intersection they have, the closer they are.

- $\mathcal{N}(\mu) :=$ set of users who are co-reviewers of a given user μ for a business b
- $\mathcal{N}(b) :=$ set of users who reviewed business b

Both sets only contain users data which will make algorithm computation easier.
Several existing algorithms will be applied and compared for each candidate new edge, The following algorithms would be considered:

- Random Benchmark
  This algorithm will simply create random edges between users and businesses, the number of users, businesses, total edges will be the same, and it will serve as a benchmark for evaluating other algorithms.
- Jaccard Coefficient [6]

$$score(u, b) = \frac{|\mathcal{N}(\mu) \cap \mathcal{N}(b)|}{|\mathcal{N}(\mu) \cup \mathcal{N}(b)|}$$

Jaccard Coefficient is normalized into $[0, 1]$ and can measure the strength of relationship between two nodes when the existence of common neighbors is simply owing to the existence of lots of neighbors for each node. This methodology can be used in both prediction and recommendation. In prediction part, the rate for one business will be close to another business when users who rate them have a large Jaccard Coefficient. In matching, the system will recommend users to go to businesses where users with large Jaccard Coefficient usually go to.

- Preference Attachment [6]

  $$score(u, b) = |\mathcal{N}(\mu)| \cdot |\mathcal{N}(b)|$$

  Assume the chance that μ will review b is proportional to the number of other users who have already reviewed b, and it's correlated with the number of users who have similar behavior as μ.

- PageRank [5]
  PageRank is first introduced by the founder of Google Larry Page and is used for Google searcher ranking. In my paper, build an adjacency matrix of users with their Jaccard Coefficient, then apply the matrix to the previous place they have been to, get a result and rank them, recommend the top one to users.

6.1 Link Prediction

Three methods will be used to do link prediction: Jaccard Coefficient, Random Benchmark and a method called **minimum-common-neighbors** introduced by Kevin (Junhui) Mao, Xiao Cai and Ya Wang [4]. [4] algorithm is shown as following:

Algorithm 1 Minimum common neighbors

```
Threshold is pre-set and adjusted based on performance, and for each user business pair $(\mu, b)$ :
    (a) if $(\mu, b) \in$ edge: continue
    (b) compute score $(\mu, b)$ : using Jaccard Coefficient or Preference Attachment
    (c) if $|\mathcal{N}(\mu) \cap \mathcal{N}(b)| <$ threshold : continue
    (d) output $(\mu, b, score(\mu, b))$
Sort the $(\mu, b)$ pairs by decreasing order of score $(\mu, b)$
Predict the top pairs with highest score $(\mu, b)$ as new nodes
```

References

[1] Nuno Ferreira, Jorge Poco, Huy T Vo, Juliana Freire, and Cláudio T Silva. Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2149–2158, 2013.

[2] Business Insider. Uber isn't making nyc traffic worse.

[3] HV Jagadish, Johannes Gehrke, Alexandros Labrinidis, Yannis Papakonstantinou, Jignesh M Patel, Raghu Ramakrishnan, and Cyrus Shahabi. Big data and its technical challenges. *Communications of the ACM*, 57(7):86–94, 2014.

[4] Kevin (Junhui) Mao, Xiao Cai, and Ya Wang. Link prediction in bipartite networks - predicting yelp reviews.

[5] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: bringing order to the web. 1999.

[6] Anand Rajaraman, Jeffrey D Ullman, Jeffrey David Ullman, and Jeffrey David Ullman. *Mining of massive datasets*, volume 1. Cambridge University Press Cambridge, 2012.

[7] Yelp. Yelp business rating prediction.

[8] Wenliang Zhong, Rong Jin, Cheng Yang, Xiaowei Yan, Qi Zhang, and Qiang Li. Stock constrained recommendation in tmall. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2287–2296. ACM, 2015.