Competitive Relationship Prediction for Points of Interest: A Neural Graphlet Based Approach – Appendix

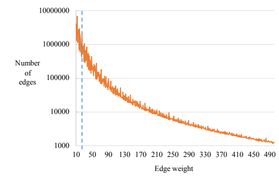
Jingbo Zhou, Tao Huang, Shuangli Li, Renjun Hu, Yanchi Liu,
Yanjie Fu, Member, IEEE, Hui Xiong, Fellow, IEEE

1 Parameter and illustration of co-query POI graph

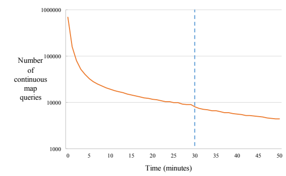
Here we first discuss how to set the edge weight threshold θ_w which controls the tradeoff of quality and quantity of the edges on the graph. In Figure 1(a) we show the distribution of edge weight on Beijing data. After building the co-query POI graph with one-month data, we first remove all the edges whose weight is smaller than 10. The reason is that these edges with low weight take a large portion of the graph (following a power-law distribution), but bring few meaningful information. Furthermore, we compute the average weight of the remending edges and remove all the edges whose weight is smaller than the average. As shown in Figure 1(a), such an average weight of the co-query POI graph in Beijing city is 49.3. In this way, we can get a co-query POI graph with high quality edges.

Second, we set the time interval $\Delta T = 30$ minutes. The ΔT defines how long between two map queries is considered to be in one search session. Figure 1(b) shows the distribution of time intervals between two continuous map queries on Beijing data. We find that 70.5% of the two continuous map queries are within 30 minutes. According to our intuitive experience, it makes sense with a high probability that the map search queries beyond half an hour do not belong to the same search session. Therefore, we set $\Delta T = 30$ in our experiments.

Figure 2 shows an illustration of the co-query POI graph of Starbucks and Windsor KTV with their neighbors. We observe that there are two facts if the weight w_{ij} is high (i.e. many users search p_i and p_j in a short time interval): (i) If p_i and p_j have a similar business (like Starbucks V.S. Line Friends Cafe and Windsor KTV V.S. Recorder KTV), they are likely to be competitive since many users were making a choice decision between them; (ii) If p_i and p_j have a different business (like Starbucks V.S. IFS Mall and Windsor KTV V.S. Peace Cinema), they are likely to be complementary (instead of competitive) since many users were planning a route to visit the two. Thus, we cannot benefit too much for competitive relationship prediction by simply incorporating the weight of the co-query POI graph into a classification model.



(a)



(b)

Fig. 1. (a): Edge weight distribution of the co-query POI graph in Beijing, and the blue line indicates the average edge weight as 49.3. (b):Time interval distribution of two continuous map search queries on Beijing data, and the blue line indicates the quantile of 70.5% cumulative distribution percentage.

Besides, in different regions, the same weight between POIs should reflect a different degree of competition since the density of users, distribution of POIs, and user preference are diverse in different regions. For example, the noisy link weight of the co-query POI graph is high at the city center, and the weight of them is low at a suburban district. It is easy to fall in the local minimum with manually-intensively setting thresholds.

2 Baselines

We compare our model with the following baselines:

- **DIST** and **check-in**. We use the distance to determine the competitive relationship if the POI pair has the same category and their distance is smaller than a threshold. In our experiment, we set the threshold as 4.2 kilometers in Beijing and 4.5 kilometers in Chengdu which have the best accuracy in these cities. We also define a baseline "Check-in" which predicts the relationship as true if the fraction of common check-in users is larger than 5%.

- *Jingbo Zhou, Tao Huang, Shuangli Li, Renjun Hu are with the Baidu Research, Business Intelligence Lab. E-mail: {zhoujingbo, huangtao40, lishuangli, hurenjun01}@baidu.com*
- *Yanchi Liu is with the Rutgers University. E-mail: yanchi.liu@rutgers.edu*
- *Yanjie Fu is with the University of Central Florida. E-mail: yanjie.fu@ucf.edu*
- *Hui Xiong is with the Rutgers University. E-mail: hxiong@rutgers.edu J. Zhou and H. Xiong are the corresponding authors.*

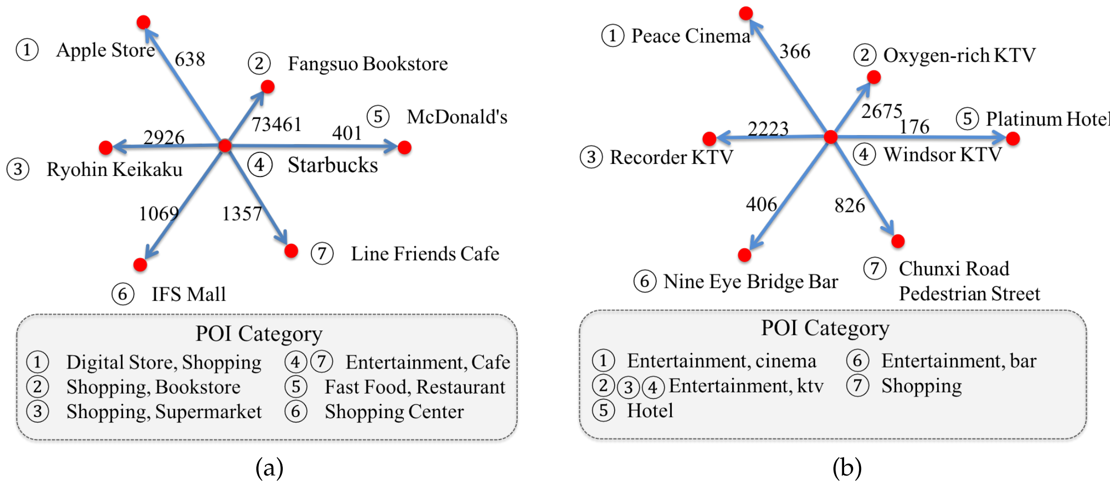


Fig. 2. Illustration of nodes in the co-query POI graph. (a) Starbucks and its neighbors; (b) Windsor KTV and its neighbors.

- **Heuristics methods**. We use a set of heuristics methods on the co-query POI graph as the baselines, including **EW** (graph edge weight), **PA** (preferential attachment [1]), **CN** (common neighbors [2]), **RA** (resource allocation [3]), **JC** (Jaccard [4]). We set a threshold based on the heuristic metrics on the graph to classify the competitive relationship. For example, for **EW**, if the edge weight of a POI pair is larger than a threshold, we think they have a competitive relationship. All other heuristics methods work similarly. We set the best threshold according to their accuracy. Note that since their Precisions, Recalls and F1-measures are too low, we do not show them in Table 3.
- Classification methods. We use the POI feature, region feature and POI-POI feature (edge weight and distance) as the input, and adopt **MLP** (Multilayer Perceptron) and **XGBoost** [5] for competitive relationship prediction.
- **PRA** and **PRA+fea**. PRA (Path Ranking Algorithm [6]) is a scalable algorithm for link prediction. Specifically, **PRA** refers to an MLP model with path feature as input; and **PRA+fea** refers to an MLP model with path feature, POI feature, region feature, and POI-POI feature as input. We generate the path features based on the co-query POI graph and then combine the path feature and other features to an MLP model.
- **Node2vec** and **node2vec+fea**. Node2vec [7] is a state-of-the-art unsupervised method to learn graph node representation. We can also use the node representation of the co-query POI graph as a feature for classification. Similar to PRA, **node2vec** refers to an MLP model with node2vec representation as input; and **node2vec+fea** refers to an MLP model with node representation, POI feature, region feature, and POI-POI feature as input.
- **GNN models** We compared NGP with several state-of-the-art graph neural network models for link prediction which are GNN-SEAL [8], Geom-GCN [9] and GIN [10]. We build the model on the co-query POI graph, and then treat competitive relationship prediction as a link prediction problem.

References

[1] M. E. Newman, "Clustering and preferential attachment in growing networks," *Physical review E*, vol. 64, no. 2, p. 025102, 2001.
[2] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American society for information science and technology*, vol. 58, no. 7, pp. 1019–1031, 2007.
[3] T. Zhou, L. Lü, and Y.-C. Zhang, "Predicting missing links via local information," *The European Physical Journal B*, vol. 71, no. 4, pp. 623–630, 2009.
[4] G. Salton and M. J. McGill, "Introduction to modern information retrieval," 1986.
[5] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *KDD*. ACM, 2016, pp. 785–794.
[6] M. Gardner and T. Mitchell, "Efficient and expressive knowledge base completion using subgraph feature extraction," in *EMNLP*, 2015, pp. 1488–1498.
[7] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *KDD*. ACM, 2016, pp. 855–864.
[8] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," in *NIPS*. ACM, 2018.
[9] H. Pei, B. Wei, K. C.-C. Chang, Y. Lei, and B. Yang, "Geom-gcn: Geometric graph convolutional networks," in *ICLR*, 2020.
[10] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *ICLR*, 2019.