Link Prediction via Second Order Cone Programming

Shreya Malani
BingAds Applied Research
Microsoft, India
shreyam@microsoft.com

Dinesh Dileep Gaurav
BingAds Applied Research
Microsoft, India
digaurav@microsoft.com

Rahul Agrawal
BingAds Applied Research
Microsoft, India
rahulagr@microsoft.com

ACM Reference Format:
Shreya Malani, Dinesh Dileep Gaurav, and Rahul Agrawal. 2019. Link Prediction via Second Order Cone Programming. In *11th ACM Conference on Web Science (WebSci '19), June 30-July 3, 2019, Boston, MA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3292522.3326053

1 INTRODUCTION

Link Prediction in social Networks (e.g. Facebook, Twitter) is a widely studied research area in the quest to understand user relationships. Most networks rarely seek explicit positive/neutral feedback from a user on their relations and even more rarely do they ask for stating a negative vote/sentiment. However, predicting this relation is interesting from the network's perspective, as it is useful in various social network analysis tasks of commercial interest like Community Detection, Node Ranking, Node classification, Recommendation and many more [5, 7]. Positive links are far more easier to obtain compared to negative links (e.g. friends in Facebook). In most of these social graphs, we also have access to user interaction data which is very rich in information on links, for e.g., the posts and comments from Facebook, or tweets in Twitter. In [5] and other contemporary works, these additional information is leveraged for link prediction. For more recent results in link prediction, we refer the interested reader to [6, 8].

In this paper, we formulate the link prediction as a learning problem based on the features of links closely following the lines of [5] and reference therein. Though the proposed approaches in [5] and related papers show promise, they inherently suffer from the computational complexity of the learning frameworks they use. We follow a classification-after-clustering approach wherein we cluster the classes individually before applying a learning framework. We design maximum margin classifiers on this clustered data which can be then converted into standard Second Order Cone programs.

2 PROPOSED LEARNING APPROACH

Our proposed approach is combination of clustering and a learning framework. Initially, we will cluster each link type into different clusters (feature vector generation is explained later) using an off-the-shelf clustering approach. For each cluster, we will estimate the cluster center and the covariance matrix given as $(\bar{\mathbf{x}_i}, \Sigma_i)$. For our classification, we will use a two-class classification model. Thus, clusters belonging to class 1 will have label 1 and -1 for the other. Let $y_i \in \{1, -1\}$ denote the label of cluster i.

WebSci '19, June 30-July 3, 2019, Boston, MA, USA

ACM ISBN 978-1-4503-6202-3/19/06.
https://doi.org/10.1145/3292522.3326053

In first formulation, we propose to fit an ellipse to each cluster i whose center and covariance matrix are given as $(\bar{\mathbf{x}_i}, \Sigma_i)$. In first formulation, we propose to fit an ellipse to each mixture i whose center, variance are given as $(\bar{\mathbf{x}_i}, \Sigma_i)$. In addition, we define a user-defined parameter γ_i which controls the radius of ellipse fitting cluster i. This allows us to consider an optimization problem where we try to find a hyperplane which accurately separates the ellipses corresponding to clusters of both the classes. Let $\mathcal{E}(\bar{\mathbf{x}_i}, \Sigma_\mathbf{i}, \gamma_i)$ denote the ellipse corresponding to cluster i. Our approach is similar to standard soft-margin SVM formulation, where individual points in each class are supposed to be separated by the optimal hyperplane. Similarly, we require the ellipses in same class to lie on the same side of the maximum margin hyperplane and also the points in them. This can be formulated as

$$\begin{aligned} \min_{\mathbf{w}, b, \epsilon} \; & ||\mathbf{w}||_2^2 + \sum_{i=1}^{K} \epsilon_i \\ \text{s.t. } & y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \epsilon_i \,, \; \forall \mathbf{x}_i \in \mathcal{E}(\bar{\mathbf{x}_i}, \Sigma_\mathbf{i}, \gamma_i) \\ & \epsilon_i \geq 0 \; i \in \{1, \ldots, K\} \end{aligned} \tag{1}$$

Consider each ellipse constraint in (1) which should be satisfied for every point $\mathbf{x}_i$ in the ellipse. It is not hard to prove that each elliptical constraint of (1) is equivalent to the Second Order Cone constraint [4]

$$y_i(\mathbf{w}^T \bar{\mathbf{x}}_i + b) \geq 1 - \epsilon_i + \gamma_i ||\Sigma_i^{\frac{1}{2}} \mathbf{w}||_2 \tag{2}$$

Substituting (2) into (1), we have the following Second-Order Cone Program (SOCP)

$$\begin{aligned} \min_{\mathbf{w}, b, \epsilon} \; & ||\mathbf{w}||_2^2 + \sum_{i=1}^{K} \epsilon_i \\ \text{s.t. } & y_i(\mathbf{w}^T \bar{\mathbf{x}}_i + b) \geq 1 - \epsilon_i + \gamma_i ||\Sigma_i^{\frac{1}{2}} \mathbf{w}||_2 \,, \; \forall i \\ & \epsilon_i \geq 0 \; i \in \{1, \ldots, K\} \end{aligned} \tag{3}$$

We note that the complexity of above formulation is in the order of $O(K^6)$, which is the number of clusters (typically within hundreds). Though, this is high, the reduction in computation compared to using the entire set of data points can be useful in most situations. It is noteworthy that in earlier works [3, 5], the sample size is a determining factor in the complexity of the proposed approach. However, as the size of the social graph increases, this can be a limiting factor. However, clustering prior to the classification, significantly reduces the computational complexity while still retaining a comparable performance. Any standard convex package for Cone Programs can be used for solving the optimization problem in (3). We forsake the algebraic conversion of (3) into standard SOCP format as this is widely known[4]. In our second formulation, we set $\gamma_i = 0$ which is equivalent to reducing the elliptical radius to zero. It is straightforward to see that (3) now reduces to

a standard SVM with centroids as the data points. This also further reduces the complexity to $O(K^2)$, possibly at cost of accuracy in classification. We also note that determining the clusters for a given dataset is a well studied problem. For instance, a straight forward approach would be to do a nearest neighbor clustering [4] which will give a rough estimate for K. For more accuracy, we can also use more sophisticated approaches as in [2].

3 EXPERIMENTS

3.1 Methodology

We experimented our proposed framework on the publicly available Epinions graph which is well studied in previous literature as well[3, 5]. Epinions network allows users to explicitly state their trust (positive links) or distrust (negative links) on other users. We refer the interested reader to [3, 5] for a brief overview and a detailed study of the statistical nature of Epinions graph. Missing links correspond to the nonexisting edges. However, negative links are extremely scarce compared to positive and missing links. To alleviate this, we follow algorithm 1 in [5] to construct the negative links based on additional content centric information (available as comments and ratings in Epinions). Positive links are directly taken from the Epinions graph. We follow the lines of [5] and references therein for feature extraction from links. As explained earlier, we constructed two class experiments held across different combinations of missing links, negative links and positive links. For instance, in one of the experiments, negative links will be considered as the first class whereas positive links were considered the second, and vice-versa. Once such a class separation is decided, we constructed clusters on each class (using their feature vectors) using a standard implementation of Gaussian Mixture Models. We note that one can use more advanced clustering mechanisms like BIRCH which scale really well to massive data sets. Once the centroids and the covariances are estimated, we use it as input to our proposed models in (3). We used the convex package CVX [1] to solve the cone programs. We rerun our experiments for different combinations of clusters on each classes as well as different class separations. We choose to report only relevant results due to brevity of space.

3.2 Results and Discussions

In Table 1, we present the performances of our proposed models for each class separation. *Neg*, *Pos*, *Miss* denotes negative, positive and missing links respectively. First column contains the class type in the named order. Second column states the respective number of clusters in colon separated form. We experimented with multiple combinations of number of clusters on each class type and report the precision and recall for predicting class 1. We choose to report only selected combinations of clusters with acceptable performances. We observe that the proposed model has varied performance across different classification scenarios. Further it seems that the performance is superior when class 1 is the larger class (negative links are scarce). However, performance achieved when predicting negative links against positive links is acceptable in comparison to models proposed in [3, 5]. However, the performance was not satisfactory when the comparison was between negative link and missing links. Intuitively, this can be ascribed to the scarcity of negative links. We note that missing links can exist between any two chosen nodes in the graph, however negative links and positive links exist only in a neighborhood depending on the relationship/interactions between nodes[1] in social graphs. This rather makes missing links arbitrary and difficult to model via clusters. Thus, further experiments were conducted restricting the neighborhood of a link to within 4 hops. This numbers are also reported in Table 1 which seems to improve the precision and recall. We observe that the performance reported here is comparable to earlier works [3, 5] (skipped due to brevity of space). We remark that this performance comes at a significant reduction of complexity due to the clustering-before-classification approach.

Table 1: Performance of Proposed Model

Type	Clusters	Precision	Recall
Neg vs *Pos*	10:3	0.304	0.478
Pos vs *Neg*	50:60	0.873	0.91
Neg vs *Pos* within 4 Hops	5:70	0.202	0.5731
Pos vs *Neg* within 4 Hops	10:3	0.950	0.8123
Neg vs *Miss*	30:30	0.107	0.258
Neg vs *Miss* within 4 Hops	50:60	0.112	0.235
Miss vs *Neg* within 4 Hops	10:2	0.909	0.899

3.3 Future Work

An interesting direction is to pursue and evaluate more advanced clustering techniques as this does seem to affect the performance. Further, though the proposed model scales really well, it is impacted by the large number of missing links which is on the order of billions. Thus, it will be interesting to limit all missing links within a neighborhood of the given node whose relationships with others has to be studied.

REFERENCES

[1] M Andersen, Joachim Dahl, and Lieven Vandenberghe. 2013. CVXOPT: A Python package for convex optimization. *abel. ee. ucla. edu/cvxopt* (2013).

[2] Mario A. T. Figueiredo and Anil K. Jain. 2002. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 3 (2002), 381–396.

[3] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. 2010. Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*. ACM, 641–650.

[4] Pannagadatta K Shivaswamy, Chiranjib Bhattacharyya, and Alexander J Smola. 2006. Second order cone programming approaches for handling missing and uncertain data. *Journal of Machine Learning Research* 7, Jul (2006), 1283–1314.

[5] Jiliang Tang, Shiyu Chang, Charu Aggarwal, and Huan Liu. 2015. Negative link prediction in social media. In *Proceedings of the eighth ACM international conference on web search and data mining*. ACM, 87–96.

[6] Jiliang Tang, Yi Chang, Charu Aggarwal, and Huan Liu. 2015. A survey of signed network mining in social media. *arXiv preprint arXiv:1511.07569* (2015).

[7] Jiliang Tang, Yi Chang, Charu Aggarwal, and Huan Liu. 2016. A survey of signed network mining in social media. *ACM Computing Surveys (CSUR)* 49, 3 (2016), 42.

[8] Peng Wang, BaoWen Xu, YuRong Wu, and XiaoYu Zhou. 2015. Link prediction in social networks: the state-of-the-art. *Science China Information Sciences* 58, 1 (2015), 1–38.