Online Community Link Prediction with Collective Classification

Ryan Compton
rcompton@ucsc.edu

Abstract

Research in online communities hasn't made much progress in understanding the mechanics of online community ecologies. Previous link prediction work within social networks has been focused on network structural information without taking into account the relational features that communities can share. In this experiment, collective classification is used to predict future links within online communities which found that a collective classification model can outperform previous structural based algorithms for link prediction. Member and tool features are also compared in ability to predict future links, where member features are found to be the dominant feature in this domain.

Introduction

Online communities are ubiquitous in personal life and business. Usenet, a worldwide distributed internet discussion system, had over 160,000 newsgroups active in 2006. Yahoo has claimed to host over a million online groups of users. Raverly.com, a community for knitting and crocheting hobbyists, stated they had more than 400,000 members in July, 2009 [1]. Online communities are a critical resource for many functions including answering questions, providing support, and developing new social connections.

Communities are not limited to the information and interactions that exist within themselves. Communities can form an ecological network between other communities that share common themes and practices. These connections are formed by the members of a community through linking [2]. The ability of a member to link to another community enriches the complexity of the information present within community by expanding it with external sources. However, the fundamental mechanism of community ecology remains unknown. The ability to define the likelihood of a community to community interaction would provide a mechanism to understand and predict the connection communities form.

Attempts have been made to use structural information of a social network in order to predict the future links that will occur, but the developed models were found to be weak[4]. Since these structural models were poor at link prediction, the link prediction problem may require more network relational based information. One type of classification model that are specific to network based classification is Collective Classification [6]. This method differs from traditional classification techniques in that it does not assume that labels and features are independently, identically distributed. Collective Classification uses the different types of correlations that can exist within the observed and unobserved data. The types of correlations are those that can be between the labels and the observed attributes, the labels and the observed attributes of the surrounding neighbors in the network, and the labels and the unobserved labels of the surrounding neighbors. Due to the dynamic and cohesive nature of online communities, incorporating these types of correlations seems necessary.

Task

The task of this experiment will be to create a collective classification model in which community linking could be predicted. This will enhance the existing means of link prediction within social networks giving insight into what drives a link across communities. This can be used to predict future community networks as well as intelligently intervene with that network to create connections that wouldn’t be formed otherwise.

Further work will be to examine the specific features used in the model and their ability to predict links. Each feature will be compared to a baseline, giving insight into the types of data that is significant in community connectedness. In this experiment, member features and tool features will be examined. Member features are the observed attributes specific to the members of the community. Tool features are the attributes to the types of tools that a community has access to.

Methods

Data

The network examined consists of 127 industrial online communities and 1,629 links between them. This network information was collected through crawling text within the communities to identify hyperlinks. For each link, the following information was captured:

- Source Community: Where it was posted
- Source Tool: Which tool was the link posted in

- Target Location: Internal or external to Source Community
- Target Tool: What type of tool is the link pointing to
- Date: Time stamp when link was posted
- Author ID: Used to identify which author posted the link
- Author Role: A member or an owner of the community they posted in

Community information was also gathered, but was limited to examining the types of tools that were being used, as well as how many tool posts there were. The types of tools that can exist within these communities are: Wikis, Forums, Blogs, Bookmarks and other various specific tools to the company they are residing within.

Labels

This experiment will use only two labels for classification: whether a pair of communities share a link or not. Each possible pair of communities will have to be examined in order to see if a link will be predicted or not, thus running time for the experiment is highly dependent on the size of the network. In the online community domain it is also possible for a community to link to itself, so self-links will also be included in the experiment.

Features

Features were also constructed in order to create relational features between the communities, as well as more descriptive features.

- Relational Features:
  - Shared Authors: Author is present within two communities
  - Tool Diversity: A comparison of how similar two communities are with their tool use
- Descriptive Features:
  - Member Diversity: Average number of communities that members are active in
  - Active Tools: Number of tools used within a community

These constructed features were the main features of interest, but total number of posts was also examined as preliminary data analysis observed that links are correlated with the amount of content present within a community.

Data Splitting

In this experiment, the data set will have a 75-25 split where 75% of the data will be used as a training data set, and then the remaining 25% will be used as a test set. This split is based on date information for the links. The training set will have links that were within the range of April 2009 to June 2012, while the test set will be the range July 2012 to April 2013. This will test if past network information can be used to predict future links. This type of data split was made to conduct a similar split done by Liben - Nowell and Kleinberg [4].

Implementation

Probabilistic Soft Logic (PSL) was used to create the model. PSL is a framework specifically for probabilistic reasoning. It was chosen because it can incorporate the relational and structural features of the community ecology into inferential rules to predict links. Each rule is a first order logic rule. PSL will learn the rules and then learn the weights of each rule. In order to learn the weights, Lazy Max Likelihood was used [3]. Figures 1 and 2 show an example of a rule set used in the experiment.

$$(\ HasPostFrom(C1,\ A,\ N)\ \&\ HasPostFrom(C2,\ A,\ M)\ )\ >>\ HasLink(C1,\ C2)$$

Figure 1.

$$(\ HasLink(C1,\ C2)\ \&\ HasPostFrom(C2,\ A,\ N)\ \&\ HasPostFrom(C3,\ A,\ M)\ )\ >>\ HasLink(C2,\ C3)$$

Figure 2.

Capital letters within the rules represent a random variable. Within this rule set, variables beginning with a C will indicate a community ID. Each rule is set up as an implication. Within figure 1, if community C1 has N posts from author A and C2 has M posts from author A, then communities C1 and C2 will have a link. This rule corresponds with the hypothesis that an active member that is within two communities will link across the two.

Figure 2 shows a collective classification rule. This rule is similar to the rule shown in figure 1, but it has the predicate HasLink within the head. This makes new links dependent on the observed and predicted links made throughout the network. Rules such as these were created for each feature using their associated predicates[1].

Baseline

Previous work has shown that common neighbors is a fair predictor of links when compared to more advanced neighborhood and path length algorithms[4]. This will be used as the baseline of performance for this experiment. Common neighbors is a simple algorithm, so it is easy to set up and run on large networks. It produces a ranked list of nodes where the highest ranked nodes are the most likely to form a link. The rank is measured based on the intersection of two communities’ neighborhoods. Hence, it is

[1] Rules shown in appendix

examining which neighbors are in common between two nodes. The idea behind using this as a link predictor is that nodes that are close in proximity in the network are more likely to form a link in the future than those that are far away. This will be implemented within PSL as well.

$$\Gamma(C) \ = set\ of\ neighbors\ for\ community\ C$$
$$CommonNeighbros(C1, C2) \ = |\,\Gamma(C1) \cap \Gamma(C2)\,|$$

Figure 3. Common Neighbor Algorithm.

Evaluation

Two types of evaluation metrics will be used. Area under ROC curve will be used to examine overall model performance in the task. The ROC curve is a good metric to use for comparing classifiers across different models [5]. It’s a combination of True Positive Rate and False Positive Rate in one single measure that is easy to interpret.

$$True\ Positive\ Rate \ = \frac{(True\ Positives) Positives\ Correctly\ Classified}{Total\ Positives}$$

$$False\ Positive\ Rate \ = \frac{(False\ Negatives)\ Negatives\ Incorrectly\ Classified}{Total\ Negatives}$$

Figure 4. True Positive Rate and False Positive Rate Formulas.

The second metric used is Area under positive class PR curve. Instead of True Positive and False Positive Rate, this will use precision and recall. Recall is equivalent to True Positive Rate, but precision does not take into account true negatives as false positive rate does.

$$Precision \ = \frac{(True\ Positives)\ Positives\ Correctly\ Classified}{Positives\ (True)\ Correctly\ and\ (False)\ Incorrectly\ Classified}$$

The PR curve will be used to compare the different features ability to predict the positive class, which in this experiment is the case of two communities having a link between one another. Due to the large amount of non-existing links within the data set (14,500 non-existing links), positive class needs to be examined since a model can still be fairly accurate even when only predicting no links will be made in the future and using the PR curve is a better alternative when the labels are so skewed [9].

Results

	Area under ROC curve
Common Neighbors Baseline	0.6377

Collective Classification Model	0.7019

Figure 5. Main comparison of collective classification and baseline models.

The incorporation of relational features and collective classification was able to increase the collective classification model's performance above that of the baseline. The rules that were found to be best for classification were those of the member features (Shared Authors and Member Diversity) and total number of posts[2]. The tool features tool diversity and active tools were found to only decrease the model's performance and hence were removed. Collective Classification rules for both member features were also found to improve the model's performance, however total number of posts did not improve performance when incorporated into such a rule. Since the baseline is a fair predictor of links, it too was incorporated into the model, as well as its own collective classification rule. Inclusion of these rules boosted the model to the 0.7 AUC range.

Having the various features added individually to the model gave some expectations as how they would be as positive class predictors and to examine the effects of each feature on the model, area under the positive-class PR curve was examined. Each feature had a model created solely using the rules that were including itself.

	Area under positive-class PR curve
Baseline	0.1603
Shared Authors	0.4344
Member Diversity	0.0223
Member Features	0.0997
Member Features (Including Common Neighbors)	0.1256
Member Features (Collective Classification)	0.1728
Tool Diversity	0.0190
Active Tools	0.0064
Tool Features	0.0064

[2] Model output shown within appendix

Tool Features (Including Common Neighbors)	0.0281
Tool Features (Collective Classification)	0.0320
Overall Collective Classification Model*	0.1759

Figure 6. Feature Specific performance comparisons.
*Overall Collective Classification Model is same as the Collective Classification model in Figure 5.

The majority of features performed below 0.1, indicating that they were negative class predicting models since their value was below random (0.5) [5]. This wasn’t a surprise, as previously stated, this network is very sparse and only contains around 10% of the possible links that could exist. If a model was judged on accuracy alone, it would appear to be doing well as predictors as they were predicting no existing links more often than existing links. When compared to the baseline, only the models of Shared Authors and the collective classification of all member features were able to perform better. Member diversity was surprising to have such a low value as it was found to improve the overall model in the main experiment. This may be due to the overall model needing more rules predicting the negative class more accurately.

The tool features were shown to perform very poorly as positive class predictors. This combined with the results of their decrease in performance of the overall model lead to the conclusion that tool features were poor features to use for link prediction.

Common Neighbors was incorporated into each of the two main feature set models, Member and Tool. This addition improved both models’ ability in positive class prediction. Incorporating the collective classification rule set into each main feature model also made improvements.

Conclusion

It was found that a classifier using collective classification methods was able to outperform the baseline in online community link prediction. While the model still needs to be further explored in order to become a good predictor, this showed a step in the right direction for link prediction methods in the social network domain. Further more, member based attributes were shown to be the stronger features for such a model. Tool features were shown to be poor features to incorporate as descriptive or relational features for this type of link prediction.

Out of all the features examined, Shared Authors was found to the best performer in predicting links. This is likely because two communities are only going to be linked together by an active member and a member that is active within both communities is more likely to spread the information of one to another.

Future Work

- The size of the network examined is rather low compared to previous work [4]. A new data set to examine is needed with a higher amount of nodes. A large set of edges would also be useful as the disparity between the two classes (link vs no-link) was very large in this network.
- New features are required to be constructed. More information can be extracted from the communities, such as the text that surrounds the links being posted. With this text, a content analysis can be conducted and content relational features can be used for a stronger collective classification model.
- Due to the positive results of having common neighbors added to the models, more network structural features could be incorporated. Such features are path based metrics: PageRank/SimRank [7] or Katz [8].

Bibliography:

[1] Resnick, P. & Kraut. R. “Evidence-based social design: Introduction.” *Evidence-based social design: Mining the social sciences to build online communities*. Cambridge, MA: MIT Press.

[2] D. Gibson, J. Kleinberg, P. Raghavan. Inferring Web communities from link topology. Proc. 9th ACM Conference on Hypertext and Hypermedia, 1998.

[3] Kimmig, A., Bach, S., Broecheler, M., Huang, B., & Getoor, L. (2012). A short introduction to probabilistic soft logic. In *Proceedings of the NIPS Workshop on Probabilistic Programming: Foundations and Applications* (pp. 1-4).

[4] Liben - Nowell, David, and Jon Kleinberg. "The link - prediction problem for social networks." *Journal of the American society for information science and technology* 58.7 (2007): 1019-1031.

[5]Fawcett, Tom. "An introduction to ROC analysis." *Pattern recognition letters* 27.8 (2006): 861-874.

[6]Sen, Prithviraj, et al. "Collective classification in network data." *AI magazine* 29.3 (2008): 93.

[7] Glen Jeh and Jennifer Widom. SimRank: A measure of structural-context similarity. In

Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 2002

[8] Leo Katz. A new status index derived from sociometric analysis. Psychometrika, 18(1):39–43, March 1953.

[9] Davis, Jesse, and Mark Goadrich. "The relationship between Precision-Recall and ROC curves." *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006.

Appendix:

Rules used for PSL model

```
( HASPOSTFROM(C1, A, N) & HASPOSTFROM(C2, A, M) ) >> HASLINK(C1, C2)
( ( HASLINK(C1, C2) & HASPOSTFROM(C2, A, N) ) & HASPOSTFROM(C3, A, M) ) >> HASLINK(C1, C3)
COMMONNEIGHBORS(C1, C2, X) >> HASLINK(C1, C2)
( HASLINK(C1, C2) & COMMONNEIGHBORS(C2, C3, X) ) >> HASLINK(C1, C3)
( MEMBERDIVERSITY(C1, Y) & MEMBERDIVERSITY(C2, Z) ) >> HASLINK(C1, C2)
( HASLINK(C1, C2) & MEMBERDIVERSITY(C2, Y) & MEMBERDIVERSITY(C3, Z) ) >> HASLINK(C1, C3)
( POSTTOTAL(C1, B) & POSTTOTAL(C2, D) ) >> HASLINK(C1, C2)
( HASLINK(C1, C2) & POSTTOTAL(C2, Y) & POSTTOTAL(C3, Z) ) >> HASLINK(C1, C3)
( TOOLDIVERSITY(C1, E) & TOOLDIVERSITY(C2, F) ) >> HASLINK(C1, C2)
( HASLINK(C1, C2) & TOOLDIVERSITY(C2, Y) & TOOLDIVERSITY(C3, Z) ) >> HASLINK(C1, C3)
( ACTIVETOOLS(C1, G) & ACTIVETOOLS(C2, H) ) >> HASLINK(C1, C2)
( HASLINK(C1, C2) & ACTIVETOOLS(C2, Y) & ACTIVETOOLS(C3, Z) ) >> HASLINK(C1, C3)
~( HASLINK(C1, C2) )
```

Model output for best performing model:

```
{2.6216731956138024} ( HASPOSTFROM(C1, A, N) & HASPOSTFROM(C2, A, M) ) >> HASLINK(C1, C2)
{squared}
{1.0371928672377855} ( ( HASLINK(C1, C2) & HASPOSTFROM(C2, A, N) ) & HASPOSTFROM(C3, A, M)
) >> HASLINK(C1, C3) {squared}
{2.045236978095972} COMMONNEIGHBORS(C1, C2, X) >> HASLINK(C1, C2) {squared}
{2.1559527550848485} ( COMMONNEIGHBORS(C1, C2, X) & COMMONNEIGHBORS(C2, C3, X) ) >>
HASLINK(C1, C3) {squared}
{1.0896814262317869} ( HASLINK(C1, C2) & COMMONNEIGHBORS(C2, C3, X) ) >> HASLINK(C1, C3)
{squared}
{4.039950485267633} ( MEMBERDIVERSITY(C1, Y) & MEMBERDIVERSITY(C2, Z) ) >> HASLINK(C1, C2)
{squared}
{1.1639675161717375} ( ( HASLINK(C1, C2) & MEMBERDIVERSITY(C2, F) ) & MEMBERDIVERSITY(C3,
F) ) >> HASLINK(C1, C3) {squared}
```

```
{3.8915581527270837} ( POSTTOTAL(C1, C) & POSTTOTAL(C2, D) ) >> HASLINK(C1, C2) {squared}
{3.7949065667295865} ~( HASLINK(C1, C2) ) {squared}
```