Replication of : Exploiting Social Network Structure for Person-to-Person Sentiment Analysis

Tristan Labetoulle
College of Computing
Georgia Institute of Technology
Atlanta, GA 30308
`ltristan3@gatech.edu`

Matthew King
College of Computing
Georgia Institute of Technology
Atlanta, GA 30308
`matthew.king@gtri.gatech.edu`

1 Introduction

Developing a strong model to make predictions about the relationships between people can be utilized over an expansive application space. With all the data available today, such as on Social platforms, this work is important to help improve the analysis of Social Networks by harnessing the features that are available (inferred from conversations between friends on Social platforms for example). Former research shows that sign-network analysis (Cartwright and Harary, 1956) and sentiment analysis (Pang and Lee, 2008) can be used alone to predict the relationship between people. In the paper (West, 2014) replicated by this work, the authors demonstrated the advantages of combining both sign-network analysis and sentiment analysis to develop a robust relationship prediction model. The objective of this paper replication was to understand the methods used by the original authors to develop a prediction model using the same datasets in order to achieve the same results. In addition, we show the trade-offs between the weight values chosen to tune the prediction model. The replication code is available at `https://github.com/tristanlabetoulle/replication-project`

2 Data Sources

2.1 Data Acquisition

The datasets used in the study were extracted from the following websites :

- Wikipedia Requests for Adminship corpus (RfA Dataset) : `http://snap.stanford.edu/data/wiki-RfA.html`
- The Convote corpus of Congressional speeches (Convote Dataset) : `http://www.cs.cornell.edu/home/llee/data/convote.html`

These datasets were chosen by the original authors. The first dataset has strong textual data that can be captured through sentiment analysis, but a poor network structure while the opposite is true for the second one.

2.2 Data Preprocessing

2.2.1 The RfA Dataset

The dataset is composed of 11199 nodes, 172062 edges and 1,157,656 triangles. The directed graph was built using the NetworkX library, removing edges where some information was missing (missing source, target, vote or comment). We made sure two nodes had at most one edge to simplify the study of the graph. We finally removed neutral votes to keep only upvotes (noted as 1) and downvotes (noted as 0). We then created 10 sets to average the error. We chose 10 nodes randomly in the graph and applied a BFS search to extract 10 subgraphs of 200 nodes (originally 350 in the paper but we lowered this number as the computational time became too long otherwise). We then removed overlapping edges between the subgraphs so that they have no common nodes and edges. Since the graph has many edges, the number of overlapping edges would be low in number. We take the first subgraph and compute the overlapping edges with all the nine other subgraphs pairwise and remove those edges to the nine subgraphs, we then take the second subgraph and do the same process with the 8 following graphs. We do this until we reach the tenth graph. The resulting subgraphs showed subsets of people voting for others. The subgraph edges were composed of vote data and the comment about the vote as shown in Figure 1.

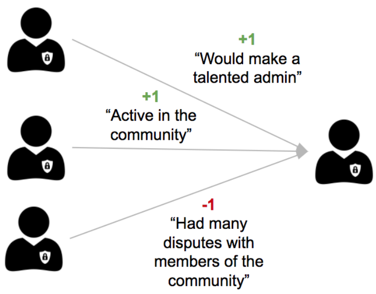

Figure 1: The RfA Dataset consists of voting results for new admins on Wikipedia with a comment accompanying it.

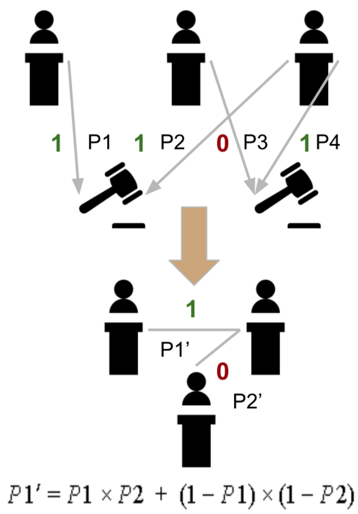

Figure 2: The Convote Dataset consists of voting results for bills and comments about votes. The person-to-item graph is transformed to person-to-person graph by linking two persons if they voted both on at least one bill.

2.2.2 The Convote Dataset

We transformed the undirected person-to-item graph of the Convote dataset into an undirected person-to-person graph. From the data we extracted the debates, the voters of the bills, the votes and the comments about the votes. We linked two voters if they voted on the same bill and assigned a "+1" signed edge when the two voters agreed on more than half of the bills they commonly voted on and "0" otherwise. Since the sentiment data was the score computed for each edge by SVMs classifier, we used the Platt Scaling technique (Platt, 1999) (logistic regression on all the SVM outputs with the corresponding vote) to transform this output into a sentiment probability. We then adapted the person-to-item Sentiment probability to a person-to-person probability as the average agreement probability. If q_u is the sentiment probability for node u on a bill and q_v is the sentiment probability for node v on the same bill, then their agreement probability is $q = q_u q_v + (1 - q_u)(1 - q_v)$ as shown if Figure 2. We then split the graph into 5 folds by picking the edges randomly (each fold had then 20% of the total number of edges). The resulting graph has 267 nodes, 14690 undirected edges and 506,327 triangles.

2.2.3 Data Use

The two datasets were selected for their textual content and graph like structure. With an aim at proving that strong Edge Inference can be obtained using both Sentiment Analysis and Network Social Theories, the chosen dataset were flexible and adapted well to different types of graphs. The Wikipedia dataset Sentiment Distribution is a very good indicator of the edge sign (average AUROC of 0.85 on the whole Dataset) but has relatively weak Network Structure (103.4 triangles per node on average). The Convote dataset had little information via the Sentiment Analysis (average AUROC of 0.66 on the whole Dataset), but a strong Network Structure (1896.4 triangles per node on average).

3 Data Analysis

3.1 Prediction Models

In this work we compared four prediction models which include a random model, sentiment model, network model and combined model. The combined model utilizes both the sentiment and network model.

3.1.1 Random model

We randomly pick an edge sign for the edges in the testing dataset. This model serves as a baseline to compare the performance of the other models.

3.1.2 Sentiment model

We predict the edge signs using our Sentiment Classifier. In the case of the Rfa Dataset the comment is the input to the model and the output is the Sentiment probability. In the case of the Convote Dataset, the edge sign score is equal to the preprocessed Sentiment probability distribution.

3.1.3 Network model

For the network model we minimize a cost function based on the edge signs of the triangles of the testing graph. In the case of the Rfa Dataset the cost function is minimized for triangles that respect the Social Status theory as shown in Figure 3 and maximal for other configurations. In the case of the Convote Dataset, we minimize the same cost function but with regards to the Social Balance theory as shown in Figure 4.

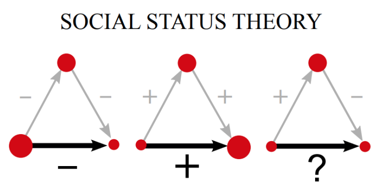

Figure 3: Social Status Theory graphs

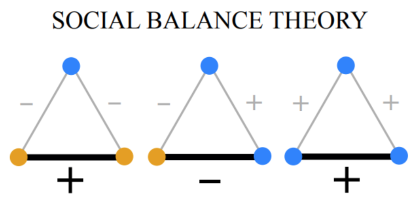

Figure 4: Social Balance Theory graphs

3.1.4 Combined model

For the combined model we minimize a cost function that depends on the Sentiment probability distribution over the edges and triangles configurations (respecting the Social Status theory/Social Balance theory). The weights for the individual components in the cost function can be changed according to the dataset for "best" fit. We then aim at proving that the combined model has the best scores outperforming Random model, Network model and Sentiment model.

3.2 Model Implementation and Analysis Methods

The models were implemented using a custom built Sentiment Classifier/preprocessed Sentiment probabilities for the Sentiment model, Probabilistic soft logic (PSL) (bach and jmlr17, 2017) for the Network Model and using both for the Combined Model.

3.2.1 Sentiment Classifier for RfA Dataset

The RfA Dataset contained text comments, and we built a sentiment classifier using Logistic Regression and L2 penalty. We selected the 10,000 most frequent words as features of the model and picked 1,000 comments randomly from the graph as well as the vote for training. We preprocessed the comments to feed the classifier by removing stop words, stemming the words and identifying duplicate words that were written differently (e.g. "don't" and "do not").

3.2.2 PSL Model Rules

The cost function for the network and combined model is defined by the weights used for the PSL rules. The PSL rules are stated in the form of an logical expression with a weight for each rule, where the weight states the importance of the rule. A prior can be applied to the edges of the testing graph. For the combined model implementation the Sentiment probability was as our prior. See the appendix for more specific information on the rules.

3.2.3 The Metrics

To compare the efficiency of the different models we used four metrics. The areas under the curve of the receiver operating characteristic (AUC/ROC), the areas under the curve of the negative Precision-Recall curve (AUC/negPR) and the Positive and Negative Precision-Recall curve. The goal for metric is to show how robust the combined model is when compared to other models. To demonstrate the combined model we weaken the contribution from the Sentiment or Network model and see if our combined model is able to obtain the best results dynamically. In the first case, to test the network analysis contribution, we reduce the evidence ratio of the testing graphs. That is, we reduce the number of observations from the edges. As expected the Network model would decline in performance while the Sentiment model performance would remain stable. On the other end to test the contribution from the sentiment analysis contribution we dropped the top features of the Sentiment Classifier to see how robust the combined model was when the Sentiment model starts to perform poorly. The features were sorted by their mutual information with the edge signs on the whole graph (the higher the MI, the higher the feature is predictive for the edge sign). For the Convote Dataset we unfortunately cannot drop the textual features as they were directly picked from the SVM output (and transformed into probabilities).

3.2.4 Analysis Effort

The preprocessing and the analysis took about a two weeks with over 50 hours to code. Several engineering problems occurred that took some time to overcome. The PSL package required significant computational resources and we had to reduce the testing sets of the RfA corpus to 200 nodes (instead of 350 nodes) as it overloaded the computer memory. A personal laptop was used.

4 Replication Results

4.1 Results of replication : Wikipedia Dataset

The results we obtained were very similar to the ones presented in the original research paper. The results from the replication are compared to the original results with graphs of the models performance using same metrics, scaling, and trace colors.

We see from the graphs on the next page that combined model is more sensitive to the contribution from Sentiment on the edges than from the Network Structure. Indeed, in the first two graphs Figures 5 and 6 there is a slight increase in initial performance of the Combined model compared to the Network model. However, in the next two graphs Figures 7 and 8 the dropping performance is greater, showing greater effect of the decrease in performance of the Sentiment model. We note that the Combined model is robust, performing well even in the case where the Sentiment model performs poorly.

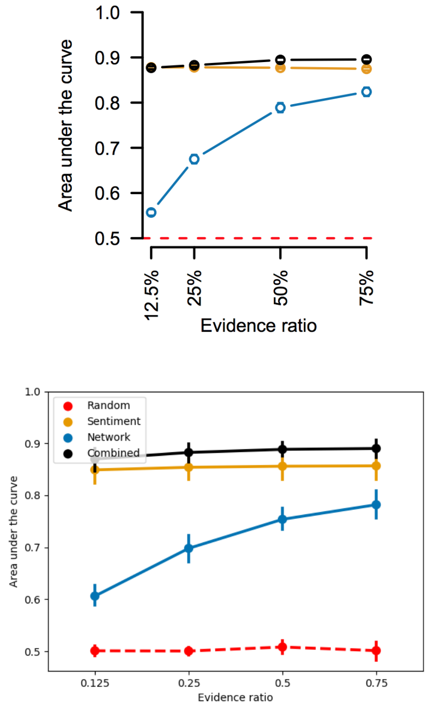

Figure 5: Analysis of Wikipedia data shown as AUC for ROC versus Evidence Ratio with a 95% confidence interval (original results on top and replicated results on bottom).

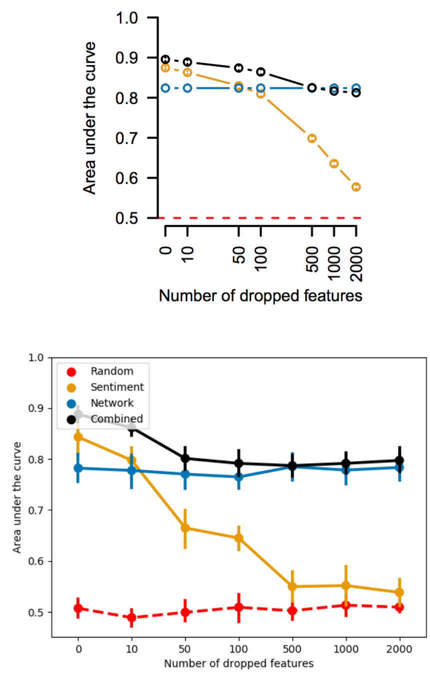

Figure 7: Analysis of Wikipedia data shown as ROC versus number of dropped features with a 95% confidence interval (original results on top and replicated results on bottom).

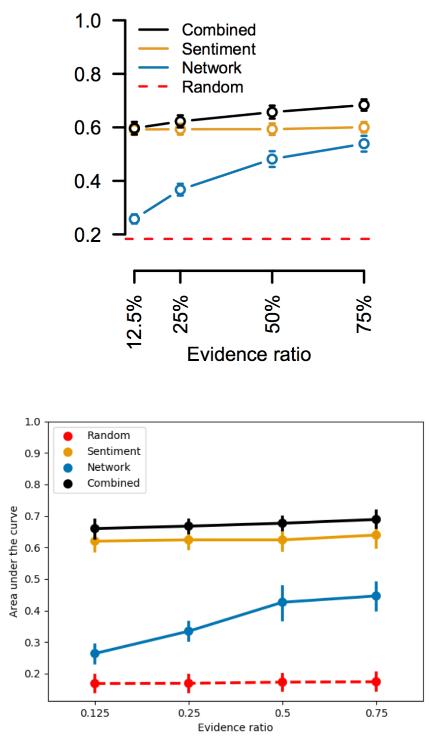

Figure 6: Analysis of Wikipedia data shown as AUC for negative PR versus Evidence Ratio with a 95% confidence interval (original results on top and replicated results on bottom).

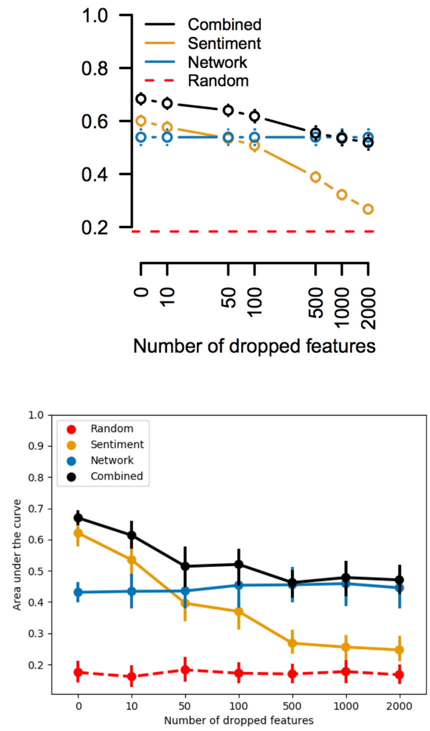

Figure 8: Analysis of Wikipedia data shown as negative PR versus number of dropped features with a 95% confidence interval (original results on top and replicated results on bottom).

The combined model has similar performance to the Network model when the Sentiment model contribution is significantly reduced. It is interesting to note that our results appear better than the results published in the original paper. We were perplexed that our results show an improvement as we took a simplified approach to their model implementation (the weights were not found using an unsupervised learning approach and a non-uniform weight distribution was not used on the Sentiment probabilities). The difference in our results can be explained by the fact that the weights used by the original authors were suboptimal for the problem. The optimizer used could have fallen into a local optimum while finding weight values in their approach. The model has sensitivity to the weights in cost function and rule weights and will further be explored in the extension of this replication.

4.2 Results of replication : Convote Dataset

The results we obtained were very similar to the ones presented in the original research paper, and in some cases improved (see Figures 9, 10, 11, 12). The results from the replication are compared to the original results with graphs of the models performance using same metrics, scaling, and trace colors.

The Combined model again outperforms the sole use of either the Network or Sentiment model alone. Even with lower levels of evidence the combined model outperforms the Sentiment and the Network model. In the original paper the combined model appears to be sensitive to the Network model and reducing the evidence ratio dramatically reduces the performance. The sensitivity from Network model could be reduced if more weight was biased towards the prior (Sentiment model).

4.3 Limitations and Discussion

The strength of the Combined model comes from the advantage of using both the network structure and textual data. Network and textual data is commonly found in modern social media data. In the case of social media, the network structure can be defined by the “Friend” network structure and the comments forming discussions on the edges between friends. In some cases a social media dataset may have a stronger sentiment with a weak network structure, and vice versa. Since the Combined models uses both sentiment and network structure to reinforce the prediction, this model is adaptable to a various range of datasets.

In the original work, the weights for both the PSL model rules and the cost function were chosen using a unsupervised learning approach. In this work the weight were chosen heuristically. It is interesting that this work, in some ways, showed an improvement in performance of our implementation of the Combined model. The distinction in the performance could be attributed to suboptimal choice of weight found in a local optimal.

From the results that were replicated, the original work has proven to be reproducible. The advantages of using a Combined model have been demonstrated on the same datasets used in the original work. The methods used in this work align with the approaches suggested by the original authors, but with significant modifications in terms of implementation details. Nevertheless, the essence of the methods provided in the original paper can be followed to construct a robust model for predicting relationships between people.

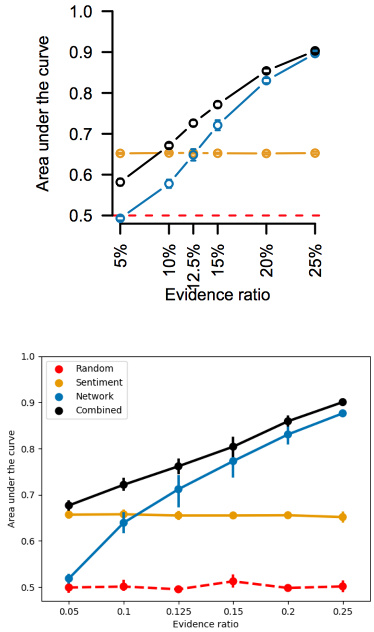

Figure 9: Analysis of Convote data shown as AUC for ROC versus Evidence Ratio with a 95% confidence interval (original results on top and replicated results on bottom).

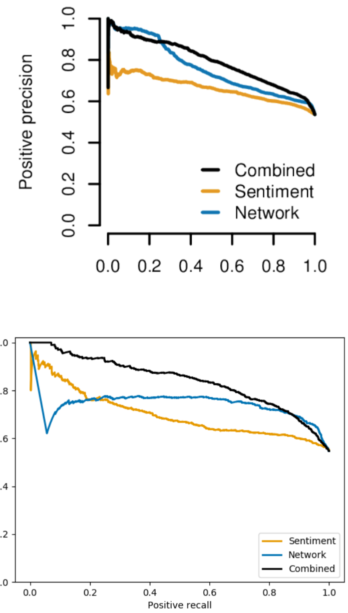

Figure 11: Analysis of Convote data shown as precision versus recall for positive person-person sign values (original results on top and replicated results on bottom).

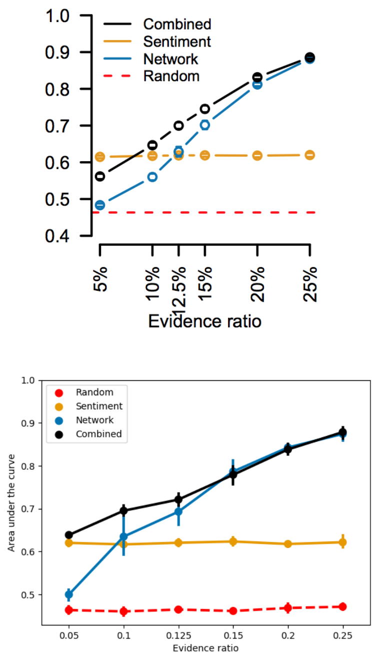

Figure 10: Analysis of Convote data shown as AUC for negative PR versus Evidence Ratio with a 95% confidence interval (original results on top and replicated results on bottom).

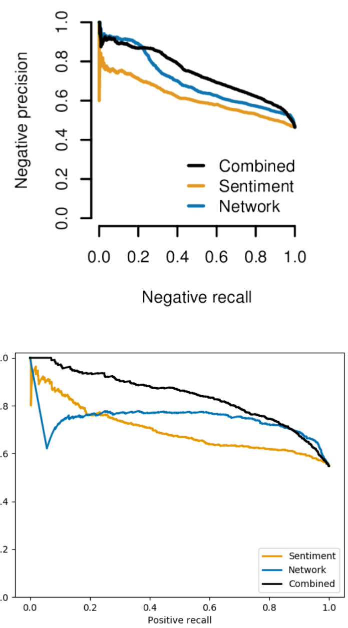

Figure 12: Analysis of Convote data shown as precision versus recall for negative person-person sign values (original results on top and replicated results on bottom).

5 Extension : Sensitivity to the constraint weights

To extend this work, we wanted to understand the impact on the weights selected for the PSL rules. This is an important aspect of the model which shows how critical the choice of the weights of the PSL rules is. We will focus on the results displayed in Figure 9 as we have seen the most difference in performance for low evidence ration when compared to the original paper results. We define Prior Weight (PW) as the ratio of Prior weight to the Network weight. For example, in the graph we obtained in Figure 9, the weight for the Network rules is 1.0 and is also 1.0 for the Prior (Sentiment probability distribution), making a ratio of 1. the following figure shows how much of an impact the ratio has on the performance of the Combined Model as shown in Figure 13.

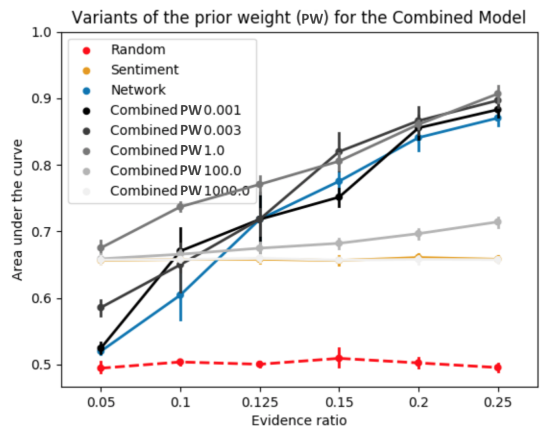

Figure 13: Convote data AUC/ROC versus evidence data with Combined Model using various Prior Weights (PW) ratios.

We can see that the combined model improves very quickly as PW increases and has a large band, for a given evidence ratio value, where the combined model performs better than Sentiment and Network model (see Figure 13. With larger PW values, Combined Model begins to closely resemble the Sentiment model, and a smaller PW values resembles the Network Model (see Figure 14. The range of performance results generated across the spectrum of PW values suggests that care should be taken when selecting the weights in order to find the optimal performance for the combined model as the peak of the optimal performance has a very small band (the length of the range of the PW in the peak is around 1 for figure 13). The decrease in performance being relatively slow, it is still possible to have a good performance (but suboptimal) for a broad range of values (typically between 0.1 and 10 in Figure 13).

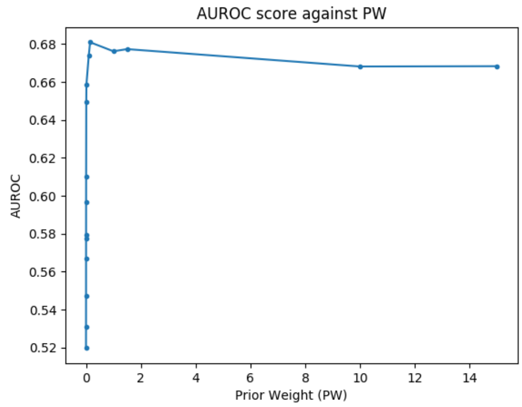

Figure 14: AUC/ROC against the Prior Weight (PW) for the Convote Dataset on the point of interest of 5% for the evidence ratio

Acknowledgments

The authors of this work would like to acknowledge Dr. Jacob Eisentein for his guidance and patience with our quandaries.

References

Robert West and Hristo S. Paskov and Jure Leskovec and Christopher Potts. 2014. *Exploiting Social Network Structure for Person-to-Person Sentiment Analysis*, CoRR.

Bach, Stephen H. and Broecheler, Matthias and Huang, Bert and Getoor, Lise. 2017. *Hinge-Loss Markov Random Fields and Probabilistic Soft Logic*, Journal of Machine Learning Research (JMLR).

Pang, Bo and Lee, Lillian. 2008. *Opinion mining and sentiment analysis.*, Foundations and Trends in Information Retrieval.

Cartwright, Dorwin and Harary, Frank. 2008. *Structure balance: A generalization of Heider's theory.*, Psychological Review.

Platt, John. 1999. *Probabilistic outputs for support vector machines and comparisons to regularize likelihood methods.*, Advances in Large Margin Classifiers.

appendix

WIKIPEDIA COMBINED MODEL, PSL RULES

```
16.0: (knows(A,B) & prior(A,B) & (A-B)) >> trusts(A,B) ^2
16.0: (knows(A,B) & trusts(A,B) & (A-B)) >> prior(A,B) ^2

1.0: (knows(A,B) & knows(B,C) & knows(A,C) & trusts(A,B) & trusts(B,C) & (A - B)
& (B - C) & (A - C)) >> trusts(A,C) ^2
1.0: (knows(A,B) & knows(B,C) & knows(A,C) & trusts(A,B) & ~trusts(B,C) & (A - B)
& (B - C) & (A - C)) >> ~trusts(A,C) ^2
1.0: (knows(A,B) & knows(B,C) & knows(A,C) & ~trusts(A,B) & trusts(B,C) & (A - B)
& (B - C) & (A - C)) >> ~trusts(A,C) ^2
1.0: (knows(A,B) & knows(B,C) & knows(A,C) & ~trusts(A,B) & ~trusts(B,C) & (A -
B) & (B - C) & (A - C)) >> trusts(A,C) ^2

1.0: (knows(A,B) & knows(C,B) & knows(A,C) & trusts(A,B) & trusts(C,B) & (A - B)
& (B - C) & (A - C)) >> trusts(A,C) ^2
1.0: (knows(A,B) & knows(C,B) & knows(A,C) & trusts(A,B) & ~trusts(C,B) & (A - B)
& (B - C) & (A - C)) >> ~trusts(A,C) ^2
1.0: (knows(A,B) & knows(C,B) & knows(A,C) & ~trusts(A,B) & trusts(C,B) & (A - B)
& (B - C) & (A - C)) >> ~trusts(A,C) ^2
1.0: (knows(A,B) & knows(C,B) & knows(A,C) & ~trusts(A,B) & ~trusts(C,B) & (A -
B) & (B - C) & (A - C)) >> trusts(A,C) ^2

1.0: (knows(B,A) & knows(B,C) & knows(A,C) & trusts(B,A) & trusts(B,C) & (A - B)
& (B - C) & (A - C)) >> trusts(A,C) ^2
1.0: (knows(B,A) & knows(B,C) & knows(A,C) & trusts(B,A) & ~trusts(B,C) & (A - B)
& (B - C) & (A - C)) >> ~trusts(A,C) ^2
1.0: (knows(B,A) & knows(B,C) & knows(A,C) & ~trusts(B,A) & trusts(B,C) & (A - B)
& (B - C) & (A - C)) >> ~trusts(A,C) ^2
1.0: (knows(B,A) & knows(B,C) & knows(A,C) & ~trusts(B,A) & ~trusts(B,C) & (A -
B) & (B - C) & (A - C)) >> trusts(A,C) ^2

1.0: (knows(B,A) & knows(C,B) & knows(A,C) & trusts(B,A) & trusts(C,B) & (A - B)
& (B - C) & (A - C)) >> trusts(A,C) ^2
1.0: (knows(B,A) & knows(C,B) & knows(A,C) & trusts(B,A) & ~trusts(C,B) & (A - B)
& (B - C) & (A - C)) >> ~trusts(A,C) ^2
1.0: (knows(B,A) & knows(C,B) & knows(A,C) & ~trusts(B,A) & trusts(C,B) & (A - B)
& (B - C) & (A - C)) >> ~trusts(A,C) ^2
1.0: (knows(B,A) & knows(C,B) & knows(A,C) & ~trusts(B,A) & ~trusts(C,B) & (A -
B) & (B - C) & (A - C)) >> trusts(A,C) ^2
```

CONVOTE COMBINED MODEL, PSL RULES

```
1.0: (covoted(A,B) & prior(A,B) & (A-B)) >> agree(A,B) ^2
1.0: (covoted(A,B) & agree(A,B) & (A-B)) >> prior(A,B) ^2

1.0: (covoted(A,B) & covoted(B,C) & covoted(C,A) & ~agree(A,B) & ~agree(B,C) & (A
- B) & (B - C) & (C - A)) >> agree(C,A) ^2
1.0: (covoted(A,B) & covoted(B,C) & covoted(A,C) & ~agree(A,B) & ~agree(B,C) & (A
- B) & (B - C) & (C - A)) >> agree(A,C) ^2
1.0: (covoted(A,B) & covoted(C,B) & covoted(C,A) & ~agree(A,B) & ~agree(C,B) & (A
```

Appendices

```
- B) & (B - C) & (C - A)) >> agree(C,A) ^2
```

PSL rules

```
1.0: (covoted(A,B) & covoted(C,B) & covoted(A,C) & ~agree(A,B) & ~agree(C,B) & (A
- B) & (B - C) & (C - A)) >> agree(A,C) ^2
1.0: (covoted(B,A) & covoted(B,C) & covoted(C,A) & ~agree(B,A) & ~agree(B,C) & (A
- B) & (B - C) & (C - A)) >> agree(C,A) ^2
1.0: (covoted(B,A) & covoted(B,C) & covoted(A,C) & ~agree(B,A) & ~agree(B,C) & (A
- B) & (B - C) & (C - A)) >> agree(A,C) ^2
1.0: (covoted(B,A) & covoted(C,B) & covoted(C,A) & ~agree(B,A) & ~agree(C,B) & (A
- B) & (B - C) & (C - A)) >> agree(C,A) ^2
1.0: (covoted(B,A) & covoted(C,B) & covoted(A,C) & ~agree(B,A) & ~agree(C,B) & (A
- B) & (B - C) & (C - A)) >> agree(A,C) ^2

1.0: (covoted(A,B) & covoted(B,C) & covoted(C,A) & ~agree(A,B) & agree(B,C) & (A
- B) & (B - C) & (C - A)) >> ~agree(C,A) ^2
1.0: (covoted(A,B) & covoted(B,C) & covoted(A,C) & ~agree(A,B) & agree(B,C) & (A
- B) & (B - C) & (C - A)) >> ~agree(A,C) ^2
1.0: (covoted(A,B) & covoted(C,B) & covoted(C,A) & ~agree(A,B) & agree(C,B) & (A
- B) & (B - C) & (C - A)) >> ~agree(C,A) ^2
1.0: (covoted(A,B) & covoted(C,B) & covoted(A,C) & ~agree(A,B) & agree(C,B) & (A
- B) & (B - C) & (C - A)) >> ~agree(A,C) ^2
1.0: (covoted(B,A) & covoted(B,C) & covoted(C,A) & ~agree(B,A) & agree(B,C) & (A
- B) & (B - C) & (C - A)) >> ~agree(C,A) ^2
1.0: (covoted(B,A) & covoted(B,C) & covoted(A,C) & ~agree(B,A) & agree(B,C) & (A
- B) & (B - C) & (C - A)) >> ~agree(A,C) ^2
1.0: (covoted(B,A) & covoted(C,B) & covoted(C,A) & ~agree(B,A) & agree(C,B) & (A
- B) & (B - C) & (C - A)) >> ~agree(C,A) ^2
1.0: (covoted(B,A) & covoted(C,B) & covoted(A,C) & ~agree(B,A) & agree(C,B) & (A
- B) & (B - C) & (C - A)) >> ~agree(A,C) ^2

1.0: (covoted(A,B) & covoted(B,C) & covoted(C,A) & agree(A,B) & agree(B,C) & (A -
B) & (B - C) & (C - A)) >> agree(C,A) ^2
1.0: (covoted(A,B) & covoted(B,C) & covoted(A,C) & agree(A,B) & agree(B,C) & (A -
B) & (B - C) & (C - A)) >> agree(A,C) ^2
1.0: (covoted(A,B) & covoted(C,B) & covoted(C,A) & agree(A,B) & agree(C,B) & (A -
B) & (B - C) & (C - A)) >> agree(C,A) ^2
1.0: (covoted(A,B) & covoted(C,B) & covoted(A,C) & agree(A,B) & agree(C,B) & (A -
B) & (B - C) & (C - A)) >> agree(A,C) ^2
1.0: (covoted(B,A) & covoted(B,C) & covoted(C,A) & agree(B,A) & agree(B,C) & (A -
B) & (B - C) & (C - A)) >> agree(C,A) ^2
1.0: (covoted(B,A) & covoted(B,C) & covoted(A,C) & agree(B,A) & agree(B,C) & (A -
B) & (B - C) & (C - A)) >> agree(A,C) ^2
1.0: (covoted(B,A) & covoted(C,B) & covoted(C,A) & agree(B,A) & agree(C,B) & (A -
B) & (B - C) & (C - A)) >> agree(C,A) ^2
1.0: (covoted(B,A) & covoted(C,B) & covoted(A,C) & agree(B,A) & agree(C,B) & (A -
B) & (B - C) & (C - A)) >> agree(A,C) ^2
```