Understanding the effects of collective classification on learning and inference

Jennifer Neville
Departments of Computer Science and Statistics
Purdue University

(joint work with Hoda Eldardiry, Ankit Kuwadekar, and Rongjing Xiang)
Network prediction models

Goal: Estimate joint distribution $P(Y|\{X\}_n, G)$
... or conditional distribution $P(Y_i|X_i, X_R, Y_R)$
Network prediction models

**Within-network prediction**: Use joint distribution to collectively infer unobserved class labels.
What are the challenges to developing collective inference models in single network domains?
Datasets are sparsely labeled

Many collective learning methods assume a fully labeled training graph

Facebook Political Views

Conservative

Unknown
There are multiple types of links between people

Multi-relational methods can model link types but this may not be the best way to exploit link structures
Network structure affects accuracy of collective inference

Random graph
Labeled nodes: 30%
Autocorrelation: 50%
Network structure affects accuracy of collective inference

Collective inference process introduces a new source of error for both learning and prediction

Small world graph
Labeled nodes: 30%
Autocorrelation: 50%
Exploiting multiple networks in collective classification models
Reducing error of collective inference models

• Collective classification models have two sources of error \((MLJ’08)\)
  • Error due to learning: same error as in i.i.d. settings
  • Error due to inference process: new type of due to collective inference

• Ensemble methods (e.g., bagging) are useful for reducing error due to variance
  • Have proved to be very useful in i.i.d. domains
  • But they have only considered reduction in \textit{learning variance}

• \textbf{Goal}: Improve prediction accuracy in network domains by reducing errors due to learning and inference
Conventional ensembles

Training Set → \( M_1 \) → Test Set

Training Set → \( M_2 \) → Test Set

Training Set → \( M_3 \) → Test Set

Samples

Models

Model predictions
Ensemble classification for network domains

- Training Set
- Samples
- Learn models from samples
- Test set
- Model predictions

Set $M_1, M_2, M_3$ learn models from samples.
Many network domains have multiple types of links
Collective ensemble classification (AAAI’11)

Basic ensemble approach:
Learn an ensemble of models, one on each link source, and combine predictions after collective inference

Collective fusion approach:
Exploits unique opportunity… to **interleave and aggregate across** the multiple collective inference processes
Experimental results

CEC significantly outperforms alternative methods over a wide range of conditions.

CEC accuracy improvement corresponds to reduction in inference variance.
Observations

- Collective inference can introduce **error due to variance** in the inference process (propagation effects magnify error)
- Using ensembles to average predictions **during collective inference**, can reduce both learning and inference variance, and thus increase accuracy of predictions

**Lesson learned:**

- Different link types can be “redundant” sources of information, better to learn independent models and combine rather than learn a complex joint model
Active learning of collective inference models in sparsely labeled networks
Active Learning

- In some domains there are an abundance of unlabeled data instances, but it is costly to acquire labels for those instances (to learn classification models).

- The goal of active learning is to identify the **best instances to label**, i.e., those that offer the most benefit to learning.

- There has been little work focusing on active learning methods for relational networks.
  - In relational domains, the utility of labeling a node may depend on its larger **context** in the network.
Challenges for active learning in relational domains
Learning with partially-labeled networks

If learning methods ignore unlabeled nodes then graph structure changes as labeling increases.
Prediction uncertainty is typically estimated by using i.i.d. bootstrap sampling to learn and apply multiple models...

...but sampling repeatedly from a single, interconnected network is difficult.
Separating error due to learning and inference

Collective inference uses labeled nodes to seed the inference process...

Predictions can be improved BOTH by learning a better model AND by increasing the availability of labels for inference.

Thus to improve learning we need to distinguish learning and inference uncertainty.
Relational active learning *(ICML'11)*

• **Main ideas**

  • **Network-based utility metric** that accounts for both node and neighborhood characteristics

  Main finding: When label propagation is used for learning... we find that labeling based on **certainty** is more effective than labeling based on uncertainty.

  • **Across-network evaluation** to separate effects of label propagation on learning and prediction
Ensemble prediction

- Resample a set of pseudosample networks from the partially-labeled training data \((\text{Eldardiry} \& \text{Neville} \ '08)\)

- Using semi-supervised learning, estimate a model and predict values for the unlabeled nodes in each pseudosample
  - Use pseudolikelihood EM \((\text{Xiang} \& \text{Neville} \ '09)\) to learn a RDN model over a partially-labeled graph
  - Calculate the utility of labeling a node based on the set of predicted values from the ensemble
Network utility measure

- **Weighted density disagreement (WDD):**
  - Maximize KL divergence between node predictions and the average predictions for all unlabeled nodes in the graph, and...
  - Minimize KL divergence between node predictions and the predictions of its neighbors

\[
u_{WDD}(i) = v(i) \times \sum_{j \in N_i} e^{-KL[\hat{P}(i)||\hat{P}(j)]}
\]

where
\[
v(i) = KL[\hat{P}(i)||\hat{P}(UL)] = \sum_y \hat{P}(i) \times \log \frac{\hat{P}(i)}{\hat{P}(UL)}
\]

- Favors nodes that have:
  - Highly confident predictions (i.e., diverge from the overall avg predictions)
  - Neighbors with similar predictions
Comparison of algorithms

IMDb

AdHealth data
Metric comparison

Synthetic data

AUC

SIZE OF LABELED TRAINING SET

Legend:
- MOST CONFIDENT - MOST CONSISTENT
- LEAST CONFIDENT - MOST INCONSISTENT
- LEAST CONFIDENT - MOST CONSISTENT
- MOST CONFIDENT - MOST INCONSISTENT
Observations

• RAL achieves significantly higher accuracy compared to other iid and relational methods
  - The WDD measure favors nodes that have:
    Highly confident predictions (i.e., max divergence from overall avg)
    Neighbors with similar predictions (i.e., min divergence)

• This is opposite to many utility metrics used in i.i.d. settings, which favor nodes with high uncertainty.

• We conjecture that WDD success is due to semi-supervised network learning, which propagates inferences during learning
  - Estimation could be biased if nodes with less consistent neighborhoods are labeled initially... need theoretical analysis to show this.
Understanding the impact of propagation error for learning collective classification models
Collective classification with Markovian models

- General model:

\[ P(y_G|x_G) = \frac{1}{Z(\theta, x_G)} \prod_{T \in T} \prod_{C \in C(T(G))} \Phi_T(x_C, y_C; \theta_T) \]

- Model parameters are estimated from a labeled subgraph drawn from the underlying network domain.

- Learned model is then applied to collectively infer the unknown class labels in a separate (test) subgraph from the same underlying domain.
Classification Error

\[
\text{Error}(y^*_G, P_\theta) = \frac{1}{|G \setminus L|} \sum_{i \in G \setminus L} P(y_i \neq y_i^* | y_L^*)
\]

\[
= 1 - \frac{1}{|G \setminus L|} \sum_{i \in G \setminus L} P(y_i^* | y_L^*)
\]

- Note that the amount and spread of test set labels affects classification performance
  - However, there has been little work on how to learn models for different label availability scenarios
Learning Relational Models

- Much of previous research has focused on developing efficient approximations to maximum likelihood estimation for network domains (e.g., pseudolikelihood)

<table>
<thead>
<tr>
<th>Maximum Likelihood Estimation</th>
<th>Maximum Pseudolikelihood</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta^{\text{MLE}} = \arg\max_{\theta} \log P(y_G</td>
<td>x_G)$</td>
</tr>
<tr>
<td>$= \arg\max_{\theta} \sum_{C \in C(G)} \langle \theta_T, \phi_T(x_C, y_C) \rangle - \log Z(\theta, x_G)$</td>
<td>$= \arg\max_{\theta} \sum_{i \in G} \left( \varphi_i - \log \sum_{y_i, y_{\partial i}} \exp(\varphi_i) \right)$</td>
</tr>
<tr>
<td>Enforces global consistency by normalization</td>
<td>Does not enforce global consistency</td>
</tr>
<tr>
<td>Computationally intensive: exponential in $n$</td>
<td>Computationally efficient: polynomial in $n$</td>
</tr>
</tbody>
</table>

- This focus on MLE is because MLE is statistically optimal for i.i.d. data... however, this does not carry over to networks with partially observed labels
Comparison of learning algorithms
Discrepancy between MPLE and MLE

• Why does MPLE outperform MLE when there are abundant labels?
  • In this case, the predictive distribution $P_\theta(y_i|x_G, y_L)$ is close to the MPLE distribution $P_\theta(y_i|x, x_{\partial_i}, y_{\partial_i})$ and thus the empirical distribution on the test graph $\tilde{\pi}(y_i|x, x_{\partial_i}, y_{\partial_i})$

• Why does MPLE underperform when labeled nodes are scarce?

  Varying levels of label availability change the relative importance of local dependency and global propagation.

• MPLE also fails to enforce global consistency during training and consequently attributes all dependencies in the training network to local (e.g., pairwise) dependency
  • When applying the MPLE parameter estimates collectively in the test network, the local dependencies are propagated globally throughout the unlabeled instances, which results in propagation error.
Considering labeling effects during learning \((ICDM’11)\)

- Inference error decomposition for each instance \(i\):
  \[
  \epsilon_i = 1 - P(y_i^* | y_L^*) = [1 - P(y_i^* | y_{\partial i}^*)] + [P(y_i^* | y_{\partial i}^*) - P(y_i^* | y_L^*)]
  \]
  - **Base error**: reflects the quality of model family specification and the training accuracy
  - **Propagation error**: caused by the collective inference mechanism based on partially observed labels

- Proposed model:
  - Use a local mixture model for model selection to combine the strengths of MPLE and a low propagation model (e.g., MLE):
    \[
    \mu(y_i) = \lambda_i P_{\theta}(y_i | x_i, y_i, x_{\partial i}, y_{\partial i}) + (1 - \lambda_i) \tilde{P}(y_i | x_i, x_{\partial i}, y_{\partial i})
    \]
  - We use an estimate of the propagation upper bound as mixture indicator, localized to each node to take instance heterogeneity into account
  - Propagation bound is based on estimate of microscopic dependencies
Experimental results

- Our mixture model outperforms both MPLE and the low propagation model across the whole spectrum of label availability.
Observations

• For collective inference models, the relative performance of different learning methods is inconsistent across the spectrum of test set label availability
  • This indicates a previously unidentified trade-off between MPLE and MLE-type estimation methods

• We investigated this issue by characterizing the propagation error of MPLE in order to analyze the trade-off
  • Our analysis points to a novel learning method that can improve collective classification performance across the full range of test set label availability
Conclusion

- Relational dependencies can significantly improve predictions through the use of collective inference models...
- ...but current methods make assumptions about data and model characteristics that are often not appropriate
  - Link information is heterogeneous, not uniform/stationary
  - Label and attribute information is sparse, not fully labeled
  - Data comprises a single network, not a population of networks
  - Error arise from both estimation and prediction, not just estimation
- Need to consider graph/data structure carefully and understand its impact on modeling in order to best exploit the relational information for prediction
  - In particular, we need to characterize effects analytically... but heterogeneous, dependent structure makes this difficult
Questions?

neville@cs.purdue.edu
www.cs.purdue.edu/~neville