Hidden Common Cause Relations in Relational Learning

- Ricardo Silva (Gatsby Unit/UCL)
- Wei Chu (Columbia)
- Zoubin Ghahramani (Cambridge)

silva@statslab.cam.ac.uk
In a Nutshell

- The problem: classification with non-iid data
  - The source of non-iidness: relational information
- A new family of models:
  - Where conditioning creates dependence
  - This means chains of training points generate “long distance” dependencies
  - Distinct from and complements Markov networks
- Experiments with classification of text documents
Learning with Non-IID Data
Hidden Common Cause Relations

X:
- Capital (GE)
- Capital (Westinghouse)

Y:
- Stock price (GE)
- Stock price (Westinghouse)

Industry factor 1
Industry factor 2
Industry factor k?

…
Notation: Directed Mixed Graphs

X:
- Capital(GE)
- Stock price(GE)

Y:
- Capital (Westinghouse)
- Stock price (Westinghouse)

Richardson (2003)
What are the implications? – a comparison with Markov networks/CRFs

\[ Y_i \] observed node \quad \[ Y_i \] unobserved node

\[ Y_1 \] \rightarrow \[ Y_2 \] \rightarrow \[ Y_3 \] \rightarrow \[ Y_4 \]  
Information from \[ Y_1 \] passes to \[ Y_4 \]

\[ Y_1 \] \rightarrow \[ Y_2 \] \rightarrow \[ Y_3 \] \rightarrow \[ Y_4 \]  
Information from \[ Y_1 \] does not pass to \[ Y_4 \]

\[ Y_1 \] \leftrightarrow \[ Y_2 \] \rightarrow \[ Y_3 \] \rightarrow \[ Y_4 \]  
Information from \[ Y_2 \] does not pass to \[ Y_4 \]

\[ Y_1 \] \leftrightarrow \[ Y_2 \] \rightarrow \[ Y_3 \] \rightarrow \[ Y_4 \]  
Information from \[ Y_2 \] passes to \[ Y_4 \]
Model for Binary Classification

- Non-parametric probit regression

\[ P(y_i = 1 | x_i) = P(y^*(x_i) > 0) \]
\[ y^*(x_i) = f(x_i) + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, 1) \]

- Zero-mean Gaussian process prior over \( f(\cdot) \)

- Relational dependency model:
  - Make \( \{\varepsilon\} \) dependent multivariate Gaussian, unit variance
  - For convenience, decouple it into two error terms

\[ \varepsilon = \varepsilon^* + \zeta \]
Dependency Model: the Decomposition

\[ \varepsilon = \varepsilon^* + \zeta \]

Independent from each other

Marginally independent

Dependent according to relations

\[ \sum \varepsilon = \sum \varepsilon^* + \sum \zeta \]

Diagonal

Not diagonal, with 0s only on unrelated pairs
Dependency Model: the Decomposition

\[ y^*(x_i) = f(x_i) + \varepsilon = f(x_i) + \zeta + \varepsilon^* = g(x_i) + \varepsilon^* \]

- If \( K \) was the original kernel matrix for \( f(\cdot) \), the covariance of \( g(\cdot) \) is simply

\[ \Sigma_{g(.)} = K + \Sigma_{\varepsilon^*} \]

- Plugging-in Expectation-Propagation:
  - Likelihood does not factorize over \( f(\cdot) \), but factorizes over \( g(\cdot) \! \):

\[ p(g \mid x, y) \propto p(g \mid x) \prod_i p(y_i \mid g(x_i)) \]
Parameterizing the Relational Covariance $\Sigma_\zeta$

- “Poking” zeroes in a covariance matrix is tricky:

\[
\begin{bmatrix}
1 & 0.8 & 0.8 \\
0.8 & 1 & 0.8 \\
0.8 & 0.8 & 1
\end{bmatrix}
\]

- (Note: Markov network forces zeros on the inverse)

\[
\Sigma^{-1} =
\begin{bmatrix}
1 & \rho_{12} & 0 \\
\rho_{12} & 1 & \rho_{23} \\
0 & \rho_{23} & 1
\end{bmatrix}
\]
Parameterizing the Relational Covariance $\Sigma_\zeta$

- Find all cliques and create a latent variable for each.
- Rescale marginal correlation matrix $U$ by a factor $\rho$
  - $\Sigma_\zeta = \rho U$
  - $\rho$ becomes a hyperparameter in $[0, 1]$
- In practice, cannot extract all cliques
- Suggestion: triangulate and then extract
  - A relaxation of the problem (not always harmless)
Experimental Setup

- Three text classification tasks

Comparisons:
- Standard Gaussian Process classifiers
- Standard GPs with link features
- The relational GP (RGP) of Chu et al. (2006 – Last NIPS)
- Our Mixed Graph Gaussian Process: XGP
- Linear kernels

Criterion:
- Area under the curve (AUC)

Transductive setting:
- Test points given in advance
Experiment I: Political Books dataset

- 105 books: conservative or liberal?
- Text extracted from Amazon.com front pages
- Available at www.statslab.cam.ac.uk/~silva

- 50% training, 50% test
- AUC for standard GP: 0.92
- AUC for RGP and XGP about the same: 0.98
Experiment II: Subset of CORA

- Database of publications in Computer Science
- 1% for training, 99% for test (too easy)
  - Very “uniform” links – mostly between same class papers
- XGP cannot do better than RGP when there is so little training data to propagate information

Table 1: The averaged AUC scores of citation prediction on test cases of the Cora database are recorded along with standard deviation over 100 trials. “\( n \)” denotes the number of papers in one class. “Citations” denotes the citation count within the two paper classes.

<table>
<thead>
<tr>
<th>Group</th>
<th>( n )</th>
<th>Citations</th>
<th>GPC</th>
<th>GPC with Citations</th>
<th>XGP</th>
</tr>
</thead>
<tbody>
<tr>
<td>5vs1</td>
<td>346/488</td>
<td>2466</td>
<td>0.905 ± 0.031</td>
<td>0.891 ± 0.022</td>
<td>0.945 ± 0.053</td>
</tr>
<tr>
<td>5vs2</td>
<td>346/619</td>
<td>3417</td>
<td>0.900 ± 0.032</td>
<td>0.905 ± 0.044</td>
<td>0.933 ± 0.059</td>
</tr>
<tr>
<td>5vs3</td>
<td>346/1376</td>
<td>3905</td>
<td>0.863 ± 0.040</td>
<td>0.893 ± 0.017</td>
<td>0.883 ± 0.013</td>
</tr>
<tr>
<td>5vs4</td>
<td>346/646</td>
<td>2858</td>
<td>0.916 ± 0.030</td>
<td>0.887 ± 0.018</td>
<td>0.951 ± 0.042</td>
</tr>
<tr>
<td>5vs6</td>
<td>346/281</td>
<td>1968</td>
<td>0.887 ± 0.054</td>
<td>0.843 ± 0.076</td>
<td>0.955 ± 0.041</td>
</tr>
<tr>
<td>5vs7</td>
<td>346/529</td>
<td>2948</td>
<td>0.869 ± 0.045</td>
<td>0.867 ± 0.041</td>
<td>0.926 ± 0.076</td>
</tr>
</tbody>
</table>
Experiment III: WebKB

- Hardest task: “outlier” detection
  - Identify pages that are not student/faculty/department/project
- Notice that links between pages are of all sorts
  - Makes sense to propagate information only if class label is given
- 10% for training, 90% for test

Table 2: Comparison of the three algorithms on the task “other” vs. “not-other” in the WebKB domain. Results for GPC and RGP taken from [2]. The same partitions for training and test are used to generate the results for XGP. Mean and standard deviation of AUC results are reported.

<table>
<thead>
<tr>
<th>University</th>
<th>Numbers</th>
<th>Other</th>
<th>All</th>
<th>Link</th>
<th>GPC</th>
<th>RGP</th>
<th>XGP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cornell</td>
<td></td>
<td>617</td>
<td>865</td>
<td>13177</td>
<td>0.708 ± 0.021</td>
<td>0.884 ± 0.025</td>
<td>0.917 ± 0.022</td>
</tr>
<tr>
<td>Texas</td>
<td></td>
<td>571</td>
<td>827</td>
<td>16090</td>
<td>0.799 ± 0.021</td>
<td>0.906 ± 0.026</td>
<td>0.949 ± 0.015</td>
</tr>
<tr>
<td>Washington</td>
<td></td>
<td>939</td>
<td>1205</td>
<td>15388</td>
<td>0.782 ± 0.023</td>
<td>0.877 ± 0.024</td>
<td>0.923 ± 0.016</td>
</tr>
<tr>
<td>Wisconsin</td>
<td></td>
<td>942</td>
<td>1263</td>
<td>21594</td>
<td>0.839 ± 0.014</td>
<td>0.899 ± 0.015</td>
<td>0.941 ± 0.018</td>
</tr>
</tbody>
</table>
Conclusions

- Truly new relational model
  - Remember to think: graphical models are more than drawings
- Trivial to implement
  - One can reuse GP classifier code easily
- Requires one more hyperparameter only
- Many directions to explore:
  - So far, extremely simple covariance parameterizations
    - Several alternatives of parameterization as open directions
  - Combination of different relationships
    - Multiple kernel learning
  - Different models, heteroskedastic noise, full Bayesian learning, etc.
- Code available at http://www.statslab.cam.ac.uk/~silva