Predictive Discrete Latent Factor Models
for large incomplete dyadic data

Deepak Agarwal, Srujana Merugu, Abhishek Agarwal
Y! Research
MMDS Workshop, Stanford University
6/25/2008
Agenda

• Motivating applications
• Problem Definitions
• Classic approaches
• Our approach – PDLF
  – Building local models via co-clustering
• Enhancing PDLF via factorization
• Discussion
Motivating Applications

- Movie (Music) Recommendations
  - (Netflix, Y! Music)
  - Personalized; based on historical ratings
- Product Recommendation
  - Y! shopping: top products based on browse behavior
- Online advertising
  - What ads to show on a page?
- Traffic Quality of a publisher
  - What is the conversion rate?
DATA

COVARIATES
$(u_i, v_j)$

DYAD $(i, j)$

RESPONSE: $y_{ij}$ (ratings, click rates, conversion rates)

user, webpage

Movie, music, product, ad
Problem Definition

GOAL:

\[(i, j); \]
Training data

Predict Response

• CHALLENGES

Scalability: Large dyadic matrix
Missing data: Small fraction of dyads
Noise: SNR low; data heterogeneous but there are strong interactions
Classical Approaches

- **SUPERVISED LEARNING**
  - Non-parametric Function Estimation
  - Random effects: to estimate interactions

- **UNSUPERVISED LEARNING**
  - Co-clustering, low-rank factorization,…

- Our main contribution
  - Blend supervised & unsupervised in a model based way; scalable fitting
Non-parametric function estimation

\[ y_{ij} = h(x_{ij}) + \text{noise} \]

- E.g. Trees, Neural Nets, Boosted Trees, Kernel Learning,…
  - capture entire structure through covariates
  - Dyadic data: Covariate-only models shows “Lack-of-Fit”, better estimates of interactions possible by using information on dyads.
Random effects model

• Specific term per observed cell

\[ f(y_{ij}; z_{ij}, x_{ij}) = \int f_{\psi}(y_{ij}; z_{ij}^T \beta + x_{ij}^T \delta_{ij}) \pi(\delta_{ij}; G) d\beta_{ij} \]

• Smooth dyad-specific effects using prior(“shrinkage”)
  – E.g. Gaussian mixture, Dirichlet process,..

• Main goal: hypothesis testing, not suited to prediction
  – Prediction for new cell is based only on estimated prior

• Our approach
  – Co-cluster the matrix; local models in each cluster
  – Co-clustering done to obtain the best model fit
Classic Co-clustering

• Exclusively capture interactions  
  → No covariates included!

• **Goal**: Prediction by Matrix Approximation

• Scalable  
  – Iteratively cluster rows & cols  
    → homogeneous blocks
Smooth Rows using column clusters; reduces variance

After ROW CLUSTERING

After COLUMN Clustering

Iterate Until convergence
Our Generative model

\[ f(y_{ij}, x_{ij}, z_{ij}) = \sum_{i=1}^{K} \sum_{j=1}^{L} P(\rho_i = I, \gamma_j = J) f_{\psi}(y_{ij}; z_{ij}^T \beta + x_{ij}^T \delta_{i,j}) \]

\[ \rho_i = \text{Cluster id for row } i ; \gamma_j = \text{Cluster id for column } j \]

- Sparse, flexible approach to learn dyad-specific coeffs
  - borrow strength across rows and columns

- Capture interactions by co-clustering
  - Local model in each co-cluster
  - Convergence fast, procedure scalable
  - Completely model based, easy to generalize

- We consider \( x_{ij} = 1 \) in this talk
**Scalable model fitting**

**EM algorithm**

**Hard** assignment or "Winner - Take all"

Row/col assigned to the best cluster

\[ \rho_i = \arg \max_j \left( \sum_{j:(i,j) \in \kappa} (y_{ij} \delta_{l_{ij}} - \psi(x_{ij}^T \beta + \delta_{l_{ij}}) ) \right) \]

\[ \gamma_j = \arg \max_i \left( \sum_{i:(i,j) \in \kappa} (y_{ij} \delta_{\rho_i,j} - \psi(x_{ij}^T \beta + \delta_{\rho_i,j}) ) \right) \]

Easily done in parallel; we use Map - Reduce

Several million dyads; thousands of rows/columns take few hours

**Conditional on cluster assignments:**

Estimate parameters via usual statistical procedures

**Complexity:** \( O(N((K + L) + s^2)) \)
Simulation on Movie Lens

- User-movie ratings
  - Covariates: User demographics; genres
  - Simulated (200 sets): estimated co-cluster structure
  - Response assumed Gaussian

<table>
<thead>
<tr>
<th></th>
<th>$\beta_0$</th>
<th>$\beta_1$</th>
<th>$\beta_2$</th>
<th>$\beta_3$</th>
<th>$\beta_4$</th>
<th>$\sigma^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>truth</td>
<td>3.78</td>
<td>0.51</td>
<td>-0.28</td>
<td>0.14</td>
<td>0.24</td>
<td>1.16</td>
</tr>
<tr>
<td>95%  c.i</td>
<td>3.66,3.84</td>
<td>-0.63,0.62</td>
<td>-0.58,-0.16</td>
<td>-0.09,0.18</td>
<td>-0.68,1.05</td>
<td>0.90,0.99</td>
</tr>
</tbody>
</table>
Regression on Movie Lens

Rating > 3: +ve
23 covariates

Precision

Recall

Regression
Co-clustering
LatentFactor

PDLF
Pure co-clustering
Logistic Regression
Click Count Data

Goal:
Click activity on publisher pages from ip-domains

Dataset:
- 47903 ip-domains, 585 web-sites, 125208 click-count observations
- two covariates: ip-location (country-province) and routing type (e.g., aol pop, anonymizer, mobile-gateway), row-col effects.

Model:
- PDLF model based on Poisson distributions with number of row/column clusters set to 5

We thank Nicolas Eddy Mayoraz for discussions and data
Co-cluster Interactions: Plain Co-clustering

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet-related</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e.g., buydomains</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shopping search</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e.g., netguide</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AOL &amp; Yahoo</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smaller publishers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e.g. blogs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smaller portals</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e.g. MSN, Netzero</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The heat map shows the interactions between different categories of publishers and the traffic sources. The colors range from dark red to yellow, indicating the strength of interactions. The circles highlight areas with significant interactions.
Co-cluster Interactions: PDLF

Publishers: IP Domains

WebMedia e.g., usatoday

Online Portals (MSN, Yahoo)

Tech Companies

ISPs

![Heatmap](image.png)
Smoothing via Factorization

- Cluster size vary in PDLF, smoothing across local models works better

Row profile: \( u_i = (u_{i1}, u_{i2}, ..., u_{ir}) \); col profile: \( v_i = (v_{i1}, v_{i2}, ..., v_{ir}) \)

Regularized Weighted Factorization (RWF): 
\[
\delta_{ij} = u_i^T v_j; \quad u_i, v_i \text{ drawn from Gaussian prior}
\]

Squashed Matrix Factorization (SMF): 
Co-cluster and factorize cluster profiles 
\[
\delta_{ij} = U_i^T V_j
\]
Synthetic example (moderately sparse data)

Finer interactions

Missing values are shown as dark regions
Synthetic example (highly sparse data)
## Table 6.14. Prediction accuracy (5-fold cross-validation) on MovieLens dataset.

<table>
<thead>
<tr>
<th>Metric</th>
<th>RWF</th>
<th>COCLUST</th>
<th>SMF</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAE</td>
<td>0.8012 ± 0.0041</td>
<td>0.8481 ± 0.0082</td>
<td>0.7882 ± 0.0055</td>
</tr>
<tr>
<td></td>
<td>$r = 5$</td>
<td>$k = 5, l = 15$</td>
<td>$r = 5$</td>
</tr>
<tr>
<td>RMSE</td>
<td>0.3659 ± 0.0017</td>
<td>0.3676 ± 0.0021</td>
<td>0.3586 ± 0.0022</td>
</tr>
<tr>
<td></td>
<td>$r = 2$</td>
<td>$k = 10, l = 30$</td>
<td>$r = 5$</td>
</tr>
</tbody>
</table>
Estimating conversion rates

- Back to ip x publisher example
  - Now model conversion rates
    - Prob (click results in sales)
  - Detecting important interaction helps in traffic quality estimation

<table>
<thead>
<tr>
<th>Parameters</th>
<th>COCLUST</th>
<th>SMF</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSE</td>
<td>0.0406 ± 0.0011</td>
<td>0.0383 ± 0.0009</td>
</tr>
<tr>
<td>$R^2$</td>
<td>0.3485</td>
<td>0.4202</td>
</tr>
<tr>
<td>Parameters</td>
<td>$(k = 10, l = 500)$</td>
<td>$(k = 15, l = 750, r = 5)$</td>
</tr>
</tbody>
</table>

i.15. Prediction accuracy (with 5-fold cross-validation) on ip-click dataset.
Summary

- Covariate only models often fail to capture residual dependence for dyadic data
- Model based co-clustering attractive and scalable approach to estimate interactions
- Factorization on cluster effects smoothes local models; leads to better performance
- Models widely applicable in many scenarios
Ongoing work

• Fast co-clustering through DP mixtures (Richard Hahn, David Dunson)
  – Few sequential scans over the data
  – Initial results extremely promising

• Model based hierarchical co-clustering (Inderjit Dhillon)
  – Multi-resolution local models; smoothing by borrowing strength across resolutions