Large-scale Collaborative Prediction Using a Nonparametric Random Effects Model

Kai Yu†

Joint work with John Lafferty‡ and Shenghuo Zhu†

†NEC Laboratories America, ‡Carnegie Mellon University
Learning multiple tasks

For input vector $z_j$, and its outputs $Y_{ij}$ under various conditions (tasks), the standard regression model is

$$Y_{ij} = \mu + m_i(z_j) + \epsilon_{ij},$$

where $\epsilon_{ij} \overset{iid}{\sim} N(0, \sigma^2)$, $i = 1, \ldots, M$, and $j = 1, \ldots, N$. 
A kernel approach to multi-task learning

To model the dependency between tasks, a hierarchical Bayesian approach may assume

\[ m_i \overset{iid}{\sim} \text{GP}(0, \Sigma) \]

where

- \( \Sigma(z_j, z_{j'}) \succ 0 \) is a **shared covariance function among inputs**;
- Many multi-task learning approaches are similar to this.\(^a\)

\(^a\)ICML-09 tutorial, Tresp & Yu
Using task-specific features

Assuming task-specific features $x$ are available, a more flexible approach is to model the data jointly, as

$$Y_{ij} = \mu + m(x_i, z_j) + \epsilon_{ij},$$

where $\epsilon_{ij} \sim \text{N}(0, \sigma^2)$, $m_{ij} = m(x_i, z_j)$ is a relational function.
A nonparametric kernel-based approach

- Assume the relational function follows\(^a\)

\[ m \sim \text{GP}(0, \Omega \otimes \Sigma) \]

where

- \( \Omega(x_i, x_i') \) is a covariance function on tasks;
- \( \Sigma(z_j, z_j') \) is a covariance function on inputs;
- any sub matrix follows

\[ m \sim \text{N}(0, \Omega \otimes \Sigma) \Rightarrow \text{Cov}(m_{ij}, m_{i'j'}) = \Omega_{ii'} \Sigma_{jj'}; \]

- If \( \Omega = \delta \), the prior reduces to \( m_i \overset{\text{iid}}{\sim} \text{GP}(0, \Sigma). \)

\(^a\)Yu et al., 2007; Bonilla et al., 2008
The collaborative prediction problem

This essentially a multi-task learning problem with task features;
- Matrix factorization using additional row/column attributes;
- The formulation applies to many *relational prediction* problems.
Challenges to the kernel approach

- **Computation**: the cost $O(M^3N^3)$ is prohibitive.
  - Netflix data: $M = 480,189$ and $N = 17,770$.

- **Dependent “noise”**: when $Y_{ij}$ cannot be fully explained by the predictors $x_i$ and $z_j$, the conditional independence assumption is invalid, which means,

  \[ p(Y \mid m, x, z) \neq \prod_{i,j} p(Y_{ij} \mid m, x_i, z_j) \]

  - User and movie features are weak predictors;
  - The relational observations $Y_{ij}$ alone are informative to each other.
This work

- Novel multi-task model using both input and task attributes;
- Nonparametric random effects to resolve dependent “noises”;
- Efficient algorithm for large-scale collaborative prediction problems.
Nonparametric random effects

\[ Y_{ij} = \mu + m_{ij} + f_{ij} + \epsilon_{ij}, \]

- \( m(\mathbf{x}_i, \mathbf{z}_j) \): a function depending on known attributes;
- \( f_{ij} \): random effects for dependent “noises”;
  - modeling dependency in observations with repeated structures.
- Let \( f_{ij} \) be nonparametric: dimensionality increases with data size;
  - “nonparametric matrix factorization”
Efficiency considerations in modeling

- To save computation, we absorb $\epsilon$ into $f$ and obtain

$$Y_{ij} = \mu + m_{ij} + f_{ij},$$

- Introduce a special generative process for $m$ and $f$...

$$m, f \sim \cdot, \cdot | \Omega_0(x_i, x_i'), \Sigma_0(z_j, z_j'),$$
The row-wise generative model

\[ \Sigma \sim \text{IWP}(\kappa, \Sigma_0 + \lambda \delta) \]
The row-wise generative model

\[ \Sigma \sim \text{IWP}(\kappa, \Sigma_0 + \lambda \delta) \]

\[ m \sim \text{GP}(0, \Omega_0 \otimes \Sigma) \]

\[ m(x_i, z_j) \]
The row-wise generative model

\[ \Sigma \sim \text{IWP}(\kappa, \Sigma_0 + \lambda \delta) \]

\[ m \sim \text{GP}(0, \Omega_0 \otimes \Sigma) \]

\[ f_i \overset{\text{iid}}{\sim} \text{GP}(0, \tau \Sigma) \]
The row-wise generative model

\[ \Sigma \sim \text{IWP}(\kappa, \Sigma_0 + \lambda \delta) \]

\[ m \sim \text{GP}(0, \Omega_0 \otimes \Sigma) \]

\[ f_i \overset{\text{iid}}{\sim} \text{GP}(0, \tau \Sigma) \]
The row-wise generative model

\[ \Sigma \sim \text{IWP}(\kappa, \Sigma_0 + \lambda \delta) \]

\[ m \sim \text{GP}(0, \Omega_0 \otimes \Sigma) \]

\[ f_i \overset{\text{iid}}{\sim} \text{GP}(0, \tau \Sigma) \]
The row-wise generative model

\[ \Sigma \sim \text{IWP} (\kappa, \Sigma_0 + \lambda \delta) \]

\[ m \sim \text{GP} (0, \Omega_0 \otimes \Sigma) \]

\[ f_i \overset{\text{iid}}{\sim} \text{GP} (0, \tau \Sigma) \]
The row-wise generative model

\[ \Sigma \sim \text{IWP}(\kappa, \Sigma_0 + \lambda \delta) \]

\[ m \sim \text{GP}(0, \Omega_0 \otimes \Sigma) \]

\[ f_i \overset{\text{iid}}{\sim} \text{GP}(0, \tau \Sigma) \]
The row-wise generative model

\[ \Sigma \sim \text{IWP}(\kappa, \Sigma_0 + \lambda \delta) \]

\[ m \sim \text{GP}(0, \Omega_0 \otimes \Sigma) \]

\[ f_i \overset{\text{iid}}{\sim} \text{GP}(0, \tau \Sigma) \]

\[ Y_{ij} = \mu + m_{ij} + f_{ij} \]
The column-wise generative model

\[ \Omega \sim \text{IWP}(\kappa, \Omega_0 + \tau \delta) \]

\[ m \sim \text{GP}(0, \Omega \otimes \Sigma_0) \]

\[ f_j \overset{\text{iid}}{\sim} \text{GP}(0, \lambda \Omega) \]

\[ Y_{ij} = \mu + m_{ij} + f_{ij} \]
Two generative models

\[ Y_{ij} = \mu + m_{ij} + f_{ij}, \]

**column-wise model:**

\[ \Omega \sim \text{IWP}(\kappa, \Omega_0 + \tau \delta), \]
\[ m \sim \text{GP}(0, \Omega \otimes \Sigma_0), \]
\[ f_j \overset{\text{iid}}{\sim} \text{GP}(0, \lambda \Omega), \]

**row-wise model:**

\[ \Sigma \sim \text{IWP}(\kappa, \Sigma_0 + \lambda \delta), \]
\[ m \sim \text{GP}(0, \Omega_0 \otimes \Sigma), \]
\[ f_i \overset{\text{iid}}{\sim} \text{GP}(0, \tau \Sigma), \]
Two generative models are equivalent

- Both models lead to the same **matrix-variate Student-t process**
  \[ Y \sim \text{MTP}(\kappa, 0, (\Omega_0 + \tau\delta), (\Sigma_0 + \lambda\delta)), \]

- The model “learns” both \( \Omega \) and \( \Sigma \) simultaneously;

- Sometimes one model is computationally cheaper than the other.
An idea of large-scale modeling

Assuming $M \gg N$, we

- choose the row-wise model,
- let $\Omega_0(x_i, x_{i'})$ be low-rank.

$Y \sim \text{MT}(\kappa, 0, (\Omega_0 + \tau I_M), (\Sigma_0 + \lambda I_N)),$
Modeling large-scale data

- If $\Omega_0(x_i, x_{i'}) = \langle \phi(x_i), \phi(x_{i'}) \rangle$, $m \sim \text{GP}(0, \Omega_0 \otimes \Sigma)$ implies
  \[ m_{ij} = \langle \phi(x_i), \beta_j \rangle \]

- Without loss of generality, let $\Omega_0(x_i, x_{i'}) = \langle p_1^2 x_i, p_1^2 x_{i'} \rangle$, $x_i \in \mathbb{R}^p$. On a finite observational matrix $Y \in \mathbb{R}^{M \times N}$, $M \gg N$, the row-wise model becomes

  \[
  \begin{align*}
  \Sigma &\sim \text{IW}(\kappa, \Sigma_0 + \lambda I_N), \\
  \beta &\sim \text{N}(0, I_p \otimes \Sigma), \\
  Y_i &\sim \text{N}(\beta^\top x_i, \tau \Sigma), \quad i = 1, \ldots, M
  \end{align*}
  \]

  where $\beta$ is $p \times N$ random matrix.
Approximate Inference - EM

- E-step: compute the sufficient statistics \( \{\nu_i, C_i\} \) for the posterior of \( Y_i \) given the current \( \beta \) and \( \Sigma \):

\[
Q(Y) = \prod_{i=1}^{M} p(Y_i | Y_{O_i}, \beta, \Sigma) = \prod_{i=1}^{M} N(Y_i | \nu_i, C_i),
\]

- M-step: optimize \( \beta \) and \( \Sigma \):

\[
\hat{\beta}, \hat{\Sigma} = \arg \min_{\beta, \Sigma} \left\{ \mathbb{E}_{Q(Y)} [- \log p(Y, \beta, \Sigma | \theta)] \right\}
\]

and then let \( \beta \leftarrow \hat{\beta}, \Sigma \leftarrow \hat{\Sigma} \).
Some notation

- Let $J_i \subset \{1, \ldots, N\}$ be the index set of the $N_i$ observed elements in the row $Y_i$;

- $\Sigma_{[::J_i]} \in \mathbb{R}^{N \times N_i}$ is the matrix obtained by keeping the columns of $\Sigma$ indexed by $J_i$;

- $\Sigma_{[J_i::J_i]} \in \mathbb{R}^{N_i \times N_i}$ is obtained from $\Sigma_{[::J_i]}$ by further keeping only the rows indexed by $J_i$;

- Similarly, we can define $\Sigma_{[J_i::]}$, $Y_{[i,J_i]}$ and $m_{[i,J_i]}$. 
The EM algorithm

- E-step: for $i = 1, \ldots, M$

  $$
  \begin{align*}
  m_i &= \beta^\top x_i, \\
  \nu_i &= m_i + \Sigma_{[,]i} \Sigma_{[J_i,J_i]}^{-1} (Y_{[i,J_i]} - m_{[i,J_i]})^\top, \\
  C_i &= \tau \Sigma - \tau \Sigma_{[,]i} \Sigma_{[J_i,J_i]}^{-1} \Sigma_{[J_i,\cdot]}.
  \end{align*}
  $$

- M-step:

  $$
  \begin{align*}
  \hat{\beta} &= (x^\top x + \tau I_p)^{-1} x^\top \nu, \\
  \hat{\Sigma} &= \tau^{-1} \left[ \sum_{i=1}^M (C_i + \nu_i \nu_i) - \nu^\top x (x^\top x + \tau I_p)^{-1} x^\top \nu \right] + \Sigma_0 + \lambda I_N \\
  &= \frac{M + 2N + p + \kappa}{M + 2N + p + \kappa}.
  \end{align*}
  $$

On Netflix, each EM iteration takes several thousands of hours.
Let $U_i \in \mathbb{R}^{N \times N_i}$ be a column selection operator, such that $\Sigma[:,J_i] = \Sigma U_i$ and $\Sigma[J_i,J_i] = U_i^\top \Sigma U_i$.

The M-step only needs $C = \sum_{i=1}^M C_i + \nu^\top \nu$ and $\nu^\top x$ from the previous E-step. To obtain them, it’s unnecessary to compute $\nu_i$ and $C_i$. For example,

$$\sum_{i=1}^M C_i = \sum_{i=1}^M \left( \tau \Sigma - \tau \Sigma[:,J_i] \Sigma_{[J_i,J_i]}^{-1} \Sigma_{[J_i,:]} \right)$$

$$= \sum_{i=1}^M \left( \tau \Sigma - \tau \Sigma U_i \Sigma_{[J_i,J_i]}^{-1} U_i^\top \Sigma \right)$$

$$= \tau M \Sigma - \tau \Sigma \left( \sum_{i=1}^M U_i \Sigma_{[J_i,J_i]}^{-1} U_i^\top \right) \Sigma$$

Similar tricks can be applied to $\nu^\top \nu$ and $\nu^\top x$. Time for each iteration is reduced from thousands of hours to 5 hours only.
EachMovie Data

- 74424 users, 1648 movies;
- 2,811,718 numeric ratings $Y_{ij} \in \{1, \ldots, 6\}$;
- 97.17% of the elements are missing;
- Use 80% ratings of each user for training and the rest for testing;
- This random selection is repeated 10 times independently.
Compared methods

- **User Mean & Movie Mean**: prediction by the empirical mean;
- **FMMMF**: fast max-margin matrix factorization \(^a\);
- **PPCA**: probabilistic principal component analysis \(^b\);
- **BSRM**: Bayesian stochastic relational model \(^c\), BSRM-1 uses no additional user/movie attributes \(^d\);
- **NREM**: Nonparametric random effects model, NREM-1 uses no additional attributes.

\(^a\)Rennie & Srebro (2005).
\(^b\)Tipping & Bishop (1999).
\(^c\)Zhu, Yu, & Gong (2009).
\(^d\)Top 20 eigenvectors from of binary matrix indicating if a rating is observed or not.
**Results on EachMovie**

**TABLE: Prediction Error on EachMovie Data**

<table>
<thead>
<tr>
<th>Method</th>
<th>RMSE</th>
<th>Standard Error</th>
<th>Run Time (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Mean</td>
<td>1.4251</td>
<td>0.0004</td>
<td></td>
</tr>
<tr>
<td>Movie Mean</td>
<td>1.3866</td>
<td>0.0004</td>
<td></td>
</tr>
<tr>
<td>FMMMF</td>
<td>1.1552</td>
<td>0.0008</td>
<td>4.94</td>
</tr>
<tr>
<td>PPCA</td>
<td>1.1045</td>
<td>0.0004</td>
<td>1.26</td>
</tr>
<tr>
<td>BSRM-1</td>
<td>1.0902</td>
<td>0.0003</td>
<td>1.67</td>
</tr>
<tr>
<td>BSRM-2</td>
<td>1.0852</td>
<td>0.0003</td>
<td>1.70</td>
</tr>
<tr>
<td>NREM-1</td>
<td>1.0816</td>
<td>0.0003</td>
<td>0.59</td>
</tr>
<tr>
<td>NREM-2</td>
<td>1.0758</td>
<td>0.0003</td>
<td>0.59</td>
</tr>
</tbody>
</table>
Netflix Data

- 100, 480, 507 ratings from 480, 189 users on 17, 770 movies;
- \( Y_{ij} \in \{1, 2, 3, 4, 5\}; \)
- A set of validation data contain 1, 408, 395 ratings;
- Therefore there are 98.81% of elements missing in the rating matrix;
- The test set includes 2, 817, 131 ratings;
Compared methods

In addition to those compared in EachMovie experiment, there are several other methods:

- **SVD**: a method almost the same as FMMMF, using a gradient-based method for optimization\(^a\).
- **RBM**: Restricted Boltzmann Machine using contrast divergence\(^b\).
- **PMF** and **BPMF**: probabilistic matrix factorization\(^c\), and its Bayesian version\(^d\).
- **PMF-VB**: probabilistic matrix factorization using a variational Bayes method for inference\(^e\).

\(^a\)Kurucz, Benczur, & Csalogany, (2007).
\(^b\)Salakhutdinov, Mnih & Hinton (2007).
\(^c\)Salakhutdinov & Mnih (2008b).
\(^d\)Salakhutdinov & Mnih (2008a).
\(^e\)Lim & Teh (2007).
### Results on Netflix

**TABLE: Performance on Netflix Data**

<table>
<thead>
<tr>
<th>Method</th>
<th>RMSE</th>
<th>Run Time (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cinematch</td>
<td>0.9514</td>
<td>-</td>
</tr>
<tr>
<td>SVD</td>
<td>0.920</td>
<td>300</td>
</tr>
<tr>
<td>PMF</td>
<td>0.9265</td>
<td>-</td>
</tr>
<tr>
<td>RBM</td>
<td>0.9060</td>
<td>-</td>
</tr>
<tr>
<td>PMF-VB</td>
<td>0.9141</td>
<td>-</td>
</tr>
<tr>
<td>BPMF</td>
<td>0.8954</td>
<td>1100</td>
</tr>
<tr>
<td>BSRM-2</td>
<td>0.8881</td>
<td>350</td>
</tr>
<tr>
<td>NREM-1</td>
<td><strong>0.8876</strong></td>
<td>148</td>
</tr>
<tr>
<td>NREM-2</td>
<td><strong>0.8853</strong></td>
<td>150</td>
</tr>
</tbody>
</table>
Predictive Uncertainty

Standard deviations of prediction residuals vs. standard deviations predicted by our model on EachMovie
Related work

- Multi-task learning using Gaussian processes, those learn the covariance $\Sigma$ shared across tasks $^a$, and those that additionally consider the covariance $\Omega$ between tasks $^b$

- Application of GP models to collaborative filtering $^c$

- Low-rank matrix factorization, e.g., $^d$. Our model is nonparametric in the sense no rank constraint is imposed.

- Very few matrix factorization methods use known predictors. One such a work $^e$ introduces low-rank multiplicative random effects in modeling networked observations.

---

$^a$Lawrence & Platt (2004); Schwaighofer, Tresp & Yu (2004); Yu, Tresp & Schwaighofer (2005).

$^b$Yu, Chu, Yu, Tresp, & Xu, (2007); Bonilla, Chai, & Williams (2008).


$^e$Hoff (2005)
Summary

- The model provides a novel way to use random effects and known attributes to explain the complex dependence of data;

- We make the nonparametric model scalable and efficient on very large-scale problems;

- Our experiments demonstrate that the algorithm works very well on two challenging collaborative prediction problems;

- In the near future, it will be promising to perform a full Bayesian inference by a parallel Gibbs sampling method.