Our Solution on Track 1:
A Linear Ensemble of Individual and Blended Models for Music Rating Prediction

Po-Lung Chen, Chen-Tse Tsai, Yao-Nan Chen, Ku-Chun Chou, Chun-Liang Li, Cheng-Hao Tsai, Kuan-Wei Wu, Yu-Cheng Chou, Chung-Yi Li, Wei-Shih Lin, Shu-Hao Yu, Rong-Bing Chiu, Chieh-Yen Lin, Chien-Chih Wang, Po-Wei Wang, Wei-Lun Su, Chen-Hung Wu, Tsung-Ting Kuo, Todd G. McKenzie, Ya-Hsuan Chang, Chun-Sung Ferng, Chia-Mau Ni, Hsuan-Tien Lin, Chih-Jen Lin and Shou-De Lin

National Taiwan University
Three Properties of Track 1 Data

<table>
<thead>
<tr>
<th></th>
<th>track\textsubscript{1}</th>
<th>track\textsubscript{2}</th>
<th>album\textsubscript{3}</th>
<th>author\textsubscript{4}</th>
<th>\ldots</th>
<th>genre\textsubscript{i}</th>
</tr>
</thead>
<tbody>
<tr>
<td>user\textsubscript{1}</td>
<td>(100, t\textsubscript{11})</td>
<td>(80, t\textsubscript{12})</td>
<td>(70, t\textsubscript{13})</td>
<td>(?, t\textsubscript{14})</td>
<td>\ldots</td>
<td>--</td>
</tr>
<tr>
<td>user\textsubscript{2}</td>
<td>--</td>
<td>(0, t\textsubscript{22})</td>
<td>(?, t\textsubscript{23})</td>
<td>(80, t\textsubscript{24})</td>
<td>\ldots</td>
<td>--</td>
</tr>
<tr>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
</tr>
<tr>
<td>user\textsubscript{U}</td>
<td>(?, t\textsubscript{U1})</td>
<td>--</td>
<td>(20, t\textsubscript{U3})</td>
<td>--</td>
<td>\ldots</td>
<td>(0, t\textsubscript{UI})</td>
</tr>
</tbody>
</table>

similar to Netflix data, but with the following differences......

- **scale**: larger training and test sets
  - training: study mature models that are computationally feasible;
  - test: linearly combine many models w/o much overfitting

- **taxonomy**: relation graph of tracks, albums, authors and genres
  - include as features for combining models nonlinearly

- **time**: detailed; training earlier than validation earlier than test
  - include as features for combining models nonlinearly;
  - respect time-closeness during training & with val.-set blending
Selected Ideas that Did Not Work:
Deal with Zero-Variance Users

Background

- zero-variance users (7% of all users)
  —if a user gives 60, 60, 60, ... in all training ratings, how’d she rate the next item?

- Occam’s razor prediction: 60
  —only true for 80% of users, 20% changed their mind!

Idea

- conditionally (the 80%) post-process the predictions
- difficult to distinguish and thus failed
Framework of Our Solution

- single models—computationally feasible models that are diverse:
  - individual models: matrix factorization (& pPCA), pLSA
  - residual models: R-Boltz. machine, $k$-NN
  - derivative model: regression with statistical & model-based features

- validation-set blending:
  combine models nonlinearly while respecting time-closeness

- test-set blending:
  combine models linearly while fitting the leaderboard feedback

- post processing:
  polish predictions using findings during data analysis
single models: 22.7915
- individual models: best RMSE 22.9022 (MF)
- residual models: best RMSE 22.7915 ($k$-NN + MF)
- derivative model: best RMSE 24.1251 (but helps in later stages)

validation-set blending: 21.3598 [improvement 1.4317]
test-set blending: (estimated) 21.0253 [improvement 0.3345]
post processing: 21.0147 [improvement 0.0106]

both blending stages: key to the system
<table>
<thead>
<tr>
<th>model</th>
<th># used</th>
<th>best</th>
<th>average</th>
<th>worst</th>
<th>contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>MF</td>
<td>81</td>
<td>22.90</td>
<td>23.92</td>
<td>26.94</td>
<td><strong>0.3645</strong></td>
</tr>
<tr>
<td>pPCA</td>
<td>2</td>
<td>24.46</td>
<td>24.61</td>
<td>24.75</td>
<td>0.0014</td>
</tr>
<tr>
<td>pLSA</td>
<td>7</td>
<td>24.83</td>
<td>25.53</td>
<td>26.09</td>
<td>0.0042</td>
</tr>
<tr>
<td>R-Boltz. machine</td>
<td>8</td>
<td>22.80</td>
<td>24.75</td>
<td>26.08</td>
<td><strong>0.0314</strong></td>
</tr>
<tr>
<td>k-NN</td>
<td>18</td>
<td>22.79</td>
<td>25.06</td>
<td>42.94</td>
<td><strong>0.0298</strong></td>
</tr>
<tr>
<td>regression</td>
<td>10</td>
<td>24.13</td>
<td>28.01</td>
<td>35.14</td>
<td><strong>0.0261</strong></td>
</tr>
</tbody>
</table>

- **contribution (before val.-set blending):**
  - estimated RMSE diff. via leave-the-model-out in test-set blending
  - MF: most important (absorbing pPCA)
  - residual models: both quite important
  - derivative model: individually weak but adds diversity

**val.-set blending:**

95 models, best 21.36, average 23.53, worst 31.70
### Background

SGD for minimizing sum of per-example $E_n(\theta)$ (say, for MF):
- randomly pick one example $n$
- $\theta \leftarrow \theta - \eta \cdot \nabla E_n(\theta)$

### Idea

- last $M$ steps of SGD: effectively considering only the last $M$ examples picked—**final $\theta$ as if biased towards those**
- need: $\theta$ respects time-closeness to the test examples
- heuristic: deterministically pick the “newer” examples as last

**consistent $\approx 0.05$ RMSE improvement for MF**
Selected Ideas that Worked (2/5): Gaussian RBM as Residual Model

**Background**

- RBM: a recursive NNet; can be used as an individual model by discretely hidden factors

<table>
<thead>
<tr>
<th>discrete hidden factors</th>
<th>↑</th>
<th>↘</th>
</tr>
</thead>
<tbody>
<tr>
<td>per-user incomplete discrete ratings</td>
<td></td>
<td>predicted continuous ratings</td>
</tr>
</tbody>
</table>

- as individual: RMSE 24.7433, worse than MF (22.9974)

**Idea**

- MF (a first-order model) efficiently gets better performance, but can RBM digest something different?

- need: RBM that learns from the residuals of MF (continuous values)
Selected Ideas that Worked (2/5): Gaussian RBM as Residual Model

Background

<table>
<thead>
<tr>
<th>discrete hidden factors</th>
<th>↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>per-user incomplete discrete ratings</td>
<td>↑↓</td>
</tr>
<tr>
<td>predicted continuous ratings</td>
<td></td>
</tr>
</tbody>
</table>

Idea

- need: RBM that learns from the residuals of MF
- choice: Gaussian RBM (gRBM)

<table>
<thead>
<tr>
<th>discrete hidden factors</th>
<th>↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>per-user incomplete continuous residuals</td>
<td>↑↓</td>
</tr>
<tr>
<td>predicted continuous residuals</td>
<td></td>
</tr>
</tbody>
</table>

MF+gRBM: 22.8008;
better than individual MF (22.9974) or RBM (24.7433)
Background

- Binned Linear Regression: a conditional aggregation model
- different model strength on different “types” of examples
- different blending weights for different types (bins) to utilize strength

<table>
<thead>
<tr>
<th>bins</th>
<th># rating $\leq \theta_1$</th>
<th>$\theta_1 &lt; #$ rating $\leq \theta_2$</th>
<th>others</th>
</tr>
</thead>
<tbody>
<tr>
<td>weight of MF-1</td>
<td>0.4</td>
<td>0.7</td>
<td>1.0</td>
</tr>
<tr>
<td>weight of RBM-1</td>
<td>0.5</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>weight of RBM-2</td>
<td>0.1</td>
<td>0.2</td>
<td>0.0</td>
</tr>
</tbody>
</table>

- a simplified regression tree with one level (on one feature)
Selected Ideas that Worked (3/5): Multi-Feature and Multi-Stage Binned Lin. Reg.

Background
- Binned Linear Regression
  —different blending weights for different (types) bins of examples

Idea: multi-feature BLR
- rationale: “type” more sophisticated than 1-feature bin
- a special multi-level decision tree
- prevent overfitting by limiting height and bin size
- heuristic algorithm instead of traditional decision tree:
  due to simplicity by extending from one-feature BLR

<table>
<thead>
<tr>
<th>multi-feature</th>
<th>1-feature</th>
<th>4-feature</th>
<th>6-feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSE</td>
<td>22.0829</td>
<td>21.8605</td>
<td>21.8128</td>
</tr>
</tbody>
</table>
Selected Ideas that Worked (3/5):
Multi-Feature and Multi-Stage Binned Lin. Reg.

Background

- Binned Linear Regression
  —different blending weights for different (types) bins of examples

Idea: **multi-stage** BLR

- rationale: more diverse but good models before test-set blending

<table>
<thead>
<tr>
<th>bins</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>weight of MF-1</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>weight of RBM-1</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>weight of RBM-2</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>weight of BLR-1</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>weight of BLR-2</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>multi-stage</th>
<th>1-stage</th>
<th>2-stage</th>
<th>3-stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSE</td>
<td>21.7140</td>
<td>21.4591</td>
<td>21.4287</td>
</tr>
</tbody>
</table>
Selected Ideas that Worked (4/5): Offline Test Performance Predictor

Background

- given: columns $z_m = \text{test-set prediction of model } m$
- test-set linear regression:

$$w(z_1, z_2, \cdots, z_M, \lambda) = (Z^T Z + \lambda I)^{-1} Z^T r$$

- true ratings $r$ unknown but $z^T r$ can be estimated by

$$2z^T r = z^T z + r^T r - (z - r)^T (z - r)$$

$$\approx z^T z + N \cdot \text{RMSE}(0)^2 - N \cdot \text{RMSE}(z)^2$$

- common technique for RMSE ever since Netflix competition
Selected Ideas that Worked (4/5): Offline Test Performance Predictor

**Background**

\[
2z^T r = z^T z + r^T r - (z - r)^T(z - r) \\
\approx z^T z + N \cdot \text{RMSE}(0)^2 - N \cdot \text{RMSE}(z)^2
\]

**Idea**

- want: decide which \(z_m\)'s and \(\lambda\) to use
- restriction: one submission every eight hours
- solution: estimate RMSE of \(w\) without submitting more than \(z_m\)

\[
N \cdot \text{RMSE}(w)^2 = (Zw - r)^T(Zw - r) = w^T Z^T Z w - 2w^T Z^T r + r^T r
\]

compute the contribution of models; choose 221 from \(\approx 300\) models & decide \(\lambda = 10^{-6}\) offline
Selected Ideas that Worked (5/5): Clipping for Old Four-Star Days

Background
- some very different rating systems observed during data analysis:
  - four-star rating? \{0, 30, 50, 70, 90\}
  - five-star rating? \{0, 20, 40, 60, 80, 100\}
  - 100-point scale
- suspect changes in the user interface of Yahoo! Music

Idea
- existing: in five-star or 100-point scale, clip prediction to \([0, 100]\)
- new: for four-star, clip prediction to \([0, 90]\)
- what dates? \([3365, 5982]\) (7 years) or \([4281, 6170]\) (5 years)

\(\approx 0.02\) RMSE improvement on most models
NTU team: 1 class, 19 students, 3 TAs, 3 professors

shared techniques between two tracks:
- models: MF, k-NN, pLSA
- concept of diversity and blending
- taxonomy information (more for track 2)

special techniques in track 2:
- construct suitable learning problems and (new) models from raw data
- sample proper validation sets

special techniques in track 1:
- respect time-closeness
- blend deeply with validation set and broadly with test set
We truly thank

- organizers for designing a successful competition
- NTU EECS college and CSIE department for support

Thank you. Questions?