Combining Factorization Model and Additive Forest for Recommendation

Presenter: Tianqi Chen

Team ACMClass@SJTU

August 11, 2012
Team ACMClass@SJTU

- Original team name: undergrads
- Members are students from ACMClass in SJTU
- All members are undergraduates, except the presenter:)

![Team ACMClass@SJTU](image)
Outline

Overview of General Approach

Go Beyond Factorization Models

More Example Models used in Solution

Results and Conclusion
Overview of Our Solution

Modeling Approach
- Factorization Models
- Additive Forest

Rank Optimization

Combination

Final Solution

Incorporated Information:
- social network/action
- user age/gender
- item taxonomy
- timestamp...

Focus point of this presentation

One Joint Model, No Ensemble
Feature-based Matrix Factorization

\[ \hat{r}_{ui} = \left( \sum_{c \in C(u)} \alpha_c^{(u)} p_c \right)^T \left( \sum_{c \in C(i)} \beta_c^{(i)} q_c \right) + \sum_{c \in C(u,i)} \gamma_c^{(u,i)} g_c \]  

\( \Theta = \{ p, q, g \} \), trained via stochastic gradient descent

\( \alpha_c^{(u)} \): user feature of user \( u \): user social network/action, keyword/tag

\( \beta_c^{(i)} \): item feature weight of item(celebrity) \( i \): item taxonomy/network

\( \gamma_c^{(u,i)} \): global feature related to interaction between \( u \) and \( i \): user age/gender bias
Outline

Overview of General Approach

Go Beyond Factorization Models

More Example Models used in Solution

Results and Conclusion
Additive Forest

\[ \hat{r}_{ui} = \sum_{s=1}^{S} f_{s,\text{root}}(i,s)(x_u) \]  

- \[ x_u \]: property feature of user \( u \)
- \[ f_{s,\text{root}}(i,s) \]: function defined by a regression tree
- Learning via gradient boosting algorithm

- Forest 1
- Forest 2
- item 1
- item 2
- item k

item i: Kaifu LEE

Major=IT?

Occupation=Student?

Age<25?

Gender=Female?

Age>12?

Like Dislike
An Example of Additive Forest

Forest 1

\[ f_1 \]

- root: SIGKDD
- age < 20?
  - Yes: 0
  - No: 0.5

- Major = CS?
  - Yes: +1
  - No: 0

Forest 2

\[ f_2 \]

- root: SIGKDD
- Has User Tag Data Mining?
  - Yes: +2
  - No: 0

- root: Barbie
- is female?
  - Yes: +1
  - No: 0

Forest 2 is learned to complement Forest 1
## Factorization Model vs Additive Forest

<table>
<thead>
<tr>
<th></th>
<th>Factorization</th>
<th>Additive Forest</th>
</tr>
</thead>
<tbody>
<tr>
<td>handling of sparse matrix data</td>
<td>very well</td>
<td>capable, not very well</td>
</tr>
<tr>
<td>combination of different information</td>
<td>linear combina-</td>
<td>nonlinear composition</td>
</tr>
<tr>
<td></td>
<td>tion</td>
<td></td>
</tr>
<tr>
<td>handling of continuous property</td>
<td>need predefined</td>
<td>automatic segmentation</td>
</tr>
<tr>
<td></td>
<td>segmentation</td>
<td></td>
</tr>
<tr>
<td>model complexity control</td>
<td>regularization</td>
<td>feature selection, pruning</td>
</tr>
</tbody>
</table>

- Both models have their own advantages on different aspects.
- Understanding their properties and knowing when to use which one is very important.
Information Combination: User Social Network

Factorization Model

\[ \hat{r}_{ui} = \left( \frac{1}{\sqrt{|F(u)|}} \sum_{j \in F(u)} p_j \right)^T q_i \]

- \( F(u) \): follow set of \( u \)
- Linear combination

Additive Forest

- Condition composition
- Feature selection

Root: item i

Follow A?

Follow B?

Follow both A and B: \( p_A^T q_i + p_B^T q_i \)

Specific score for condition “Follow A and B”
Continuous Feature Handling: User Age

Factorization Model

\[
\hat{r}_{ui} = \mathbf{p}_u^T \mathbf{q}_i + W_{i,ag(u)} \tag{3}
\]

- \(ag(u)\): age segment index
- Require predefined partition

Additive Forest

Automatic find splitting point

root: item i

age < 17?

Yes

age < 10?

Yes

-1

No

No

+1

age partition points

\begin{align*}
W_{i,1} & : 10 \\
W_{i,2} & : 20 \\
W_{i,3} & : 30 \\
W_{i,4} & : age bias parameters
\end{align*}
## Factorization Model vs Additive Forest

<table>
<thead>
<tr>
<th></th>
<th>Factorization</th>
<th>Additive Forest</th>
</tr>
</thead>
<tbody>
<tr>
<td>handling of sparse matrix data</td>
<td>very well</td>
<td>capable, not very well</td>
</tr>
<tr>
<td>combination of different information</td>
<td>linear combina-</td>
<td>nonlinear composition</td>
</tr>
<tr>
<td></td>
<td>tion</td>
<td></td>
</tr>
<tr>
<td>handling of continuous property</td>
<td>need predefined segmentation</td>
<td>automatic segmentation</td>
</tr>
<tr>
<td>model complexity control</td>
<td>regularization</td>
<td>feature selection, pruning</td>
</tr>
</tbody>
</table>

- Both models have their own advantages on different aspect
- Understanding their properties and knowing when to use which one is very important
Outline

Overview of General Approach

Go Beyond Factorization Models

More Example Models used in Solution

Results and Conclusion
Time-aware Model

Traditional Time Bin Model

\[ \hat{r}'_{ui}(t) = \hat{r}_{ui} + b_{i, \text{binid}(t)} \]

- \( \text{binid}(t) \): time bin index of \( t \)

Our Time-aware Model

\[ \hat{r}'_{ui}(t) = \hat{r}_{ui} + \sum_{s=1}^{S} f_{s,i}(t) \]

- \( f_{s,i}(t) \): \( k \)-piece step function

Figure: Comparison of Two Temporal Models
User Sequential Pattern

\[ \hat{r}_{ui}'(t) = \hat{r}_{ui} + \sum_{s=1}^{S} f_{s}(x_{seq}) \]  

Features include in \( x_{seq} \):

- time difference between clicks
- average click speed of current user

Figure: Single Variable Pattern \[ \sum_{s=1}^{S} f_{s}(\Delta t) \]
Final Model

\[
\hat{r}_{ui} = \left( \sum_{c \in C(u)} \alpha_{c}^{(u)} p_c \right)^T \left( \sum_{c \in C(i)} \beta_{c}^{(i)} q_c \right) + \sum_{c \in C(u,i)} \gamma_{c}^{(u,i)} g_c + \sum_{s=1}^{S} f_{s,\text{root}(s,i)}(x_{ui})
\]

- Combination of all the factorization model and additive forest
- Boosting from result of factorization part
Outline

Overview of General Approach

Go Beyond Factorization Models

More Example Models used in Solution

Results and Conclusion
## Experiment Results

<table>
<thead>
<tr>
<th>ID</th>
<th>model</th>
<th>public</th>
<th>private</th>
<th>$\Delta_{public}$</th>
<th>$\Delta_{private}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>item bias</td>
<td>34.6%</td>
<td>34.0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1 + user follow/action</td>
<td>36.7%</td>
<td>35.8%</td>
<td>2.1%</td>
<td>1.8%</td>
</tr>
<tr>
<td>3</td>
<td>2 + user age/gender</td>
<td>38.0%</td>
<td>37.2%</td>
<td>1.3%</td>
<td>1.4%</td>
</tr>
<tr>
<td>4</td>
<td>3 + user tag/keyword</td>
<td>38.5%</td>
<td>37.6%</td>
<td>0.5%</td>
<td>0.4%</td>
</tr>
<tr>
<td>5</td>
<td>4 + item taxonomy</td>
<td>38.7%</td>
<td>37.8%</td>
<td>0.2%</td>
<td>0.2%</td>
</tr>
<tr>
<td>6</td>
<td>5 + time-aware model</td>
<td>39.0%</td>
<td>37.9%</td>
<td>0.3%</td>
<td>0.1%</td>
</tr>
<tr>
<td>7</td>
<td>6 + age/gender(forest)</td>
<td>39.1%</td>
<td>38.0%</td>
<td>0.1%</td>
<td>0.1%</td>
</tr>
<tr>
<td>8</td>
<td>7 + sequential patterns</td>
<td>44.2%</td>
<td>42.7%</td>
<td>5.1%</td>
<td>4.7%</td>
</tr>
</tbody>
</table>

### Table: MAP@3 of different methods

- **User Modeling** and **Sequential Patterns** contributes the most
- **Time-aware model** is more effective in public data
- **All of them** are important for winning
Summary

- Seems Ensemble methods **do not** work in our experiment
- Choose right methods to utilize different kinds of data
  - Factorization models are powerful, but also have drawbacks
  - Additive forest can automatic cut the continuous features, sometimes smarter than human
- Use automatic cutting to build robust time-aware model
- Fully utilize the available information
- Source code: [svdfeature.apexlab.org](http://svdfeature.apexlab.org)
Thank You, Questions?
The rest parts of the slides are appendix
Objective Function

- Loss function of Pairwise Ranking: AUC optimization
  \[
  L_u = \frac{1}{|\{(i,j) | r_{ui} > r_{uj}\}|} \sum_{(i,j): r_{ui} > r_{uj}} C(\hat{r}_{ui} - \hat{r}_{uj})
  \] (6)

- Pseudo loss function of LambdaRank: MAP optimization
  \[
  L_u = \frac{1}{|\{(i,j) | r_{ui} > r_{uj}\}|} \sum_{(i,j): r_{ui} > r_{uj}} |\Delta_{ij} MAP| C(\hat{r}_{ui} - \hat{r}_{uj})
  \] (7)
  - \(\Delta_{ij} MAP\) is MAP change when we swap \(i\) and \(j\) in current list
  - \(C(x)\) is a surrogate convex loss function
    - logistic loss (BPR): \(C(x) = \ln(1 + e^{-x})\)
    - hinge loss (maximum margin): \(C(x) = \max(0, 1 - x)\)
  - \(L_u\) is normalized by number of pairs: \(|\{(i,j) | r_{ui} > r_{uj}\}|\).
  - Balance over all users is important.
BiLinear Model

\[ \hat{r}_{ui} = x_u^T W y_i \] (8)

- \( W \): weight matrix
- \( x_u \): property vector of user \( u \)
- \( y_i \): property vector of item \( i \)

Example: Social aware Model

\[ \hat{r}_{ui} = \frac{1}{\sqrt{|F(u)|}} \sum_{c \in F(u)} W_{c \rightarrow i}, \quad x_{uc} = \begin{cases} \frac{1}{\sqrt{|F(u)|}} & c \in F(u) \\ 0 & c \notin F(u) \end{cases}, \quad y_{uc} = e_i \] (9)

- \( W_{c \rightarrow i} \): confidence of rule \( u \) follows \( c \rightarrow u \) accept \( i \)
Feature-based matrix factorization can be viewed as a \textit{factorized} version of bilinear model.

- Advantage of $W$: direct modeling effect of $c \rightarrow i$
- Advantage of $P^TQ$: less parameter, topic level matching
  - When $W$ is large and with sparse data support, use factorization
  - When $W$ is small and with dense data support, use bilinear
User Social Network and Action

\[ \hat{r}_{ui} = \left( \frac{1}{\sqrt{|F(u)|}} \sum_{j \in F(u)} p_j + \frac{1}{\|\alpha_u\|_2} \sum_{j \in A(u)} \alpha_{u,j} y_j \right)^T q_i + b_i \]  

- \( F(u) \): set of items user \( u \) followed
- \( A(u) \): set of items user \( u \) has action with
- \( \alpha_u \): weight by action count
Item Taxonomy and Social Network

Taxonomy

\[ q_i' = q_i + q_{c1}(i) + q_{c2}(i) + q_{c3}(i) + q_{c4}(i) \]  (11)

- Taxonomy aware parameter sharing
- \( c^k(i) \): k-th level category of item \( i \) belongs to

Social Network

\[ q_i' = q_i + \sum_{j \in cofollow(i)} q_j \]  (12)