Discrete Factorization Machines for Fast Feature-based Recommendation

Han Liu\textsuperscript{1}, Xiangnan He\textsuperscript{2}, Fuli Feng\textsuperscript{2}, Liqiang Nie\textsuperscript{1}, Rui Liu\textsuperscript{3}, Hanwang Zhang\textsuperscript{4}

1. Shandong University
2. National University of Singapore
3. University of Electronic Science and Technology of China
4. Nanyang Technological University
Motivation

Accurate Recommender System

Quality of Service & Profit of the Service Provider

side information

ccontent-based: e.g., item descriptions

ccontext-based: e.g., when and where a purchase is made

csession-based: e.g., recent browsing history of users

user

item
Factorization Machines (FM)

FM is a score prediction function for a (user, item) pair feature $\mathbf{x}$.

$x \in \mathbb{R}^n$ is a feature representation of the side-information, concatenated one-hot user ID one-hot item ID side-information

Model bias parameter $w_0$ + $\sum_{i=1}^{n} w_i x_i$ + $\sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle v_i, v_j \rangle x_i x_j$

$\langle v_i, v_j \rangle$ models the interaction between the $i$-th and $j$-th feature dimensions.

FM models the interaction between each pair of nonzero features.
Motivation

Existing FM framework is not suitable for fast recommendation, especially for mobile users.
Discrete Factorization Machines

**Storing:**
- Easily Store: binary codes
- Impossible: real-valued vector

**Computing:**
- XOR Bit Operations: \( DFM(x) := w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle b_i, b_j \rangle x_i x_j. \)
Solution with the Constraints

**Observed score**

$$\arg\min_{w_0, \mathbf{w}, \mathbf{B}} \sum_{(x, y) \in \mathcal{V}} y - w_0 - \sum_{i=1}^{n} w_i x_i - \sum_{i=1}^{n} \sum_{j=i+1}^{n} (b_i, b_j) x_i x_j$$

$$+ \alpha \sum_{i=1}^{n} w_i^2,$$ s.t. $\mathbf{B} \in \{\pm 1\}^{k \times n}$, $\mathbf{B} \mathbf{1} = 0$, $\mathbf{B} \mathbf{B}^T = n \mathbf{I}$

**Binary codes**

**Balance Constraint:** each bit should split the dataset evenly

**De-Correlation Constraint:** each bit should be as independent as possible

However, the hard constraints of zero-mean and orthogonality may not be satisfied in Hamming space!
Our DFM Formulation

Objective Function:

\[
\arg \min_{w_0, w, B} \sum_{(x, y) \in \mathcal{V}} \left( y - w_0 - \sum_{i=1}^{n} w_i x_i - \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle b_i, b_j \rangle x_i x_j \right)^2 + \alpha \sum_{i=1}^{n} w_i^2 + \beta \| B - D \|^2
\]

Score Prediction

Constraint Trade-off

Binary Constraint:

\[
B \in \{ \pm 1 \}^{k \times n}
\]

Delegate Code

\[
D1 = 0, \quad DD^T = nI
\]

Quality Constraint:

Balance Constraint

De-correlation Constraint
Our Solution: Alternating Optimization

Alternative Procedure

B-Subproblem

\[
\arg \min_{\mathbf{B}} \sum_{(x,y) \in \mathcal{V}} (y - w_0 - \sum_{i=1}^{n} w_i x_i - \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{b}_i, \mathbf{b}_j \rangle x_i x_j)^2 - 2\beta tr(\mathbf{B}^T \mathbf{D}), \text{ s.t. } \mathbf{B} \in \{\pm 1\}^{k \times n}
\]

D-Subproblem

\[
\arg \max_{\mathbf{D}} tr(\mathbf{B}^T \mathbf{D}), \text{ s.t. } \mathbf{D} \mathbf{1} = 0, \mathbf{D} \mathbf{D}^T = m \mathbf{I}.
\]

w-Subproblem

\[
\arg \min_{w_0, \mathbf{w}} \sum_{(x,y) \in \mathcal{V}} (\phi - w_0 - \sum_{i=1}^{n} w_i x_i)^2 + \alpha \sum_{i=1}^{n} w_i^2, \quad \phi = y - \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{b}_i, \mathbf{b}_j \rangle x_i x_j
\]
B-Subproblem for Binary Codes

Objective Function

\[
\arg \min_B \sum_{(x,y) \in \mathcal{V}} (y - w_0 - \sum_{i=1}^{n} w_i x_i - \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle b_i, b_j \rangle x_i x_j)^2 - 2\beta tr(B^T D), \quad \text{s.t. } B \in \{\pm 1\}^{k \times n}
\]

For each feature code \( b \downarrow i \), optimize bit by bit

\[
\text{for } r = 1 \text{ to } n \text{ do } \\
\quad \text{repeat } \\
\quad \quad \text{for } t = 1 \text{ to } k \text{ do } \\
\quad \quad \quad \hat{b}_{rt} = \sum_{r} (x_r \psi - x_r^2 \hat{x}^T Z_t b_{rt}) \hat{x}^T z_t + \beta d_{rt}; \\
\quad \quad \quad b_{rt} \leftarrow \text{sgn}(K(\hat{b}_{rt}, b_{rt})) \\
\quad \text{end } \\
\quad \text{until converge; } \\
\text{end }
\]
D-Subproblem for Code Delegate

Objective Function

$$\arg\max_D tr(B^T D), \text{ s.t. } D1 = 0, DD^T = nI$$

Small SVD $k \times n$

$$([\hat{P}\hat{P}], Q) \leftarrow \text{SVD}([\overline{B}])$$

Orthogonalization

$$\hat{Q} \leftarrow \text{GramSchmidt}([Q1])$$

$$D \leftarrow \sqrt{n} [P \hat{P}] [Q \hat{Q}]^T$$

$k \times n$ row-centered code matrix
w-Subproblem for Bias

Objective Function

\[
\arg\min_{\mathbf{w}, w_0} \sum_{(x, y) \in \mathcal{V}} \left( \phi - w_0 - \sum_{i=1}^{n} w_i x_i \right)^2 + \alpha \sum_{i=1}^{n} w_i^2, \quad \phi = y - \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{b}_i, \mathbf{b}_j \rangle x_i x_j
\]

It is the standard multivariate linear regression problem, use Coordinate Descent algorithm
Experiment Settings

• Datasets:

<table>
<thead>
<tr>
<th>Datasets</th>
<th>#users</th>
<th>#items</th>
<th>#ratings</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yelp</td>
<td>13,679</td>
<td>12,922</td>
<td>640,143</td>
<td>0.36%</td>
</tr>
<tr>
<td>Amazon</td>
<td>35,151</td>
<td>33,195</td>
<td>1,732,060</td>
<td>0.15%</td>
</tr>
</tbody>
</table>

• Split: randomly split 50% training and 50% testing move items in the testing set that haven’t occurred in the training set to the training set.

• Evaluation Protocol: rank the testing items of a user and evaluate the ranked list with **NDCG@K**
Compared to the state-of-the-art

- **libFM**: Factorization Machines with *libFM* [Rendle et al., TIST’12] original implementation of FM
- **DCF**: Discrete Collaborative Filtering [Zhang et al., SIGIR’16] CF+binarization+direct optimization
- **DCMF**: Discrete Content-aware Matrix Factorization [Lian et al., KDD’17] CF+binarization+direct optimization+constraint
- **BCCF**: Binary Code learning for Collaborative Filtering [Zhou&Zha, KDD’12] MF+binarization+two-stage optimization
Performance Comparison

In figure, we show the recommendation performance (NDCG@1 to NDCG@10) of DFM and the baseline methods on the two datasets. The code length varies from 8 to 64.
Efficiency Study

Efficiency comparison between DFM and libFM regarding Testing Time Cost (TTC) on the two datasets.

### Yelp

<table>
<thead>
<tr>
<th>Code Length</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>libFM (TTC)</strong></td>
<td>27.18</td>
<td>56.77</td>
<td>114.10</td>
<td>217.64</td>
</tr>
<tr>
<td><strong>DFM (TTC)</strong></td>
<td>2.06</td>
<td>3.56</td>
<td>6.60</td>
<td>12.43</td>
</tr>
<tr>
<td><strong>Acceleration Ratio</strong></td>
<td>13.19</td>
<td>15.95</td>
<td>17.29</td>
<td>17.51</td>
</tr>
</tbody>
</table>

### Amazon

<table>
<thead>
<tr>
<th>Code Length</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>libFM (TTC)</strong></td>
<td>177.03</td>
<td>357.46</td>
<td>716.83</td>
<td>1,414.67</td>
</tr>
<tr>
<td><strong>DFM (TTC)</strong></td>
<td>12.67</td>
<td>22.50</td>
<td>42.56</td>
<td>81.04</td>
</tr>
<tr>
<td><strong>Acceleration Ratio</strong></td>
<td>13.97</td>
<td>15.89</td>
<td>16.84</td>
<td>17.46</td>
</tr>
</tbody>
</table>

DFM is an operable solution for many large-scale Web service to reduce the computation cost of their recommender systems.
Conclusion & Future Work

• We propose DFM to enable fast feature-based recommendation.
• We develop an efficient algorithm to address the challenging optimization problem of DFM.
• We will extend binary technique to neural recommender models such as Neural FM.
Thank you.

https://github.com/hanliu95/DFM