Using Ratings & Posters for Anime & Manga Recommendations

Jill-Jênn Vie

August 31, 2017
Recommendation System

Problem

- Every user rates few items (1%)
- How to infer missing ratings?

<table>
<thead>
<tr>
<th></th>
<th>Movie 1</th>
<th>Movie 2</th>
<th>Movie 3</th>
<th>Movie 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Ana</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Elain</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Sulman</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>
Every supervised machine learning algorithm

$$\text{fit}(X, y)$$

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$X$</td>
<td>$y$</td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>work_id</td>
<td>rating</td>
</tr>
<tr>
<td>24</td>
<td>823</td>
<td>like</td>
</tr>
<tr>
<td>12</td>
<td>823</td>
<td>dislike</td>
</tr>
<tr>
<td>12</td>
<td>25</td>
<td>favorite</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

$$\hat{y} = \text{predict}(X)$$

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$X$</td>
<td>$\hat{y}$</td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>work_id</td>
<td>rating</td>
</tr>
<tr>
<td>24</td>
<td>25</td>
<td>?disliked</td>
</tr>
<tr>
<td>12</td>
<td>42</td>
<td>?liked</td>
</tr>
</tbody>
</table>
Evaluation: Root Mean Squared Error (RMSE)

If I predict $\hat{y}_i$ for each user-work pair to test among $n$, while truth is $y_i^*$:

$$RMSE(\hat{y}, y^*) = \sqrt{\frac{1}{n} \sum_{i} (\hat{y}_i - y_i^*)^2}.$$
Dataset 1: Movielens

- 700 users
- 9000 movies
- 100000 ratings
Dataset 2: Mangaki

- 2100 users
- 15000 works
- 310000 ratings
- User can rate anime or manga
- And receive recommendations
- Also reorder their watchlist

- Code is 100% on GitHub
- Awards from Microsoft and Kokusai Kōryū Kikin
- Ongoing data challenge on universityofbigdata.net
KNN → measure similarity

*K*-nearest neighbors

- $R_u$ represents the row vector of user $u$ in the rating matrix (users × works).
- Similarity score between users (cosine):

$$score(u, v) = \frac{R_u \cdot R_v}{||R_u|| \cdot ||R_v||}.$$  

- Let’s identify the $k$-nearest neighbors of user $u$
- And recommend to user $u$ what $u$’s neighbors liked but $u$ didn’t see

**Hint**

If $R'$ the $N \times M$ matrix of rows $\frac{R_u}{||R_u||}$, we can get the $N \times N$ score matrix by computing $R' R'^T$. 
PCA, SVD → reduce dimension to generalize

Matrix factorization

\[ R = \begin{pmatrix} R_1 \\ R_2 \\ \vdots \\ R_n \end{pmatrix} = \begin{pmatrix} C \\ P \end{pmatrix} \]

Each row \( R_u \) is a linear combination of profiles \( P \).

**Interpreting Key Profiles**

If \( P \)

- \( P_1 \): adventure
- \( P_2 \): romance
- \( P_3 \): plot twist

And \( C_u \)

- \( 0,2 \)
- \( -0,5 \)
- \( 0,6 \)

\( \Rightarrow u \) likes a bit adventure, hates romance, loves plot twists.

**Singular Value Decomposition**

\[ R = (U \cdot \Sigma) V^T \]

where \( U : N \times r \) et \( V : M \times r \) are orthogonal and \( \Sigma : r \times r \) is diagonal.
Visualizing first two columns of $V_j$ in SVD

Closer points mean similar taste
Find your taste by plotting first two columns of $U_i$

You will like movies that are close to you.
Variants of Matrix Factorization

\( R \) ratings, \( C \) coefficients, \( P \) profiles (\( F \) features).

\[ R = CP = CF^T \Rightarrow r_{ij} \approx \hat{r}_{ij} \triangleq C_i \cdot F_j. \]

Objective functions (reconstruction error) to minimize

\[
\begin{align*}
\text{SVD} & : \sum_{i,j} (r_{ij} - C_i \cdot F_j)^2 \quad \text{(deterministic)} \\
\text{ALS} & : \sum_{i,j} \text{known} (r_{ij} - C_i \cdot F_j)^2 \\
\text{ALS-WR} & : \sum_{i,j} \text{known} (r_{ij} - C_i \cdot F_j)^2 + \lambda (\sum_i N_i \|C_i\|^2 + \sum_j M_j \|F_j\|^2) \\
\text{WALS by Tensorflow™} & : \sum_{i,j} w_{ij} \cdot (r_{ij} - C_i \cdot F_j)^2 + \lambda (\sum_i \|C_i\|^2 + \sum_j \|F_j\|^2)
\end{align*}
\]

Who do you think wins?
About the Netflix Prize

- On October 2, 2006, Netflix organized an online contest: *The first one who beats our algorithm (Cinematch) by more than 10% will receive 1,000,000 USD.*
- and gave anonymized data
- Half of world AI community suddenly became interested in this problem
- October 8, someone beat Cinematch
- October 15, 3 teams beat it, notably by 1.06%
- June 26, 2009, team 1 beat Cinematch by 10.05%
  - → *last call*: still one month to win
- July 25, 2009, team 2 beat Cinematch by 10.09%
- Team 1 does 10.09% also
- 20 minutes later team 2 does 10.10%
- ... Actually, both teams were *ex æquo* on the validation set
- ... So the first team to send their results won (team 1, 10.09%)
Privacy concerns

- August 2009, Netflix wanted to restart a contest
- Meanwhile, in 2007 two researchers from Texas University could de-anonymize users by crossing data with IMDb (approximate birth year, zip code, watched movies)
- In December 2009, 4 Netflix users sued Netflix
- March 2010, amicable settlement (enmankaikeitsu) → complaint is closed
ALS for feature extraction

\[ R = CP \]

Issue: Item Cold-Start

- If no ratings are available for an anime
  \[ \Rightarrow \] no feature will be trained
- If anime features are put to 0
  \[ \Rightarrow \] prediction of ALS will be constant for every unrated anime.

But we have posters!

- On Mangaki, almost all works have a poster
- How to extract information?
Illustration2Vec (Saito and Matsui, 2015)

- CNN pretrained on ImageNet, trained on Danbooru (1.5M illustrations with tags)
- 502 most frequent tags kept, outputs tag weights

<table>
<thead>
<tr>
<th>#</th>
<th>General Tag</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1girl</td>
<td>86.1%</td>
</tr>
<tr>
<td>2</td>
<td>thighhighs</td>
<td>84.0%</td>
</tr>
<tr>
<td>3</td>
<td>solo</td>
<td>79.2%</td>
</tr>
<tr>
<td>4</td>
<td>red hair</td>
<td>73.1%</td>
</tr>
<tr>
<td>5</td>
<td>long hair</td>
<td>66.4%</td>
</tr>
<tr>
<td>6</td>
<td>breasts</td>
<td>53.7%</td>
</tr>
<tr>
<td>7</td>
<td>gloves</td>
<td>38.0%</td>
</tr>
<tr>
<td>8</td>
<td>weapon</td>
<td>34.0%</td>
</tr>
<tr>
<td>9</td>
<td>elbow gloves</td>
<td>28.3%</td>
</tr>
<tr>
<td>10</td>
<td>high heels</td>
<td>14.0%</td>
</tr>
<tr>
<td>11</td>
<td>tattoo</td>
<td>10.9%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#</th>
<th>Character Tag</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#</th>
<th>Copyright Tag</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#</th>
<th>Rating</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>safe</td>
<td>68.4%</td>
</tr>
<tr>
<td>2</td>
<td>questionable</td>
<td>29.3%</td>
</tr>
<tr>
<td>3</td>
<td>explicit</td>
<td>1.92%</td>
</tr>
</tbody>
</table>
LASSO for explanation of user preferences

$T$ matrix of 15000 works $\times$ 502 tags

- Each user is described by its preferences $P$
  - $\rightarrow$ a sparse row of weights over tags.
- Estimate user preferences $P$ such that $r_{ij} \approx PT^T$.

Interpretation and explanation

- You seem to like magical girls but not blonde hair
  $\Rightarrow$ Look! All of them are brown hair! Buy now!

Least Absolute Shrinkage and Selection Operator (LASSO)

$$\frac{1}{2N_i} \| R_i - P_i T^T \|_2^2 + \alpha \| P_i \|_1.$$ 

where $N_i$ is the number of items rated by user $i$. 
Blending

We would like to do:

\[ \hat{r}_{ij}^{BALSE} = \begin{cases} \hat{r}_{ij}^{ALS} & \text{if item } j \text{ was rated at least } \gamma \text{ times} \\ \hat{r}_{ij}^{LASSO} & \text{otherwise} \end{cases} \]

But we can’t. Why?

\[ \hat{r}_{ij}^{BALSE} = \sigma(\beta(R_j - \gamma))\hat{r}_{ij}^{ALS} + (1 - \sigma(\beta(R_j - \gamma)))\hat{r}_{ij}^{LASSO} \]

where \( R_j \) denotes the number of ratings of item \( j \)

\( \beta \) and \( \gamma \) are learned by stochastic gradient descent.

We call this gate the Steins;Gate.
Blended Alternate Least Squares with Explanation

We call this model **BALSE**.
## Results

<table>
<thead>
<tr>
<th></th>
<th>Test set</th>
<th>1000 least rated (1.5%)</th>
<th>Cold-start items</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALS</td>
<td>1.157</td>
<td>1.299</td>
<td>1.493</td>
</tr>
<tr>
<td>LASSO</td>
<td>1.446</td>
<td>1.347</td>
<td>1.358</td>
</tr>
<tr>
<td>BALSE</td>
<td><strong>1.150</strong></td>
<td><strong>1.247</strong></td>
<td><strong>1.316</strong></td>
</tr>
</tbody>
</table>
Thank you!

Read this article
http://jiji.cat/bigdata/balse.pdf (soon on arXiv)

Compete to Mangaki Data Challenge
research.mangaki.fr (problem + University of Big Data)

Reproduce our results on GitHub
github.com/mangaki

Follow us on Twitter: @MangakiFR