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?

				
Bob	**4**	3	5	**1**
Ana	**2**	4	**2**	5
Elain	5	**4**	4	**3**
Sulman	2	**3**	1	4

Every supervised machine learning algorithm

fit(X, y)

X		y
`user_id`	`work_id`	`rating`
24	823	like
12	823	dislike
12	25	favorite
...	...	...

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

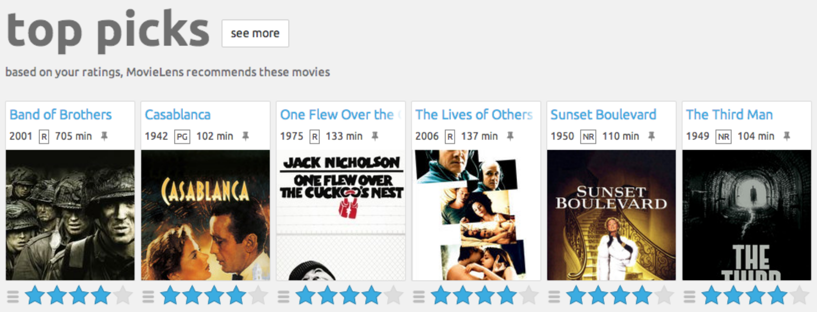
X		$\hat{y}$
`user_id`	`work_id`	`rating`
24	25	?disliked
12	42	?liked

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^*:

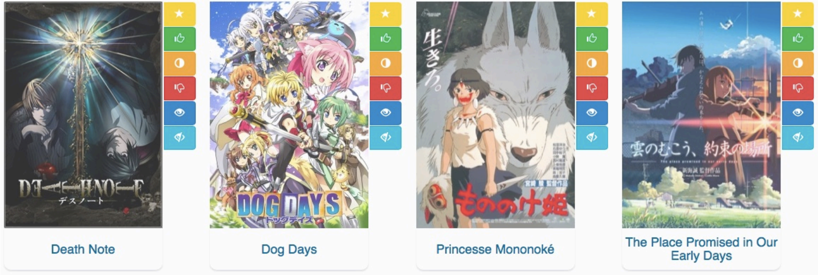
$$RMSE(\hat{y}, y^*) = \sqrt{\frac{1}{n} \sum_i (\hat{y}_i - y_i^*)^2}.$$

Dataset 1: Movielens



- 700 users
- 9000 movies
- 100000 ratings

Dataset 2: Mangaki



- 2100 users
- 15000 works *anime / manga / OST*
- 310000 ratings *fav / like / dislike / neutral / willsee / wontsee*
- 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 $\rightarrow$ measure similarity

K-nearest neighbors

- $\mathcal{R}_u$ represents the row vector of user u in the rating matrix (users $\times$ works).
- Similarity score between users (cosine):

$$score(u, v) = \frac{\mathcal{R}_u \cdot \mathcal{R}_v}{||\mathcal{R}_u|| \cdot ||\mathcal{R}_v||}.$$

- 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 $\frac{\mathcal{R}_u}{||\mathcal{R}_u||}$, we can get the $N \times N$ score matrix by computing $R'R'^T$.

PCA, SVD $\rightarrow$ reduce dimension to generalize

Matrix factorization

$$R = \begin{pmatrix} \mathcal{R}_1 \\ \mathcal{R}_2 \\ \vdots \\ \mathcal{R}_n \end{pmatrix} = \boxed{} = \boxed{C}\,\boxed{P}$$

Each row $\mathcal{R}_u$ 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

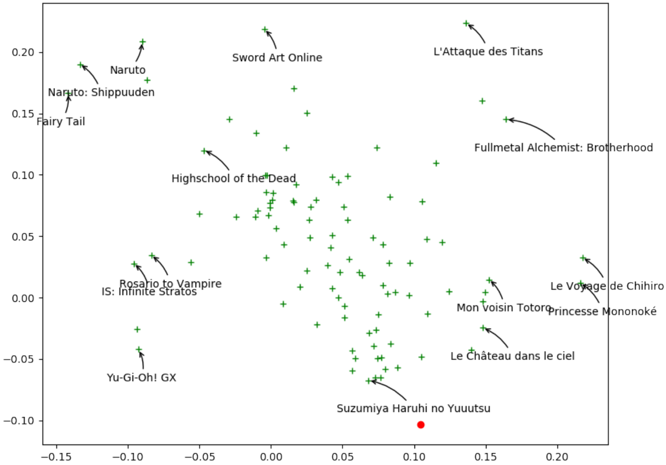
$\Rightarrow$ u likes a bit adventure, hates romance, loves plot twists.

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.

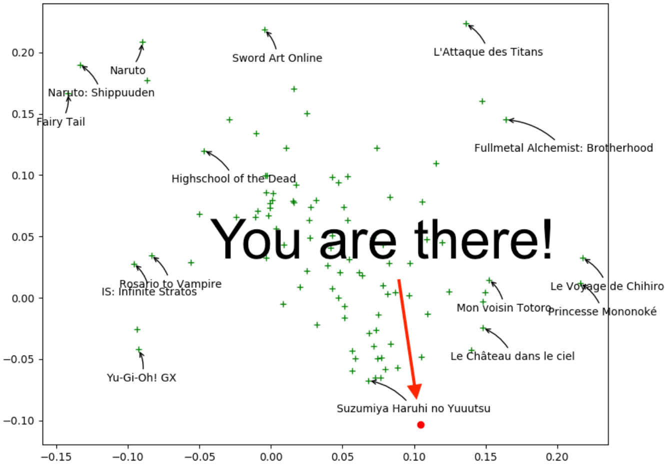
Visualizing first two columns of V_j in SVD

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).

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

Objective functions (reconstruction error) to minimize

SVD : $\sum_{i,j} \ (r_{ij} - C_i \cdot F_j)^2$ (deterministic)

ALS : $\sum_{i,j \text{ known}} \ (r_{ij} - C_i \cdot F_j)^2$

ALS-WR : $\sum_{i,j \text{ known}} \ (r_{ij} - C_i \cdot F_j)^2 + \lambda(\sum_i N_i ||C_i||^2 + \sum_j M_j ||F_j||^2)$

WALS by Tensorflow™ :

$$\sum_{i,j} w_{ij} \cdot (r_{ij} - C_i \cdot F_j)^2 + \lambda(\sum_i ||C_i||^2 + \sum_j ||F_j||^2)$$

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%
  $\rightarrow$ 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 (*enmankaiketsu*)
  $\rightarrow$ 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 at 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)

Prediction results

#	General Tag	Confidence
1.	1girl	86.1%
2.	thighhighs	84.0%
3.	solo	79.2%
4.	red hair	73.1%
5.	long hair	66.4%
6.	breasts	53.7%
7.	gloves	38.0%
8.	weapon	34.0%
9.	elbow gloves	28.3%
10.	high heels	14.0%
11.	tattoo	10.9%
#	**Character Tag**	
#	**Copyright Tag**	
#	**Rating**	
1.	safe	68.4%
2.	questionable	29.3%
3.	explicit	1.92%

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

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 $r_{ij} \simeq PT^T$.

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)

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

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

Blending

We would like to do:

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

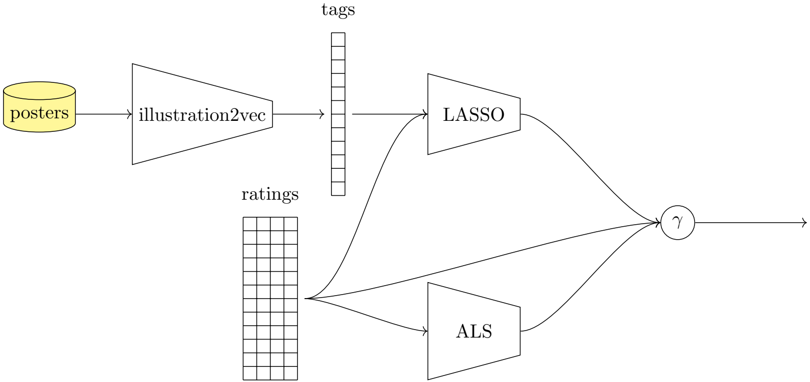
But we can't. Why?

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

where R_j denotes the number of ratings of item j
β and γ 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

RMSE	Test set	1000 least rated (1.5%)	Cold-start items
ALS	1.157	1.299	1.493
LASSO	1.446	1.347	1.358
BALSE	**1.150**	**1.247**	**1.316**

Thank you!

MANGAKI
LA RECOMMANDATION FRANCOPHONE D'ANIME ET MANGA

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