Master IASD
Data Science Lab
Explicit collaborative filtering

lecun-team:
Linghao Zeng, Zhe Huang, Joseph Amigo

October 2023

Table of content

Introduction

Problem statement

- n users, m movies
- $R^{\text{train}} = (r_{ij})_{ij} \in \mathcal{M}_{n,m}(\mathbb{R}),\ \forall i,j,\ \ r_{ij} \in [0,5] \cup \{\varnothing\}$
- $R^{\text{test}} = (\hat{r}_{ij})_{ij} \in \mathcal{M}_{n,m}(\mathbb{R}),\ \forall i,j,\ \ \hat{r}_{ij} \in [0,5] \cup \{\varnothing\}$
- Goal : Infer ratings of a user for a movie in R^{test} with $\{$ratings of other users for this movie$\}$ and $\{$ratings of this user for other movies$\}$ in R^{train}.

Results of lecun-team

Method	RMSE test set	RMSE platform	Time	Rank platform
GD MF	0.88	0.86	20s	1
ALS MF	0.91	0.87	15s	1
iDMF	**0.85**	**0.80**	250s*	1

GD MF : Gradient descent matrix factorization

ALS MF : Alternating least squares matrix factorization

iDMF : improved deep matrix factorization

* : improvable by parallelization

Method

Deep matrix factorization[1]

- Dataset : $(\text{user}_i, \text{movie}_j),\ \forall i, j$ s.t. $r_{ij} \neq \varnothing$
- 2 neural networks : $\text{NN}_{\theta_1}^{\text{movie}} : \mathbb{R}^n \to \mathbb{R}^{256}$, $\text{NN}_{\theta_2}^{\text{user}} : \mathbb{R}^m \to \mathbb{R}^{256}$
- Compute cosine similarity
  $$\delta_{\theta_1,\theta_2}(\text{user}_i, \text{movie}_j) = \frac{\langle \text{NN}_{\theta_1}^{\text{movie}}(\text{movie}_j), \text{NN}_{\theta_2}^{\text{user}}(\text{user}_i)\rangle}{\|\text{NN}_{\theta_1}^{\text{movie}}(\text{movie}_j)\| \|\text{NN}_{\theta_2}^{\text{user}}(\text{user}_i)\|} \in [-1, 1]$$
- Rescale $\delta_{\theta_1,\theta_2}(\text{user}_i, \text{movie}_j) :\ [-1, 1] \to [0, 1]$, and $r_{ij} : [0, 5] \to [0, 1]$
- Compute binary cross entropy : $\text{BCE}(\delta_{\theta_1,\theta_2}(\text{user}_i, \text{movie}_j), r_{ij})$ and run SGD with respect to θ_1 and θ_2

Interpretation

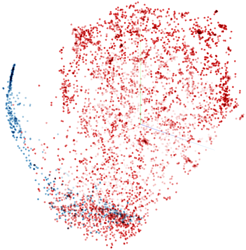

Figure – T-SNE of projections, view 1

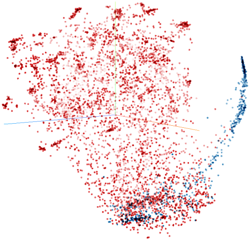

Figure – T-SNE of projections, view 2

Figure – Project movies and users in the same space such that the more a user is likely to like a movie, the more the projections of the movie is close to the projection of the user in this space.

Improvements

Improvements

- Learning the scalar product (solving over-regularization)
- Adding metadata
- Adding denoising auto-encoder (adding compatible regularization)
- Adding ensembling of K folds

Zoom on learning the scalar product

- Learn $S \in \mathcal{M}_{256,256}(\mathbb{R})$
- Use $\langle \cdot, \cdot \rangle : (X, Y) \to X^T S^T S Y$ to compute cosine similarity
- This gives the model the freedom to project movies and users into two separate subspaces, which is perhaps more suited to the problem, and then learn a distance measure to compare elements in these two separate subspaces.

Illustration of the effect

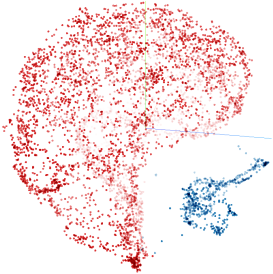

Figure – T-SNE of projections after improvements, view 1

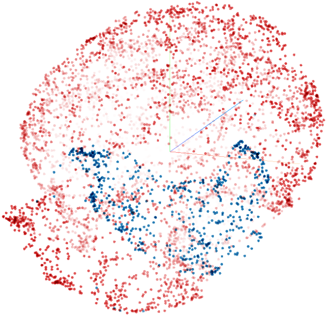

Figure – T-SNE of projections after improvements, view 2

Bibliography

[1] H.-J. Xue, X. Dai, J. Zhang, S. Huang et J. Chen, « Deep matrix factorization models for recommender systems., » in *IJCAI*, Melbourne, Australia, t. 17, 2017, p. 3203-3209.