Part 6: WRMF: Weighted Regularized Matrix Factorization

Robert Kwieciński

Adam Mickiewicz University

May 8, 2021

Idea

The matrix factorization model was designed for explicit data and gives really good results. In the learning procedure only items that the user interacted with are used.

Of course for implicit feedback we can assume all unspecified ratings to be zeros and learn our model, but:

- in case of RMSE loss function mistakes in estimating zeros (which we assumed) are then as important as mistakes in estimating ones (which we know),
- optimization through gradient descent might be really slow in case of sparse datasets.

Idea

The matrix factorization model was designed for explicit data and gives really good results. In the learning procedure only items that the user interacted with are used.

Of course for implicit feedback we can assume all unspecified ratings to be zeros and learn our model, but:

- in case of RMSE loss function mistakes in estimating zeros (which we assumed) are then as important as mistakes in estimating ones (which we know),
- optimization through gradient descent might be really slow in case of sparse datasets.

But the idea of replacing missing entries with zeros is not so bad. We will see that both problems can be overcome by Weighted Regularized Matrix Factorization (WRMF) model introduced in [1].

WRMF model

Denote by r_{ui} the "rating" made by the user u regarding the item i. It might be the real rating in case of explicit feedback dataset or number of times user visited the given page in implicit feedback. In case of no interactions we assume $r_{ui} = 0$. As in the original paper we will call r_{ui} values **observations**.

WRMF model

Denote by r_{ui} the "rating" made by the user u regarding the item i. It might be the real rating in case of explicit feedback dataset or number of times user visited the given page in implicit feedback. In case of no interactions we assume $r_{ui} = 0$. As in the original paper we will call r_{ui} values **observations**.

Define a **confidence** of an observation by $c_{ui} = 1 + \alpha r_{ui}$, where α is a hyperparameter.

WRMF model

Denote by r_{ui} the "rating" made by the user u regarding the item i. It might be the real rating in case of explicit feedback dataset or number of times user visited the given page in implicit feedback. In case of no interactions we assume $r_{ui} = 0$. As in the original paper we will call r_{ui} values **observations**.

Define a **confidence** of an observation by $c_{ui} = 1 + \alpha r_{ui}$, where α is a hyperparameter.

Also define a **preference** by: $p_{ui} = \begin{cases} 1 & \text{if } r_{ui} > 0, \\ 0 & \text{if } r_{ui} = 0. \end{cases}$

WRMF model

Denote by r_{ui} the "rating" made by the user u regarding the item i. It might be the real rating in case of explicit feedback dataset or number of times user visited the given page in implicit feedback. In case of no interactions we assume $r_{ui} = 0$. As in the original paper we will call r_{ui} values **observations**.

Define a **confidence** of an observation by $c_{ui} = 1 + \alpha r_{ui}$, where α is a hyperparameter.

Also define a **preference** by: $p_{ui} = \begin{cases} 1 & \text{if } r_{ui} > 0, \\ 0 & \text{if } r_{ui} = 0. \end{cases}$

Then our cost function is:

$$\min_{x_*, y_*} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda(\sum_u ||x_u||^2 + \sum_i ||y_i||^2),$$

where x_u is user's u f-dimensional embedding and y_i is item's i f-dimensional embedding.

ALS

In our objective function we sum over all (user, item) pairs which in real use cases can easily exceeds billions.

ALS

In our objective function we sum over all (user, item) pairs which in real use cases can easily exceeds billions.

Alternating least squares procedure assumes that either user or item latent factors are fixed and we analytically determine the second one, repeating the procedure few times.
In this case our loss function becomes quadratic and for example for each user u, vector latent representation x_u minimizing the loss function is:

$$x_u = (Y^T C^u Y + \lambda I)^{-1} Y^T C^u p(u),$$

where C^u is a diagonal matrix with $C^u_{ii} = c_{ui}$.

ALS

In our objective function we sum over all (user, item) pairs which in real use cases can easily exceeds billions.

Alternating least squares procedure assumes that either user or item latent factors are fixed and we analytically determine the second one, repeating the procedure few times.
In this case our loss function becomes quadratic and for example for each user u, vector latent representation x_u minimizing the loss function is:

$$x_u = (Y^T C^u Y + \lambda I)^{-1} Y^T C^u p(u),$$

where C^u is a diagonal matrix with $C^u_{ii} = c_{ui}$. Note that

$$Y^T C^u Y = Y^T Y + Y^T (C^u - I) Y,$$

and $Y^T Y$ is independent of u.

It leads to the fact ALS procedure is fast and scales linearly with the size of the data.

Python implementation

To do (especially for absent students):

- Go through - *P6. WRMF (Implicit ALS)* notebook to:
  - check implementation of WRMF model using Implicit library - Implicit library is really fast,
  - compare the results using different loss functions,
  - check the impact of some hyperparameters on evaluation measures.
  - **project task 7: check how number of iterations of WRMF model influence the evaluation metrics**

References I

[1] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp. 263–272, Dec. 2008. DOI: 10.1109/ICDM.2008.22.