ALS model for recommendation system

Abdelwahid Benslimane
wahid.benslimane@gmail.com

Recommendation based on the alternative least squares (ALS) model is an approach that falls more generally within the scope of collaborative filtering methods.

These methods consider that users, like product items, are described (almost) exclusively by the contents of a user - products items matrix (matrix X) that lists preferences/ratings already assigned by users.

	Product item 1	Product item 2	Product item 3	Product item 4	Product item 5
User 1	5		2		
User 2	5	4	7	6	1
User 3				3	2
User 4		5	6		
User 5			6		
User 6	4	2			

More specifically, ALS is an algorithmic brick used to solve the optimization problem associated with the factorization of the matrix X.

The principle behind methods based on matrix factorization is to look for latent factors, in relatively small numbers, which well explain the content of the matrix X.

Product items and users would therefore be described by vectors of the same dimension, given by the selected number of latent factors.

One of the first factorization methods explored was the singular value decomposition (SVD). The major problem is that such a decomposition requires a complete matrix, whereas the matrix X has empty parts corresponding to the absence of notes, and an absent value cannot be assimilated to a 0 in this specific use case. A decomposition that assimilates absent values to 0 would therefore produce an irrelevant solution.

If you want to learn more about SVD, you can consult the document where I explain and use it for image compression, available in one of my GitHub repositories: abenslimaneakawahid/iterative-methods (github.com) (I encourage you to first read my documents about the power method and the deflation method).

In order to take into account only the data present in the matrix, a regularization solution is required.

Regularized factorization is a set of methods which seek an approximation of reduced rank d of a matrix, taking into account only the values present in the matrix and including a regularization technique.

The factorization can be modelled as follows:

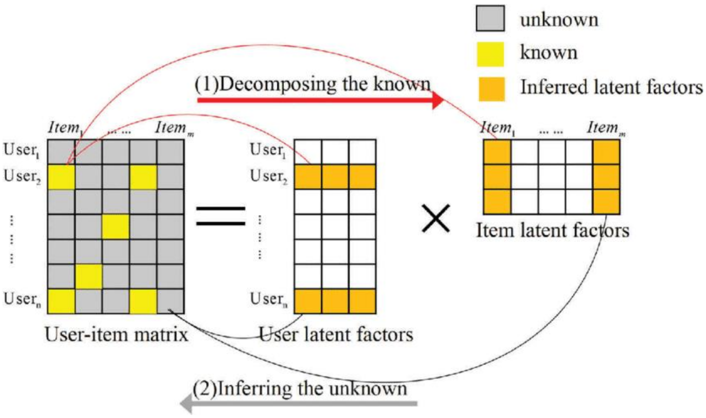

Image taken from:
https://www.researchgate.net/publication/321344494_Integrating_spatial_and_temporal_contexts_into_a_factorization_model_for_POI_recommendation

The unknown ratings can be inferred once the reduced rank matrix of users represented by latent factors (each row is the reduced representation of a user) and the reduced rank matrix of product items represented by latent factors (each column is the reduced representation of a product item) have been obtained.

If a previously unknown rating (i.e. it can be assumed that the user involved has not acquired or consumed the product in question), and inferred using this method, is considered high enough, then the product item can be recommended to the user considered.

The corresponding optimization problem is the following:

$$\min_{u_i, p_j} \sum_{Present(i,j)} \left(x_{ij} - u_i^T . p_j\right)^2 + \lambda \left(\sum_i ||u_i||^2 + \sum_j ||p_j||^2 \right) \quad (1)$$

Here, u_i is the reduced representation (of dimension d equal to the number of factors) of a user and p_j is the reduced representation (of dimension d) of a product item. The constant λ controls the regularization.

The minimization problem can be solved by an iterative algorithms such as ALS for example.

After finding the u_i and p_j for all users and respectively all product items, the prediction of the (unknown) rating that user k should give to product item l is:

$$u_{kl} = u_k^T . p_l$$

This approach, which combines regularized factorization and the associated iterative resolution algorithm, makes it possible to integrate other aspects into the model, such as the modelling of a user bias and an article bias, or the modelling of confidence levels in the scores present in the data matrix.

In Spark, the implementation of regularized factorization includes this regularization modification:

$$\min_{u_i, p_j} \sum_{Present(i,j)} \left(x_{ij} - u_i^T . p_j\right)^2 + \lambda \left(\sum_i n_i ||u_i||^2 + \sum_j n_j ||p_j||^2 \right) \quad (2)$$

n_i = total number of ratings given by the user i
n_j = total number of ratings received by the product item j

The function (2) to be minimized is not convex, but for fixed u_i it is convex with respect to p_j, and for fixed p_j it is convex with respect to u_i. ALS therefore converges to a minimum that is not necessarily global, so it can be useful to make several attempts with different initializations.

Each iteration of the ALS algorithm alternates between two phases:

- with u_i fixed, $1 \leq i \leq n_u$, it obtains the p_j, $1 \leq j \leq n_p$, as the solution of a linear system (in a way similar to the least squares solution for a linear regression)

- with p_j fixed, $1 \leq j \leq n_p$, it obtains the u_i, $1 \leq i \leq n_u$, as the solution of a linear system (in a way similar to the least squares solution for a linear regression)