Advanced topics in Machine Learning. Concept of the solution of the programming assignment

Team Chanjo.
Johannes Jendersie, Anton Niadzelka

July 2, 2013

1 Challenge decription

The challenge's objective is easily described: Create a recommender system for first names! Given a set of names for which a user has shown interest in, the recommender should provide suggestions for further names for that user. The recommender's quality will be assessed on an evaluation data set. Thus, the task can be considered a standard item recommendation task.

2 Solution

We received data about user activities from nameling.net website. Data consists of 515,848 activities made by 60,922 users. Our idea was to assign to each activity corresponding rating. The most explicit user expression will be ranked higher. We thought that *ENTER SEARCH* will receive the highest rank, as the evaluation is restricted only to this activity and all other activities are biased towards the lists of names which were displayed. However, using our evaluation measure we obtained better results when the highest rating was assigned to *ADD FAVORITE* action.

So, afterwards we obtained a table where for each name we will have some user rating or nothing. It is not possible to store the whole table on a usual laptop RAM, as the number of user names is too high. Moreover, the table is sparse as we have less than 10 activities in average pro user. So, we stored only existing ratings for each user with the index number of the corresponding name.

The recommendation approach, that we are going to use, is based on a model (1) described in a paper [2].

$$\hat{r}_{ui} = \mu + b_u + b_i + p_u^T q_i, \text{ where} \tag{1}$$

μ - average over all table ,

b_u and b_i indicate the observed deviations of user u and item i, respectively, from the average.

For a given item i, the elements of q_i measure the extent to which the item possesses those factors, positive or negative.

The elements of p_u measure attitude of the user to latent factors.

Factors mentioned above obtain using singular value decomposition(SVD) of an initial matix, as SVD maps both users and items to a joint latent factor space of dimensionality f. As it stated in book [1] the latent space tries to explain ratings by characterizing both products and users on factors automatically inferred from user feedback. Usual SVD algorithms are not easily applicanle in our case, as we have a really huge sparse matrix. So, we are going to use algorithm described in [2] in chapter 4.3.

2.1 Learning

The values b_u and b_i has to be learned after the initialization. During that learning x_j and y_j will be adapted too.

3 Evaluation

The quality of the result was measured by the root mean squared error (RMSE) (2).

$$\sqrt{\sum_{(u,i) \in TestSet} (r_{ui} - \hat{r}_{ui})^2 / \left| TestSet \right|} \tag{2}$$

We have 8 parameters that could be tuned. These are 5 initial ratings for each action, the number of latent factors f, shrinkage and threshold parameters. For that we cross validated our model with different values of these parameters and afterwards have chosen one with the best result. Cross validation was performed on user actions, not on users.

4 Result

Recommendation model with the parameters listed below has the smallest RMSE = 0.392.

Ratings:

LINK SEARCH = 0.05, *ENTER SEARCH* = 0.2, *LINK CATEGORY SEARCH* = 0.05, *NAME DETAILS* = 0.1, *ADD FAVORITE* = 0.6.

Number of the latent factors = 40, Shrinkage = 3.9, threshold = 0.0001.

References

[1] Recommender Systems Handbook by Ricci, F.; Rokach, L.; Shapira, B.; Kantor, P.B. 2011, XXIX, Springer, Chapter 5

[2] Modeling Relationships at Multiple Scales to Im-prove Accuracy of Large Recommender Systems, Bell, R.M., Koren, Y., and Volinsky, C., Proc. 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2007.