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

Team Chanjo.
Johannes Jendersie, Anton Niadzelka

June 8, 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 is to assign to each activity corresponding rating. The most explicit user expression will be ranked higher. *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.

So, afterwards we will get a table where for each name we will have some user rating or nothing. It is not possible to store RAM on usual laptop whole table, 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 store for each user only existing ratings with index number of the corresponding name.

Recommendation approach, that we are going to use, is based on a model (1) described in book Recommender Systems Handbook [1].

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T \left(|R(U)|^{-\frac{1}{2}} \sum_{j \in R(u)} (r_{uj} - b_{uj}) x_j + y_j) \right), \text{ where} \quad (1)$$

μ - average overl all table ,

$R(u)$ - set that contains all items rated by user u,

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

$$b_{ui} = \mu + b_u + b_i$$

y_j - factors used to characterize users based on the set of items they rated.

x_j - factors used to characterize items based on the set of items they rated.

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

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.

3 Evaluation

Quality of the result will be measured by the root mean squared error 2.

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

We have a few parameters that have to be tuned. They are initial ratings for each action, number of latent factors f. For that we are going to cross validate our model with different values of these parameters and afterwards choose one with the best result.

4 Work Plan

Johannes Jendersie:

1. Team work organisation,
2. Program Interface,
3. Sparse Matrix,

4. Recommender Algorithm implementation.
5. Query Interface

Anton Niadzelka:

1. Idea Description,
2. Documentation,
3. Evaluation,
4. Performance Measurement implementation,
5. Recommendation creation

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.