CIL Project: Collaborative Filtering

Dave Eschbach, Fanyi Zhou, Ulla Aeschbacher, Xiyue Shi
Department of Computer Science, ETH Zurich, Switzerland

***Abstract*—**

I. Introduction

II. Our Novel Solution

III. Comparing To Baselines

We are comparing our novel solution to two others so that we can see how much improvement was done. The first of the two other solutions is the one from Assignment 3, where we were first completing the missing ratings in the matrix by setting them to the average over all observed ratings for a particular item. Then we tried to find an underlying structure in the data by performing SVD, where $X^{train} = UDV^T$. We vaired the number k of eigenvalues used and truncated U and V accordingly. Figure 1 shows the Root-Mean-Squared-Error for k between 0 and 95, in intervals of 5. In Figure 2 we can see the error more clearly for k between 5 and 95. Apparently, taking the biggest 10 eigenvalues is giving us the best approximation to the data structure. The best result we achieved on kaggle this way was 1.16473.

The second solution we are presenting here for comparison is a neural network. We trained a bidirectional LSTM. It has a bias layer, so that we could take into account that different people are rating higher or lower in general. We trid different dropout values, number of nodes, batch-sizes and trained over various epochs. In Figure 3 you can see the Mean-Squared-Error loss over ten epochs of the neural network. The best result we achieved on kaggle this way was 1.04627. There are people that achieved much better results with a neural network [1], but our feeling is that our data was too sparse for this.

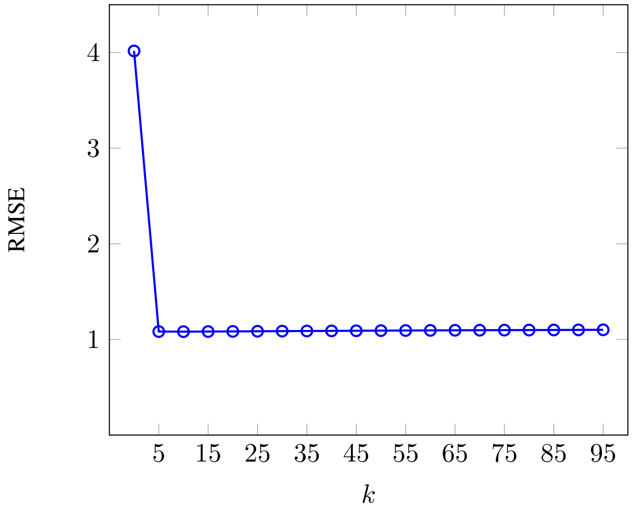

Figure 1. RMSE in SVD with different k

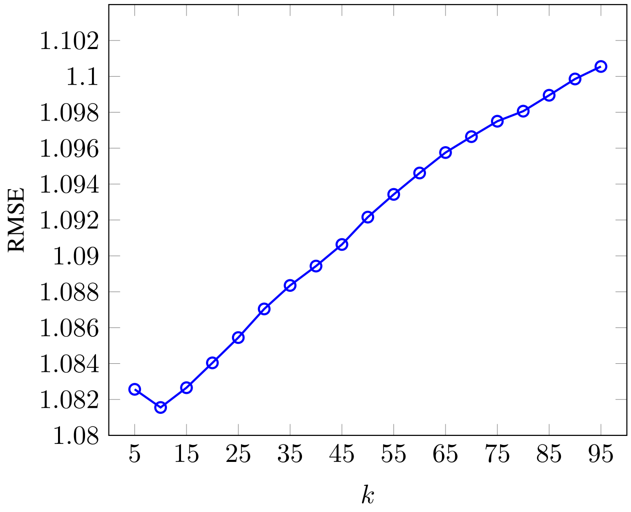

Figure 2. RMSE in SVD with different k, shown in more detail

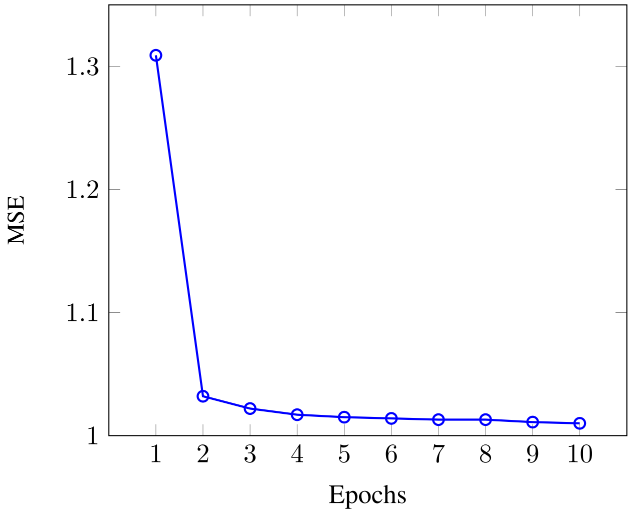

Figure 3. MSE loss in LSTM over 10 epochs.

Comparing the two solutions with our novel approach result of 0.98362 we can see

IV. Summary

V. Computational Intelligence Laboratory Requirements

Your semester project is a group effort. It consists of four parts:

1) The programming assignments you solve during the semester.
2) Developing a novel solution for one of the assignments, e.g. by combining methods from previous programming assignments into a novel solution.
3) Comparing your novel solution to previous assignments.
4) Writing up your findings in a short scientific paper.

A. Grading

There are two different types of grading criteria applied to your project, with the corresponding weights shown in brackets.

Competitive

The following criteria is scored based on your rank in comparison with the rest of the class.

- time taken for computation (10%)
- average rank for all other criteria relevant to the task, for example reconstruction error and sparsity (20%)

The ranks will then be converted on a linear scale into a grade between 4 and 6.

Non-competitive

The following criteria is scored based on an evaluation by the teaching assistants.

- quality of paper (30%)
- quality of implementation (20%)
- creativity of solution (20%)

B. Submission System

The deadline for submitting your project report is Friday, 22 June 2012. You need to submit:

- PDF of paper.
- Archive (`.tar.gz` or `.zip`) of software. Please do not forget to include author information in the source archive.

Important: Please check the submission instructions on the webpage as it is the most updated instructions.

References

[1] G. Tseng. (2017) Clustering and collaborative filtering - implementing neural networks. Https://medium.com/@gabrieltseng/clustering-and-collaborative-filtering-implementing-neural-networks-bccf2f9ff988.