DS-GA 1004 Final Project Report

Andrei Kapustin

ak7671

Overview

In this project, I applied the tools I have learned in the class to build and evaluate a recommender system. The dataset used for the project is Goodreads dataset[1]. It contains 876 thousand users, 2.4 million books and 223 million interactions. Recommender system was built using Spark and its implementation of the ALS algorithm.

Data processing

The data was split into training, validation and testing sets. All users should be present in the training set because otherwise the model can't give recommendations for users with no history (if cold start techniques are not used). Because of that, Training set contains all interactions for 60% of users and half of the interactions for another 40%. Validation and testing sets contain half of the interactions for 20% of users each. All books that had no interactions in the training set were dropped from validation and testing sets. Users with 10 or less interactions were dropped because they don't provide sufficient data for evaluation. Interactions with rating = 0 were dropped because zero rating represents the absence of rating and doesn't provide any additional information for the model

I was unable to run the model on the whole dataset due to high load on the cluster during the last days before the project deadline. All results described in this report are for the half of the dataset. It was subsampled to contain all interactions for 50% of users.

Model and experiments

The model used for the recommender system is Alternating Least Squares (ALS). ALS method iteratively approximates rating matrix *R* as a sum of two lower-rank matrices *X* and *Y* (*X * Yt = R).* During each iteration, one of the matrices is held constant, while the other is solved for using least squares. The newly-solved matrix is then held constant while solving for the other one.

The hyperparameters tuned for the ALS are *rank* and regularisation parameter (*regParam*). *Rank* determines the rank of factor matrices *X* and *Y*. *RegParam* determines the strength of regularization.

Metric used for evaluation is Root Mean Square Error (RMSE). It is a measure of prediction error (lower is better).

Other metrics that were computed are Mean Average Precision (MAP), precision at 500 and Normalized Discounted Cumulative Gain at 500 (NDCG at 500).

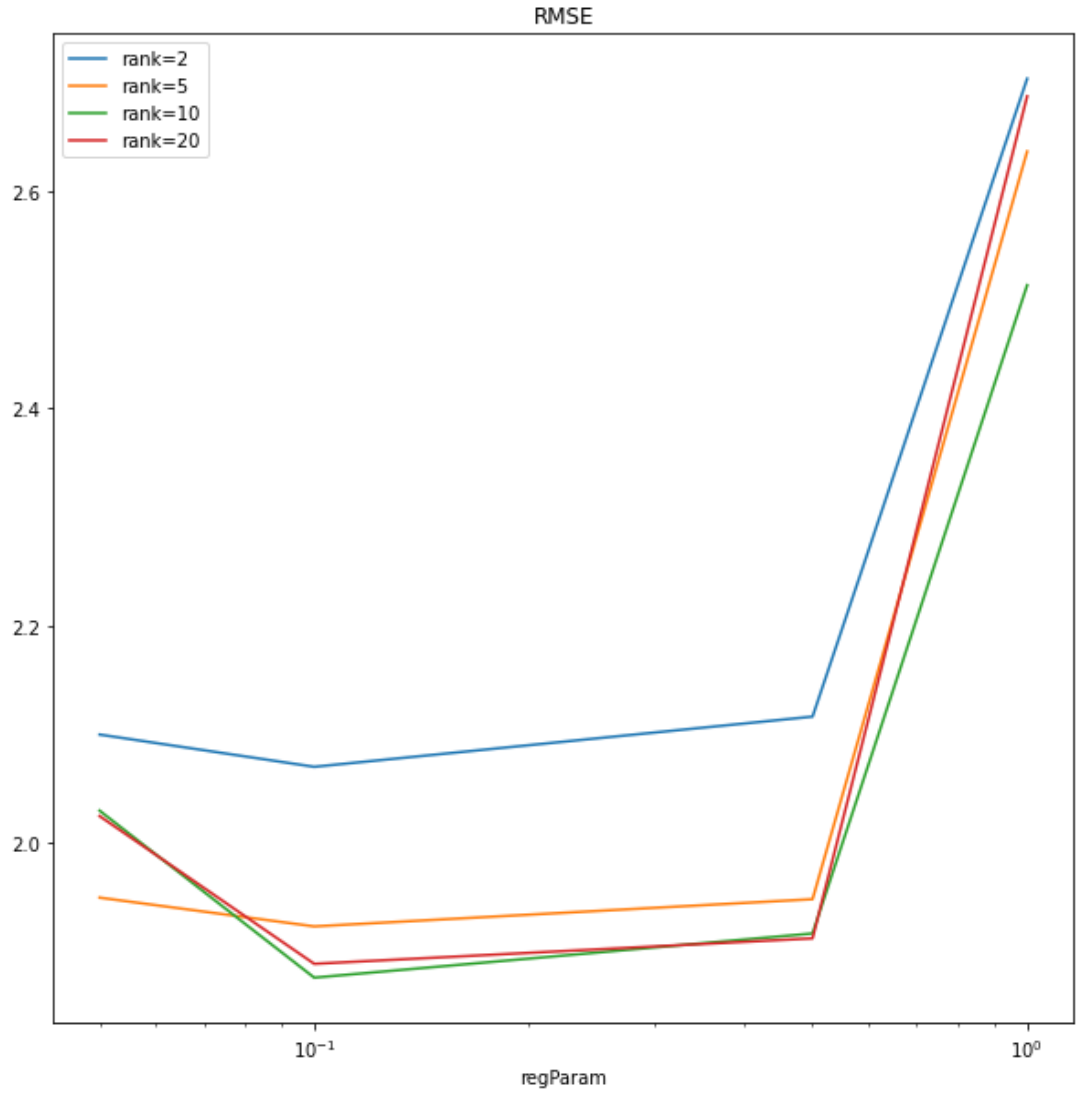
RMSE was computed on all interactions in the validation set, for ranking metrics all interactions with rating >= 3 from validation set were used as ground truth. Values of ranking metrics are low because most

of the recommended documents were in the training set but not in validation. They were computed just for comparison and were not used to tune hyperparameters.

Best hyperparameters found are *rank* = 10 and *regParam* = 0.1. Performance on the testing set with those hyperparameters:

RMSE	MAP	Precision at 500	NDCG at 500
1.81035	0.00591618	0.0787266	0.30995

Plot of RMSE for different hyperparameter values:



I made similar plots for other metrics but I can't fit them into a four page report.

Extension

For an extension I made a visualisation of latent factors using t-SNE [2] and Goodreads book genres dataset.

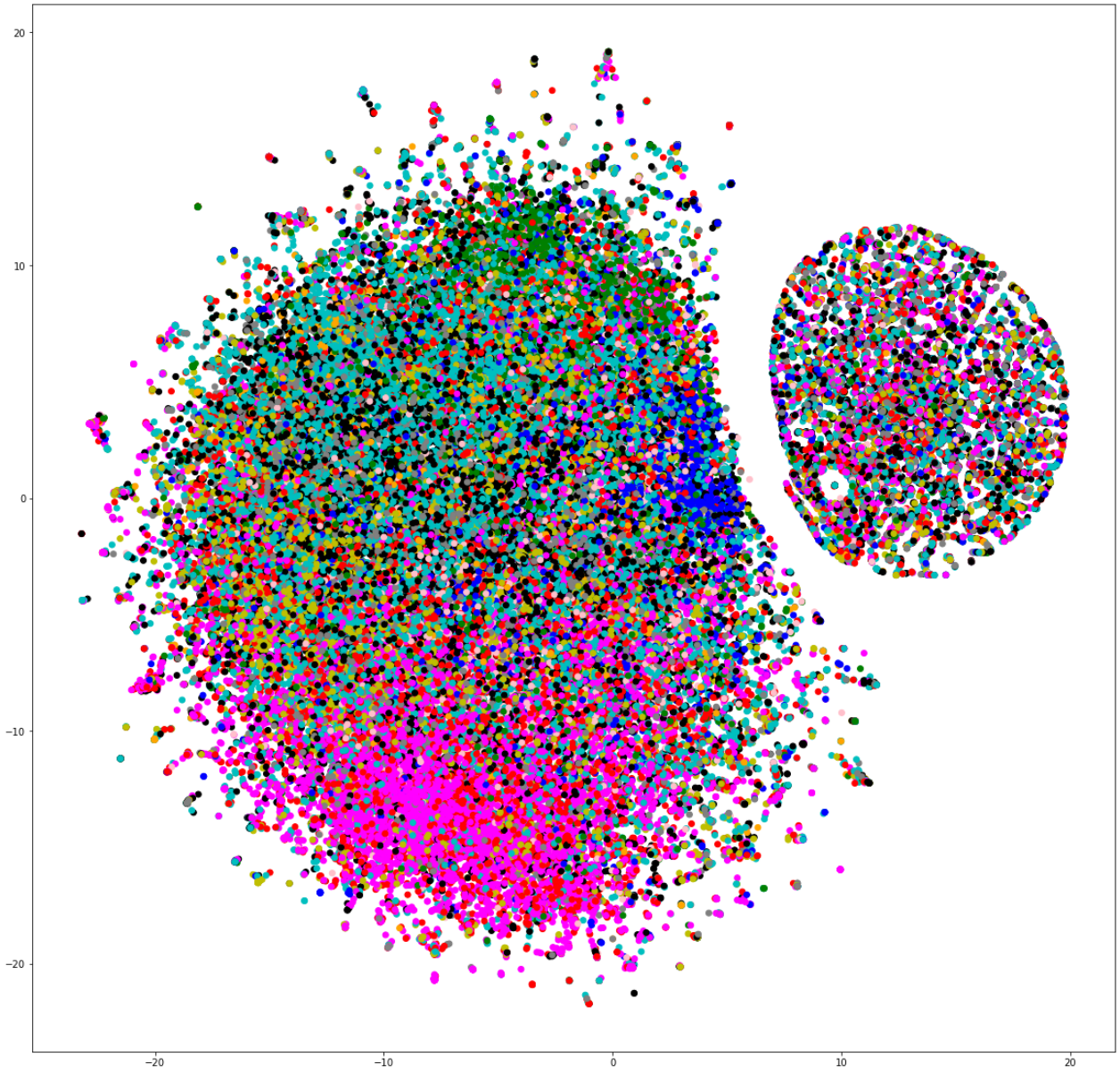
T-SNE is a nonlinear dimensionality reduction algorithm. It constructs a probability distribution over

pairs of input vectors such that the less is distance between two vectors, the higher is the probability of their pair. Then it defines a similar distribution over pairs in low-dimensional space and minimizes the KL divergence between the distributions. Its two main hyperparameters are perplexity and learning rate.

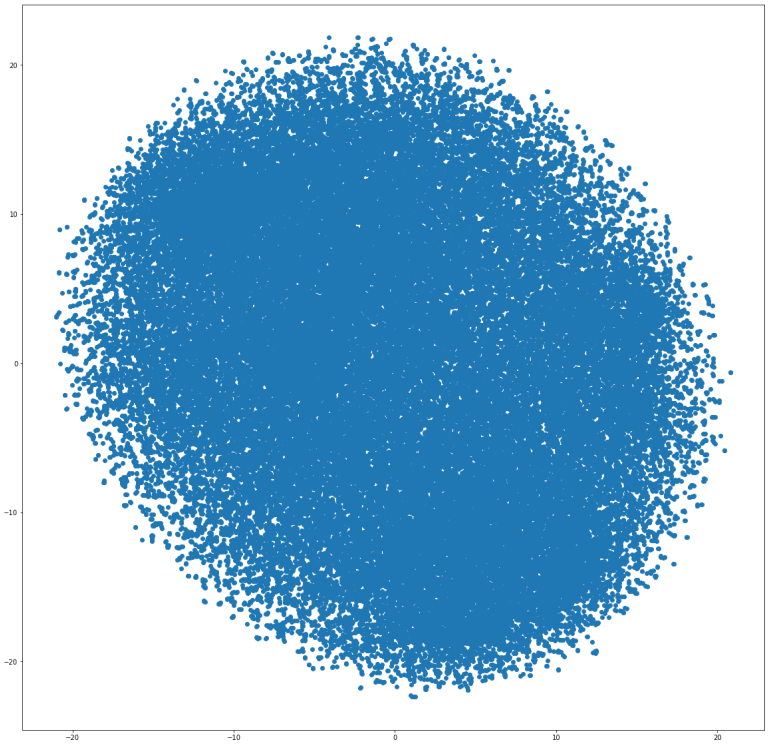
In the book genres dataset each book is associated with a dictionary that contains the number of times the book was assigned to each of the genres by users. I assigned each book the most frequent genre from its dictionary. Approximately 12% of books did not have any genres. These books were not used for the visualization.

First, I made a visualization of book latent factors from a trained ALS model. I used t-SNE to project the latent factors onto 2D space. Each book was assigned the genre that

Each point represents a book, the color of the point represents its genre. I was expecting books of the same genre to be close together, but it was not the case. As can be seen on a picture, there are two clusters, each one containing all the genres. The bigger cluster has a few regions with dominant genre: ‘romance’ (magenta) in the bottom and ‘children’ (blue) on the right. This structure remains roughly the same for any combination of hyperparameters.



I also visualized user latent factors, but for any combination of hyperparameters there always was one big cluster with no interesting structure.



Contribution of team members

Andrei Kapustin - everything

References

[1] Mengting Wan, Julian McAuley, "Item Recommendation on Monotonic Behavior Chains", RecSys 2018

[2] van der Maaten, L.J.P.; Hinton, G.E. Visualizing High-Dimensional Data Using t-SNE. Journal of Machine Learning Research 9:2579-2605, 2008