Exploring correlation between amount of supplied information and perceived quality of recommendations

Konstanty Kowalewski, Wian Stipp, Jakub Grohmann

LSESU Machine Learning Society

Introduction

How well can a machine learning algorithm predict consumer preference in the hostile conditions of the Internet? Should we worry about predictive power of information we leave online? We try to answer these questions by looking into how the quality of machine generated recommendations depend on the amount of data supplied by the user in a noisy, non-synthetic dataset.

We focused on the Amazon reviews dataset, utilized Factorization Machine model and developed a small UI for collecting responses of arbitrary length. We will crowdsource the ratings, compute the recommendations and send back the best 10 and worst 10 results asking for opinion on accuracy.

Dataset

We use 1M subset of the Amazon Books Reviews dataset [1] with slight class balancing to account for majority bias of high ratings.

	Unique	25%	50%	75%
Users	362,701	1	1	3
Items	67,496	10	10	20

Tab. 1: Quantiles of outgoing connections

Model

We were looking for an *established* model that would run *fast*. We settled on a two-way Factorization Machine [2]. It runs in linear time w.r.t. number of features and models pairwise interactions with fairly small risk of overfitting. We wrote a simple PyTorch implementation with support for sparse features in coordinate format and got a consistent <1 RMSE.

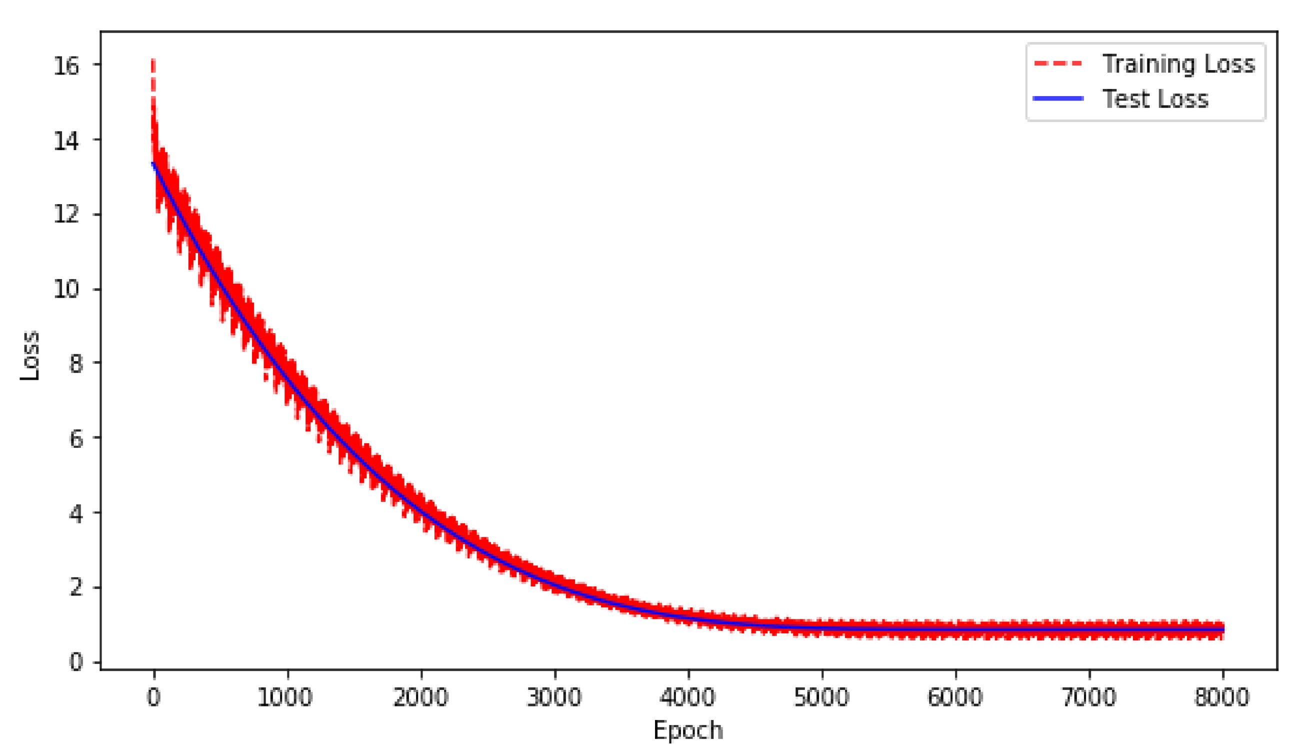


Fig. 1: Loss over 8000 iterations in 20K subset

To utilize more information, we used the review helpfulness. We flattened the score (number of positive votes over number of all votes) by setting it to

$$(1 + (p/a)^{1.8})^a \text{ for } a > 0 \text{ and } 1 \text{ for } a = 0$$

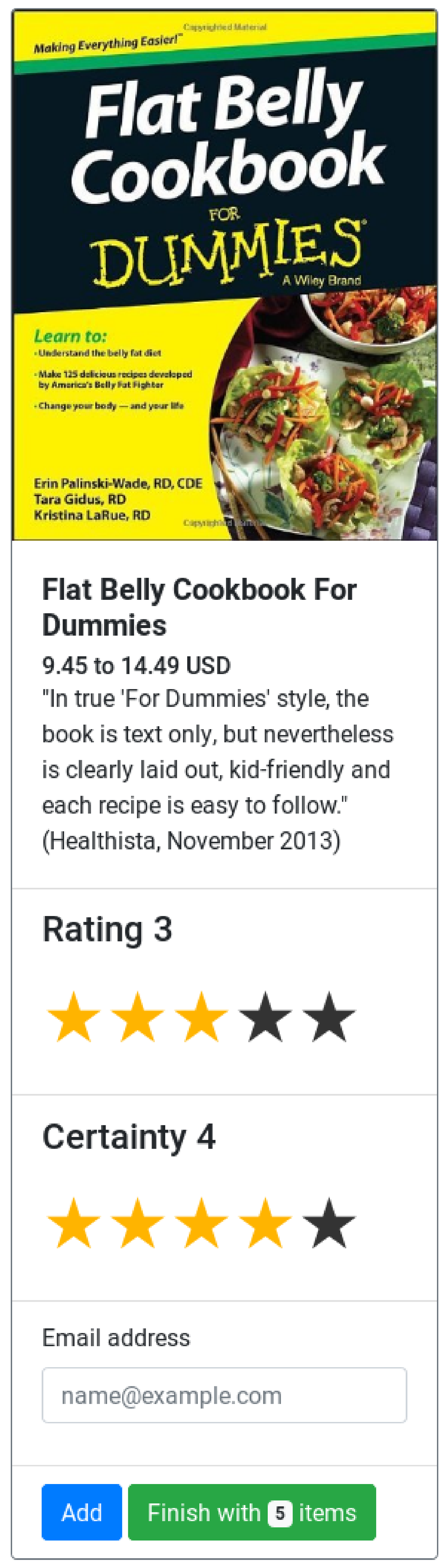
to avoid penalizing for negative votes. Following [3], we one-hot encoded users and items and concatenated with helpfulness.

Data collection

We developed a lightweight, responsive React application to collect responses of arbitrary lengths. It stores the whole sequence of responses in browser state and dynamically queries our Flask backend for new random items. When finished, it hashes email address together with timestamp and stores the information back on the server.

This project is in progress, we currently have over 40 responses of well-balanced distribution of lenghts in the pilot. As soon as the system becomes stable, we will deploy the full version.

Relevant code is available at `gitlab.com/mlatlse`. Please try the pilot application at `mlatlse.kszk.eu`!



Further work

Possible ways to improve this project would be to repeat the experiment on a less noisy, possibly synthetic dataset, such as MovieLens. It would be also interesting to try online variants of the model, such as SFTRL [4] to omit the step of computing batch recommendations.

References

[1] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, "Image-based recommendations on styles and substitutes," in *SIGIR 15*, 2015.

[2] S. Rendle, "Factorization machines," in *2010 IEEE International Conference on Data Mining*, 2010.

[3] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme, "Fast context-aware recommendations with factorization machines," in *SIGIR 11*, 2011.

[4] L. Luo, W. Zhang, Z. Zhang, W. Zhu, T. Zhang, and J. Pei, "Sketched follow-the-regularized-leader for online factorization machine," in *KDD18*, 2018.