Preventing Overfitting via Sample Reweighting for Recommender System Incremental Update (Student Abstract)

Danni Peng[1,2], Xiaobo Hu[3], Anxiang Zeng[1], Jie Zhang[1]
[1]School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore
[2]Alibaba-NTU Singapore Joint Research Institute
[3]Alibaba Group, Beijing, China
danni001@ntu.edu.sg, benjian.hxb@alibaba-inc.com, zeng0118@ntu.edu.sg, zhangj@ntu.edu.sg

Abstract

Incremental update of recommender system models using only newly arrived data may easily cause the model to overfit to the current data. To address this issue without relying on historical data, we propose a sample reweighting method based on prediction performance of previous model on current data. The proposed method effectively alleviates the problem of overfitting and improves the performance of incremental update.

Introduction

Recommender systems in real-world applications often deal with billions of user interaction data daily. In order to capture the most recent interest drift of users and change in perception of items, it is important to periodically update the predictive model with newly arrived data. However, training on new data only may easily cause the model to overfit to the current data and forget past patterns learned. To address this issue without relying on historical data, we propose an Incremental Update framework with Sample Reweighting strategy (SRIU). More specifically, we design a sample reweighting method based on prediction performance of previous model on current training data, such that samples that are well predicted by the previous model are given larger weights, while samples that are not well fitted by the previous model are given smaller weights. The proposed framework is model-agnostic. We instantiate the proposed framework on classic Matrix Factorisation model and demonstrate its effectiveness on two e-commerce datasets.

Problem Description

Let $D = \{(\mathbf{x}, y)\}$ denote the dataset consisting of user-item interactions, where $\mathbf{x}$ represents the input features from user and item, and $y \in \{0, 1\}$ indicates whether an interaction is observed (such as click, review, purchase). Datasets collected at different periods are denoted by $\{D_0, ..., D_t, D_{t+1}, ...\}$. For incremental update at period t, model from period $t-1$ is restored as initialization. It is then updated with current data D_t until convergence to obtain the model at period t. In a stochastic gradient descent step, the model parameters θ are updated as:

$$\theta \leftarrow \theta - \alpha w_i \frac{\partial l(\theta|\mathbf{x}_i, y_i)}{\partial \theta}, \tag{1}$$

where $l(\theta|\mathbf{x}_i, y_i)$ is the loss for sample $(\mathbf{x}_i, y_i)$, α is the learning rate, w_i is the weight of sample i. Different weights allow samples to contribute differently to the update of model parameters. Without applying any reweighting strategy, sample weights are uniform, i.e., $w_i = 1$ for all i.

The Proposed SRIU Framework

To prevent the model from overfitting to the current data, we propose a sample reweighting method based on prediction performance of previous model (i.e., the initialization model at each period). A similar idea has been adopted in text classification to facilitate fine-tuning in transfer learning, which assigns weight based on loss (Wang et al. 2019). Here, we propose to assign weight based on logit, the raw output generated by the model before it is passed to sigmoid normalization and loss function. In recommender system models, logit is a more direct indicator of model prediction performance on a sample, as it is usually used directly to compute predicted score for ranking.

A positive sample (i.e., y=1) is considered well predicted by the model if its logit has a large value, while a negative sample (i.e., y=0) is considered well predicted if its logit has a small value. Placing larger weights on samples that are well predicted by the previous model will encourage the updated model to preserve patterns learned in the previous model. While placing smaller weights on samples that are not well predicted will effectively prevent the updated model from overfitting to samples that deviate significantly from the learned patterns. Samples that are not well fitted by the model will result in large gradients, which during back-propagation will cause significant change in model parameters. Reducing their weights restricts the effects of outlier samples on model updating. Let $g_\theta(\mathbf{x}_i)$ denote the logit of sample i, we compute the weight w_i as:

$$w_i = \begin{cases} \dfrac{c(g_\theta(\mathbf{x}_i) - \mu_+)}{\sigma_+}, & y_i = 1 \\ \dfrac{-c(g_\theta(\mathbf{x}_i) - \mu_-)}{\sigma_-}, & y_i = 0 \end{cases}, \tag{2}$$

where μ_+ (μ_-) and σ_+ (σ_-) are the mean and standard deviation for logits of positive (negative) samples, and c is a positive constant that controls the dispersion of weights.

To avoid negative values of weights, we use softmax function to normalize w_i. Also, we want to ensure that the weights of samples that belong to the same user sum to the actual sample size of that user, so that all the users are sufficiently learned without placing too much emphasis on some well predicted users over the others. The resultant rescaled weight $\tilde{w_i}$ of sample i for user u is computed as:

$$\tilde{w_i} = \text{softmax}(w_i) \times |S_u| \,, \tag{3}$$

where S_u denotes the set of samples belonging to user u.

Algorithm 1 shows the overall incremental update framework with sample reweighting.

Algorithm 1: Incremental Update framework with Sample Reweighting (SRIU)

Input: Previous model parameters θ^*_{t-1}, current training dataset D_t
Output: Updated model parameters θ^*_t
$\theta \leftarrow \theta^*_{t-1}$ // Initialisation
for *each sample* $(\mathbf{x}, y)$ *in* D_t **do**
  Generate logit from $g_\theta(\mathbf{x})$
  Compute weight $\tilde{w}$ using (2) and (3)
  Replace $(\mathbf{x}, y)$ with $(\mathbf{x}, y, \tilde{w})$ in D_t
end
while *stop condition not reached* **do**
  Fetch a sample $(\mathbf{x}, y, \tilde{w})$ from D_t
  Update θ using (1)
end
return θ^*_t

Experiments

We evaluate the effectiveness of the proposed framework by instantiating on Matrix Factorization, a classic recommender system model that uses latent vectors to represent users and items. The loss function used here is binary cross-entropy.

We conduct experiments with two e-commerce datasets: Tmall.com and Sobazaar mobile app. For each dataset, it is divided into 31 periods. The first 18 periods are used to train an initial model, and the remaining 12 periods are used to conduct periodic training. Evaluation of model at each period is based on its prediction AUC for the next period.

We compare the proposed framework with 3 baseline training strategies: 1) Full Retraining (FR) that uses full historical data $\{D_0, ..., D_t\}$ to update model at period t, 2) Incremental Update (IU) that uses only current data D_t to update model at period t, and 3) SPMF (Wang et al. 2018) that maintains a reservoir of historical data and updates model at period t using both current data D_t and data in the reservoir. SPMF can be seen as an intermediate between FR and IU.

Table 1 shows the average AUC over 12 periods of training. FR and SPMF which use historical data together with current data are not comparable to IU. This implies that user interests and item perceptions have changed over time, and they are best captured by training with the most recent data only. Employing sample reweighting strategy further improves the performance of IU, as it serves to prevent overfitting to current data and preserve past patterns learned.

	Tmall	Sobazaar
FR	0.8062	0.7060
SPMF	0.8390	0.7306
IU	0.8408	0.7340
SRIU	**0.8497**	**0.7447**

Table 1: Average AUC over 12 periods of training

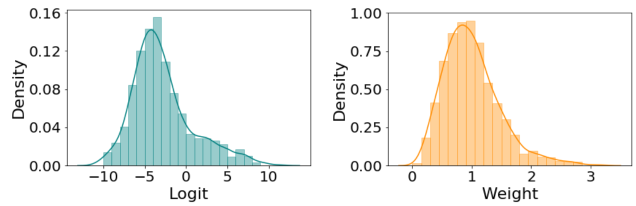

Figure 1: Distribution of logits (left) and weights (right) for a selected user from Tmall dataset.

To visualize the outcomes of the proposed weight computation method, we plot the distribution of logits and weights for a selected user from Tmall dataset (see Figure 1). It can be seen that the logits have an approximately Gaussian distribution, hence it is suitable to standardize it with mean and standard deviation in (2). The resultant weights computed from logits lie within a reasonable range from 0 to 3 and peak at 1. Our experimental codes are available at https://github.com/danni9594/SRIU.

Future Work

In the future, we will investigate the effectiveness of the proposed framework on different recommender system models. Also, we will explore other approaches to improve incremental update, such as the model-based approach (Zhang et al. 2020), and further develop a more general framework.

Acknowledgements

This research is supported, in part, by Alibaba Group through Alibaba Innovative Research (AIR) Program and Alibaba-NTU Singapore Joint Research Institute (JRI) (Alibaba-NTU-AIR2019B1), Nanyang Technological University, Singapore.

References

Wang, W.; Yin, H.; Huang, Z.; Wang, Q.; Du, X.; and Nguyen, Q. V. H. 2018. Streaming ranking based recommender systems. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 525–534.

Wang, Z.; Bi, W.; Wang, Y.; and Liu, X. 2019. Better Fine-Tuning via Instance Weighting for Text Classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 7241–7248.

Zhang, Y.; Feng, F.; Wang, C.; He, X.; Wang, M.; Li, Y.; and Zhang, Y. 2020. How to Retrain Recommender System? A Sequential Meta-Learning Method. *arXiv preprint arXiv:2005.13258* .