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

Danni Peng, Xiaobo Hu, Anxiang Zeng, Jie Zhang
School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore
Alibaba-NTU Singapore Joint Research Institute

Incremental Update

θ_t^* is obtained by training on D_t, with θ_{t-1}^* as initialization:

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

where $l(\theta|\mathbf{x}_i, y_i)$ is the loss for sample $(\mathbf{x}_i, y_i) \in D_t$, α is the learning rate, w_i is the weight of sample i. Without applying any reweighting strategy, sample weights are uniform, i.e., $w_i = 1$ for all i.

Instantiation on Matrix Factorization

Let p_u and q_v represent the latent vectors of user u and item v respectively, the logit $g(u, v)$ is computed by:

$$g(u, v) = p_u \cdot q_v,$$

We adopt negative log loss:

$$l(\theta|u, v) = -y \log(\sigma(g_\theta(u, v))) - (1-y)\log(\sigma(1-g_\theta(u, v))),$$

where $\sigma(\cdot)$ is the sigmoid function, $y \in \{0, 1\}$ is the label.

Incremental Update with Sample Reweighting (SRIU)

Weight Computation based on Logit

Placing larger weights on samples that are well predicted can help preserve past patterns, while placing smaller weights on samples that are not well fitted can restrict the effects of outlier samples on model updating. Let $g_{\theta_{t-1}^*}(\mathbf{x}_i)$ denote the logit of sample i:

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

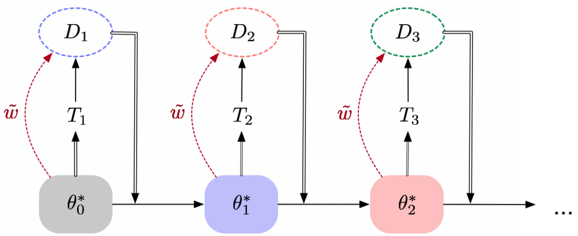
where μ_+ (μ_-) and σ_+ (σ_-) are the mean and standard deviation for logits of positive (negative) samples, and c is a positive constant that controls the dispersion of weights.

Weight Nomalization by User

To ensure that all the users are sufficiently learned without placing too much emphasis on some well predicted users. We normalize the weight $\tilde{w}_i$ of sample i for user u by:

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

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



Algorithm 1: 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

Baselines

- Full Retrain (FR) uses all the historical data $\{D_0, ..., D_t\}$.
- Incremental Update (IU) uses only current data D_t.
- SPMF (Wang et al. 2018) uses both current data D_t and samples in the reservoir.

Datasets & Evaluation Method

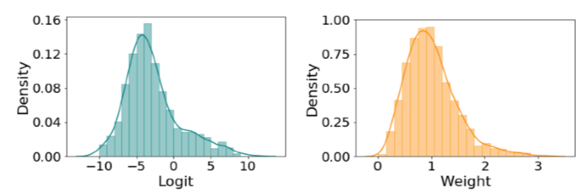
- 2 e-commerce datasets: Tmall.com and Sobazaar app.
- The first 18 periods are used to train an initial model, the remaining 12 periods are used for periodic training.
- Evaluation of model at each period is based on its prediction AUC for the next period.

Performance Comparison

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

Average AUC over 12 periods of training

Weight Distribution Visualization



Distribution of logits (left) and weights (right) for a selected user from Tmall dataset. The weights computed lie within a reasonable range from 0 to 3 and peak at 1.