PREMERE: Meta-Reweighting via Self-Ensembling for Point-of-Interest Recommendation

Minseok Kim, Hwanjun Song, Doyoung Kim, Kijung Shin, Jae-Gil Lee*
KAIST, Korea
{minseokkim, songhwanjun, doyo09, kijungs, jaegil}@kaist.ac.kr

Abstract

Point-of-interest (POI) recommendation has become an important research topic in these days. The user check-in history used as the input to POI recommendation is very imbalanced and noisy because of sparse and missing check-ins. Although sample reweighting is commonly adopted for addressing this challenge with the input data, its *fixed* weighting scheme is often inappropriate to deal with different characteristics of users or POIs. Thus, in this paper, we propose PREMERE, an *adaptive* weighting scheme based on meta-learning. Because meta-data is typically required by meta-learning but is inherently hard to obtain in POI recommendation, we *self-generate* the meta-data via self-ensembling. Furthermore, the meta-model architecture is extended to deal with the scarcity of check-ins. Thorough experiments show that replacing a weighting scheme with PREMERE boosts the performance of the state-of-the-art recommender algorithms by 2.36–26.9% on three benchmark datasets.

Introduction

With the prevalence of mobile devices and the emergence of location-based social networks (LBSNs), it has become feasible for people to share location-related contents. People visit point-of-interests (POIs) and share their check-in records to LBSN services, and discover potentially interesting POIs from the services. The check-in records, which indicate the preference of users, are harnessed to improve the quality of POI recommendation. Successful POI recommendation saves the users' time and effort in finding interesting POIs and helps business owners increase their profits by attracting potential customers to their venue. Thus, numerous POI recommender systems have been actively developed (Ye et al. 2011; Zhang and Chow 2013; Li et al. 2015).

The POI check-in history of users is represented by a *user-POI check-in matrix*, where rows correspond to users and columns correspond to POIs. Each (i, j)-th element indicates the number of check-ins at the j-th POI by the i-th user. While almost all prior studies have mainly relied on the user-POI check-in matrix, it is biased in two aspects. First, it is very sparse, where only a small fraction (around 0.1%) of entries are non-zero (Liu et al. 2017). Second, a zero entry does not necessarily indicate being uninteresting, because users may omit or forget to share their visits to POIs which they are interested in. Therefore, the class imbalance and noisy values of the user-POI check-in matrix may lead to poor performance of POI recommendation (Lian et al. 2014; Yu, Bilenko, and Lin 2017).

Sample reweighting (Ma et al. 2018; Hu, Koren, and Volinsky 2008; Yu, Bilenko, and Lin 2017; Song et al. 2020b) is one common strategy for addressing the challenge. Specifically, it increases the learning weight of positive (non-zero) samples which are very valuable owing to their scarcity (Hu, Koren, and Volinsky 2008), and/or decreases the weight of negative (zero) samples during model training (Yu, Bilenko, and Lin 2017). The solutions employed in the prior studies are to design a *weighting function* that maps the value or loss of a sample to its learning weight, to induce a recommender learn more or less of the sample when updating the model. For example, it is reasonable to increase the impact of positive samples more for conservative users than for exploratory users. However, because the fixed weighting function is used *throughout* the entire optimization process, the weighting scheme cannot address the different characteristics of users or POIs (Zhang et al. 2018).

To overcome the limitation of the *fixed* weighting function, we propose an *adaptive* weighting scheme, called PREMERE (**P**OI **RE**commendation with **ME**ta-learning based **RE**weighting). In short, PREMERE *learns to reweight samples* from data; for each sample, it produces the weight most suitable at the current stage of an optimization process. Meta-learning has been recently adopted to reweight samples for training image classifiers (Ren et al. 2018; Shu et al. 2019), and the weighting function exploits *meta-data* (i.e., an unbiased validation set) to inspect whether the reweighting is properly guiding the training process. However, applying meta-learning to sample reweighting for POI recommendation is very challenging because meta-data, which corresponds to precise user preferences or user trajectories, is practically very hard to obtain.[1]

Thus, the primary goal of this paper is to alleviate the lack of meta-data in POI recommendation for sample reweight-

*Jae-Gil Lee is the corresponding author.

[1] A clean validation set could be obtained by extensive user surveys or GPS data with high sampling rate, but either of them is not available in a large scale owing to cost and privacy issues.

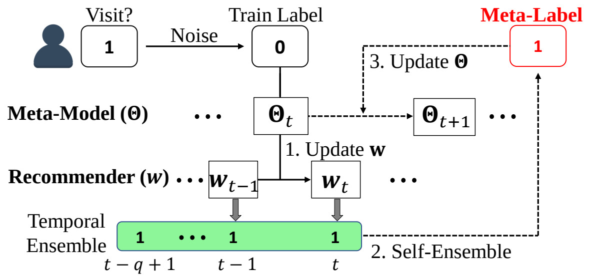


Figure 1: Overall procedure of PREMERE and its meta-data generation strategy through temporal ensembling.

ing based on meta-learning. Toward this goal, we exploit self-ensembling, or more specifically *temporal ensembling* (Laine and Aila 2017; Tarvainen and Valpola 2017), to generate meta-data. PREMERE involves two models: a recommender model and a meta-model (i.e., weighting function). The overall training procedure is illustrated in Figure 1. The recommender model parameterized by $\mathbf{w}$ infers the visit probability that each user will visit each POI at each epoch. If this prediction result for a user-POI pair is stable for a sufficient number of consecutive epochs, we regard that the prediction result tends to be the same as the ground truth, following the notion of temporal ensembling. Then, the user-POI pair annotated with the stable prediction result is confidently used as meta-data for training the meta-model parameterized by $\mathbf{\Theta}$. The parameter updates alternate between the recommender model and the meta-model.

Overall, the key contributions of this paper are summarized as follows:

- We propose PREMERE, a meta-learning approach to employ adaptive sample reweighting to mitigate the class imbalance and noisy values of the input data. Furthermore, a novel meta-model architecture, PREMERE-NET, is developed to cope with the class imbalance issue. To the best of our knowledge, our work is the first attempt to adopt meta-learning for POI recommendation.
- We alleviate the absence of the meta-data by its self-generation via temporal ensembling, which is used to update the PREMERE-NET model that understands the characteristics of users and POIs.
- The recommendation accuracy improves by 2.36–26.9% for three real-world benchmark datasets when a heuristic weighting function is replaced with PREMERE in the state-of-the-art recommender algorithms. Note that PREMERE can be applied to any recommender algorithm.

Preliminary

A *POI recommendation* algorithm receives users' check-in records and then provides a list of POIs that each user is likely to visit but has never visited before. Each check-in record contains a user u's visit to a POI l, along with the POI location and the visit timestamp. The records are aggregated to form a *user-POI check-in matrix* $\mathbf{V}$ in Definition 1.

Definition 1. Let N and M be the number of users and the number of POIs, respectively. Then, a *user-POI check-in matrix* $\mathbf{V} \in \mathbb{R}^{N\times M}$ is defined as a matrix, where each entry $v_{u,l}$ is the count of check-ins of the user u at the POI l. □

Then, we introduce a *user interest matrix* $\mathbf{X}$ in Definition 2, following the common problem setting in the recent literature for POI recommendation (Ma et al. 2018).

Definition 2. A *user interest matrix* $\mathbf{X} \in \mathbb{R}^{N\times M}$ is defined as the binarized user-POI check-in matrix for exhibiting users' interest in POIs, where $x_{u,l}$ is 1 if $v_{u,l} \geq 1$ and 0 otherwise. □

A recommender model, denoted as R parameterized by $\mathbf{w}$, receives a user interest matrix $\mathbf{X}$ and returns the user preferences $\hat{\mathbf{X}}$ over the POIs. Typically, R is trained to find out $\mathbf{w}^*$ which minimizes the loss function,

$$\begin{aligned}\mathcal{L}\left(\mathbf{X};\mathbf{w}\right) &= \frac{1}{N}\sum_{u=1}^{N}\ell\big(R(\mathbf{x}_u;\mathbf{w}),\mathbf{x}_u\big) = \frac{1}{N}\sum_{u=1}^{N}\ell(\hat{\mathbf{x}}_u,\mathbf{x}_u)\\ &= \frac{1}{|\Omega(\mathbf{X})|}\sum_{x\in\Omega(\mathbf{X})}\ell(\hat{x},x),\end{aligned} \tag{1}$$

where $\mathbf{x}_u$ (or $\hat{\mathbf{x}}_u$) denotes a user row vector of $\mathbf{X}$ (or $\hat{\mathbf{X}}$), x (or $\hat{x}$) denotes an element of $\mathbf{X}$ (or $\hat{\mathbf{X}}$), $\Omega(\mathbf{X})$ denotes the set of elements of $\mathbf{X}$, and $\ell(\hat{x},x)$ indicates any loss function, e.g., mean squared error $\|\hat{x}-x\|_2^2$. After training is complete, for a user u, the top-K POIs in $\hat{\mathbf{x}}_u \in \mathbb{R}^M$ except visited POIs are recommended to the user. However, owing to the class imbalance and noisy values in $\mathbf{X}$, naively minimizing the default loss in Eq. (1) may hinder the model R from understanding true user preferences and lead to unsatisfactory recommendation.

PREMERE: Meta-Reweight Methodology

In this section, we present the detailed procedure of PREMERE and the architecture of PREMERE-NET.

Problem Setting: In order to incorporate sample reweighting into the optimization process, we aim to train the model R by minimizing the extended loss function,

$$\mathcal{L}'(\mathbf{X};\mathbf{w},\Theta) = \frac{1}{|\Omega(\mathbf{X})|}\sum_{x\in\Omega(\mathbf{X})} f(I_x;\Theta)\ell\big(R(x;\mathbf{w}),x\big), \tag{2}$$

where Θ indicates the parameter of the meta-model f, and $I_{\mathbf{x}}$ is the set of relevant features (e.g., loss) needed for the meta-model. For each sample x, $f(I_x;\Theta)$ returns the best weight learned by meta-learning. Both R and f are supposed to be deep neural network (DNN) models. Overall, the optimal parameter $\mathbf{w}^*$ of R is determined as

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \mathcal{L}'\left(\mathbf{X};\mathbf{w},\Theta\right). \tag{3}$$

Overall Procedure

Each epoch of the training procedure for both the *recommender model* R and the *meta-model* f conducts the three steps, following the common procedure of meta-learning (Ren et al. 2018; Shu et al. 2019). The sequence of updates and data flow are illustrated in Figure 2. The

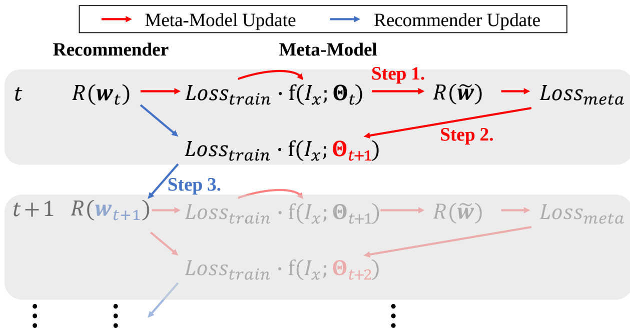


Figure 2: Sequence of the updates in a unified training procedure of PREMERE.

feedback is provided back and forth between the two models during the training procedure. For now, the meta-data is assumed to be given for ease of exposition, though it needs to be self-generated by PREMERE; the meta-data has the form of the user interest matrix in Definition 2, except that each element is a real number in $[0, 1]$ indicating a visit probability. $\mathbf{X}^{train}$ and $\mathbf{X}^{meta}$ denote the training and meta-data (validation) sets, respectively.

1. *Recommender model preliminary update*: In each iteration, a mini-batch of training samples, $\mathcal{B}_t = [\mathbf{x}_{1_{\mathcal{B}_t}}, \mathbf{x}_{2_{\mathcal{B}_t}}, \ldots, \mathbf{x}_{b_{\mathcal{B}_t}}] \in \mathbb{R}^{b \times M}$ is constructed by selecting b users from $\mathbf{X}^{train}$ uniformly at random, where b is the mini-batch size; then, the parameter $\mathbf{w}_t$ of the recommender model is updated using the *current* weighting function $f(\cdot; \Theta_t)$ to create a model with $\tilde{\mathbf{w}}$ by
$$\tilde{\mathbf{w}} = \mathbf{w}_t - \eta \nabla_{\mathbf{w}_t} \mathcal{L}' \left(\mathcal{B}_t; \mathbf{w}_t, \Theta_t\right)\big|_{\mathbf{w}_t}, \tag{4}$$
where η is a learning rate.
2. *Meta-model update*: A mini-batch of meta-data samples $\mathcal{M}_t = [\mathbf{x}^{meta}_{1_{\mathcal{M}_t}}, \mathbf{x}^{meta}_{2_{\mathcal{M}_t}}, \ldots, \mathbf{x}^{meta}_{b_{\mathcal{M}_t}}] \in \mathbb{R}^{b \times M}$ is constructed by selecting another b users from $(\mathbf{X}^{meta}_t - \mathcal{B}_t)$ uniformly at random. Note that $\tilde{\mathbf{w}}$ obtained by Eq. (4) is widely known as an inspection on the reweighting efficacy of the current meta-model (Ren et al. 2018; Shu et al. 2019). Hence, the feedback from $\tilde{\mathbf{w}}$ is exploited to update the parameter Θ_t of the meta-model by
$$\Theta_{t+1} = \Theta_t - \eta \nabla_{\Theta_t} \mathcal{L}\left(\mathcal{M}_t; \tilde{\mathbf{w}}\right)\big|_{\Theta_t}. \tag{5}$$
Ideally, because $\mathcal{M}_t$ contains true user interest, the default loss in Eq. (1) is adopted for training in general.
3. *Recommender model update*: Finally, the parameter $\mathbf{w}_t$ of the recommender model is updated using the *updated* weighting function $f(\cdot; \Theta_{t+1})$ by
$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla_{\mathbf{w}_t} \mathcal{L}' \left(\mathcal{B}_t; \mathbf{w}_t, \Theta_{t+1}\right)\big|_{\mathbf{w}_t}. \tag{6}$$

However, we argue that above meta-learning pipeline is infeasible in recommendation tasks because of the absence of $\mathcal{M}_t$ and severe class imbalance. Therefore, in the remaining of this section, we discuss (1) our temporal ensembling approach to solve the former; and (2) our novel meta-model architecture PREMERE-NET to treat the latter.

Meta-Data Generation via Temporal Ensembling

Our rationale behind the generation of meta-data is based on that POI recommendation resembles semi-supervised learning, in considering that only a few samples are labeled but most of them are not. In POI recommendation, recorded check-ins, which account for a small fraction (i.e., 0.1%) of the entire dataset, correspond to the labeled set, and missing check-ins, which are not reflected in the dataset, correspond to the unlabeled set. A common philosophy of semi-supervised learning is to infer the labels of unlabeled samples for use in training a target model. Thus, in this study, temporal ensembling (Laine and Aila 2017), which is shown to be successful in semi-supervised learning, is employed to infer the *confidence* of noisy negative samples.

For temporal ensembling, a *mean-teacher network* (Tarvainen and Valpola 2017), which is an ensemble of the current and earlier versions of a target model network, is formulated by
$$\mathbf{w}'_{t+1} = \alpha \mathbf{w}'_t + (1 - \alpha)\mathbf{w}_t, \tag{7}$$
where α is a hyperparameter for an exponential moving average. Then, for each negative sample in the training set, we calculate the variance of q consecutive ensemble prediction results to quantify its prediction stability (i.e., confidence), as shown in Definition 3.

Definition 3. Let $x_{u,l}$ be a user u's interest on a POI l and $R(x_{u,l}; \mathbf{w})$ be the predicted visit probability for $x_{u,l}$ obtained by the model with $\mathbf{w}$. Suppose that a history of q recent predictions is maintained as $\mathcal{H}$. Then, the prediction stability S of $x_{u,l}$ is formulated by
$$S_t(x_{u,l}) = var(\mathcal{H}_t(x_{u,l}; q)),$$
$$\text{where } \mathcal{H}_t(x; q) = \{R(x; \mathbf{w}'_{t-q+1}), \ldots, R(x; \mathbf{w}'_t)\}. \ \square \tag{8}$$

Then, a meta-data $\mathbf{X}^{meta}_t$ at time t is constructed depending on the value and prediction stability of each entry of $\mathbf{X}^{train}$, as stated in Definition 4.

Definition 4. A meta-data $\mathbf{X}^{meta}_t \in \mathbb{R}^{N \times M}$ is a matrix with each entry defined as
$$x^{meta}_{u,l} = \begin{cases} 1 & \text{if } \ x_{u,l} = 1 \\ \overline{\mathcal{H}_t}(x_{u,l}; q) & \text{if } \ x_{u,l} = 0 \land S_t(x_{u,l}) \leq \epsilon, \end{cases} \tag{9}$$
where $\overline{\mathcal{H}}$ is the mean of the q prediction results (i.e., meta-label), and ϵ is the hyperparameter for stable prediction. $\square$

When a mini-batch $\mathcal{M}_t$ of meta-data is constructed, only the entries in $\mathbf{X}^{meta}_t$ defined by Eq. (9) are considered. In addition, PREMERE iteratively refines $\mathbf{X}^{meta}_t$ every epoch, which is considered as a learning cycle to measure the model changes (Song et al. 2020a).

PREMERE-NET Architecture

The architecture of PREMERE-NET is improved along two directions, as shown in Figure 3. First, the meta-model input is enriched to include the context data about users and POIs. Second, the flow of the meta-model is extended to have a branch to handle the rare positive (i.e., minor class) samples separately, which is shown in the grey upper part.

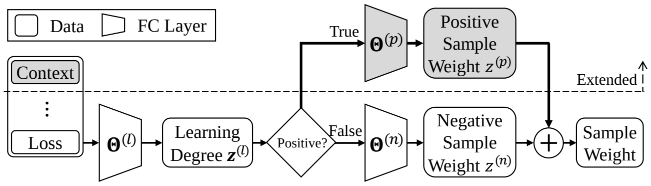


Figure 3: PREMERE-NET meta-model architecture.

Context Data For each training sample $x_{u,l}$, the context data entails some characteristics or preferences of the user u over various aspects of the POI l. Four types of the context data are designed in this work based on their availability. Here, $\mathbf{x}_u^{\mathbf{1}}$ denotes the set of the POIs visited by u.

- The *user visit entropy* represents the concentration of user visits and is formulated using the information entropy (Chandler and Percus 1988), $-\sum_{l \in \Omega(\mathbf{x}_u^{\mathbf{1}})} \mathrm{P}(l) \log \mathrm{P}(l)$, where $\mathrm{P}(l)$ is the proportion of the visits at l out of all visits by u. The lower the entropy value is, the more concentrated the user's visits are.
- The *geographical similarity* is formulated by the average of the distance from l to the other POIs visited by u, $\frac{1}{|\Omega(\mathbf{x}_u^{\mathbf{1}})|} \sum_{l' \in \Omega(\mathbf{x}_u^{\mathbf{1}})} \exp(-\mathrm{dist}(l, l')^2)$, where $\mathrm{dist}(\cdot,\cdot)$ is the Euclidean distance between two POIs. It reflects the first law of geography: "everything is related to everything else, but near things are more related than distant things."
- The *temporal similarity* is formulated by the average of the peak-time similarity between l and the other POIs visited by u, $\frac{1}{|\Omega(\mathbf{x}_u^{\mathbf{1}})|} \sum_{l' \in \Omega(\mathbf{x}_u^{\mathbf{1}})} \cos(\mathbf{h}_l, \mathbf{h}_{l'})$, where $\cos(\cdot,\cdot)$ is the cosine similarity between two vectors, and $\mathbf{h}_l$ is a 24-dimensional vector that contains the visit proportion by all users at l in each hourly interval.
- The *check-in count* is the number of the visits to l by u.

Components PREMERE-NET receives the loss of the recommender model and the context for a given sample, and returns the sample weight as the result. As shown in Figure 3, the upper part is added to the conventional meta-model architecture (Shu et al. 2019). In support of the fundamental difference between positive and negative samples, negative ones are handled by the conventional (lower) part, while positive ones are handled by the extended (upper) part.

There are *three* fully-connected layers in PREMERE-NET. First, the *learning degree* layer $\Theta^{(l)}$ adaptively infers a latent embedding $\mathbf{z}^{(l)}$, which is intended to demonstrate how much a given sample contributes to the current training,

$$\mathbf{z}^{(l)} = \sigma(\Theta^{(l)} \cdot I_{x_{u,l}} + \mathbf{b}^{(l)}), \quad \text{where } I_{x_{u,l}} = [\mathcal{L}(x_{u,l}; \mathbf{w}_t), \mathcal{C}_{x_{u,l}}]. \tag{10}$$

Here, σ is an activation function, $\mathbf{b}^{(l)}$ is the bias of the layer, and $\mathcal{C}_{x_{u,l}}$ is the context data for $x_{u,l}$. Then, the embedding $\mathbf{z}^{(l)}$ is fed to either the *positive boosting* layer $\Theta^{(p)}$ or the *negative reweighting* layer $\Theta^{(n)}$ depending on whether a given sample is positive or not. The former is intended to handle the rarity of positive samples, and the latter is aimed at dealing with the noisiness of negative samples. Last, the sample weight is determined as

$$f(I_{x_{u,l}}; \Theta) = \begin{cases} \sigma(\Theta^{(p)} \cdot \mathbf{z}^{(l)}) & \text{if} \quad x_{u,l} = 1 \\ \sigma(\Theta^{(n)} \cdot \mathbf{z}^{(l)}) & \text{otherwise.} \end{cases} \tag{11}$$

By virtue of this architectural extension, PREMERE-NET not only handles the different characteristics of users and POIs, but also relieves the class imbalance.

Algorithm 1 PREMERE Training

INPUT: *epochs*, $\mathbf{X}^{train}$: user interest matrix, b: batch size, ϵ: consistency threshold, q: history length
OUTPUT: $\mathbf{w}^*$: trained model parameter

1: $t \leftarrow 1$; $\mathbf{w}_t, \Theta_t \leftarrow$ Initialize model parameters;
2: $\mathbf{X}_t^{meta} \leftarrow$ Add positive samples from $\mathbf{X}^{train}$;
3: **for** $e = 1$ **to** $epochs$ **do**
4: **for** $iteration = 1$ **to** N/b **do**
5: /* **Meta-Model Update** */
6: $\mathcal{B}_t \leftarrow$ Sample a train mini-batch from $\mathbf{X}^{train}$;
7: $\tilde{\mathbf{w}} \leftarrow \mathbf{w}_t - \eta \nabla \mathcal{L}' (\mathcal{B}_t; \mathbf{w}_t, \Theta_t))$; /* By Eq. (4) */
8: $\mathcal{M}_t \leftarrow$ Sample a meta mini-batch from $\mathbf{X}_t^{meta}$;
9: $\Theta_{t+1} \leftarrow \Theta_t - \eta \nabla \mathcal{L} (\mathcal{M}_t; \tilde{\mathbf{w}})$; /* By Eq. (5) */
10: /* **Recommender Update** */
11: $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta \nabla \mathcal{L}' (\mathcal{B}_t; \mathbf{w}_t, \Theta_{t+1})$; /* By Eq. (6) */
12: $t \leftarrow t + 1$;
13: /* **Generate Meta-Data** */
14: $\mathbf{X}_t^{meta} \leftarrow$ Construct a meta-data by Def. 4;
15: **return** $\mathbf{w}^* \leftarrow \mathbf{w}_t$;

Training Algorithm Pseudocode

The unified training procedure of PREMERE is described in Algorithm 1, which is self-explanatory. It receives the user-interest matrix $\mathbf{X}^{train}$ for the training set (as well as the associated context data $\mathcal{C}$) and produces the optimal parameter $\mathbf{w}^*$ of the recommender model. The algorithm conducts the three steps: (1) recommender model preliminary update (Line 7), (2) meta-model update (Line 9), and (3) recommender model update (Line 11). After these model updates, the meta-data for the next epoch is newly generated by Definition 4 (Line 14). This procedure repeats for a given number of epochs. Once the training is done, the user preference over the POIs is inferred by $R(\mathbf{x}_u; \mathbf{w}^*)$, and the top-$K$ unvisited POIs are recommended to each user.

Evaluation

In this section, we verify that (1) PREMERE improves the performance of state-of-the-art POI recommender models by replacing their heuristic sample reweighting with our meta-reweighting; (2) the extension of the meta-model architecture is effective for meta-reweighting. The source code is available at https://github.com/kaist-dmlab/PREMERE.

Experiment Setting

Datasets We used *three* popular benchmark datasets, Gowalla (Liu et al. 2017), Foursquare (Yang, Zhang, and Qu 2016), and Yelp (Liu et al. 2017), which are commonly used in the POI recommendation literature (Zhou et al. 2019; Ma

Dataset	Users	POIs	Check-ins	Scarcity
Gowalla	18,737	32,510	1,278,274	99.865%
Foursquare	24,941	28,593	1,196,248	99.900%
Yelp	30,887	18,995	860,888	99.860%

Table 1: Profiles of the three real-world datasets.

Method	CDAE+Heuristic vs. CDAE+PREMERE	SAE-NAD vs. SAE-NAD+PREMERE
Gowalla	7.21%	$\mathbf{26.9}\%$
Foursquare	3.88%	2.36%
Yelp	5.79%	7.50%

Table 2: Precision@5 improvements by PREMERE in Figures 4, 5, and 6.

et al. 2018). Table 1 shows the profile of these three datasets. According to usual data preprocessing (Liu et al. 2017), we excluded the users whose check-in count is less than 10 in Foursquare and Yelp and 15 in Gowalla; and the POIs whose visitor count is less than 10 in all datasets. We randomly selected 80% of check-ins as the training set and used the rest 20% of check-ins as the test set in each dataset.

Evaluation Metrics We used three widely-accepted metrics. The *precision*$@K$ is the proportion of recommended POIs in the top-K set that are visited, and the *recall*$@K$ is the proportion of actually visited POIs found in the top-K recommendations. The *MAP*$@K$ is the mean of precision values at all ranks where actually visited POIs are found. We varied the number of recommendations $K \in \{5, 10, 20, 50\}$. For the reliability of evaluation, we repeated every task *five* times and reported the average value.

Baseline Recommenders We compared PREMERE with 11 existing POI recommender algorithms. Owing to the lack of space, the detailed discussion of seven *non-DNN-based* algorithms, MGMPFM (Cheng et al. 2012), LFBCA (Wang, Terrovitis, and Mamoulis 2013), USG (Ye et al. 2011), iGSLR (Zhang and Chow 2013), LORE (Zhang, Chow, and Li 2014), IRenMF (Liu et al. 2014), and RankGeoFM (Li et al. 2015), is deferred to the supplementary material. Here, we focus on the following four *DNN-based* algorithms.

- **CDAE** (Wu et al. 2016) is based on a collaborative denoising auto-encoder.
- **PACE** (Yang et al. 2017) is a Skip-gram (Mikolov et al. 2013)-based semi-supervised learning model to learn user preference and various contexts.
- **SAE-NAD** (Ma et al. 2018) is a stacked autoencoder (SAE)-based model to learn latent user preference in the self-attentive encoder with an attention structure (Yang et al. 2016) and to cover additional geographical influence in the neighbor-aware decoder.
- **APOIR** (Zhou et al. 2019) is a generative adversarial network (GAN) (Goodfellow et al. 2014)-based model to learn user latent preference distribution with social and geographical influence simultaneously.

Reweighting Schemes To evaluate the effectiveness of the reweighting scheme itself, we injected the following three reweighting schemes into CDAE and SAE-NAD.

- The *default* scheme involves *no* sample reweighting.
- The *heuristic* scheme involves positive sample upweighting based on a *pre-defined* heuristic function of SAE-NAD. It is defined only for positive samples as

$$f(x_{u,l}) = 1 + \gamma \log(1 + v_{u,l}/\varepsilon), \tag{12}$$

where $v_{u,l}$ is the check-in count, and γ and ε are the hyperparameters of SAE-NAD. Note that the output for a specific sample does not change throughout the training.

- PREMERE is our proposed meta-reweighting scheme.

Different reweighting schemes were tested for CDAE and SAE-NAD, where the results reported in the original papers were reproduced in our environment. CDAE, CDAE+Heuristic, and **CDAE+PREMERE** correspond to the three variations of CDAE; SAE-NAD−Heuristic, SAE-NAD, and **SAE-NAD+PREMERE** correspond to the three variations of SAE-NAD because SAE-NAD is already equipped with the heuristic scheme.

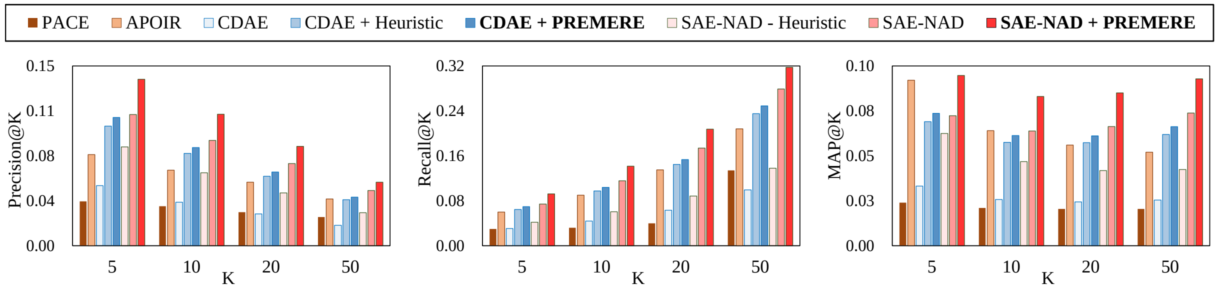
Configuration We used Adam (Kingma and Ba 2015) with a learning rate $\eta = 0.001$ and a weight decay 0.001. Regarding three hyperparameter of PREMERE, we fixed the moving average weight $\alpha = 0.95$ and the history length $q = 10$, which are known as the best-performing values from relevant studies (Tarvainen and Valpola 2017; Song, Kim, and Lee 2019); the stability threshold ϵ was set to be $0.25 * \overline{\mathcal{H}}(x; q)$, where 0.25 is the upper bound of Eq. (8). Our implementation was written using PyTorch and tested on Nvidia Tesla V100. For the existing algorithms, we followed the best hyperparameter setting suggested in the original papers and conducted additional grid search to find the best values for those not specified. Overall, we did our best to achieve the highest accuracy for all compared algorithms.

Result Highlight and Summary

The meta-reweighting scheme of PREMERE improved the performance of the two recommender algorithms in all three datasets, as shown in Figures 4, 5, and 6. Accordingly, the performance boosted by PREMERE exceeded the previously-known best performance obtained by SAE-NAD. Table 2 quantifies the *precision@5* improvements of CDAE+PREMERE and SAE-NAD+PREMERE over CDAE+Heuristic and SAE-NAD, respectively.

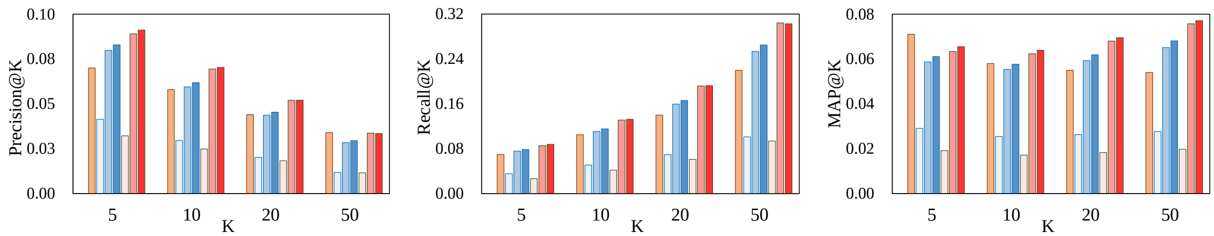
The superior performance of PREMERE is attributed to two technical innovations: (1) the meta-reweighting scheme handles noise in the negative samples as well as scarcity of the positive samples whereas the heuristic reweighting scheme handles the latter only; (2) the meta-reweighting scheme is adaptive to the training progress and context data whereas the heuristic reweighting scheme is not.

Table 3 shows the *precision@5* results of seven non-DNN-based algorithms. PREMERE far exceeded the performance of these algorithms. See the supplementary material for details.



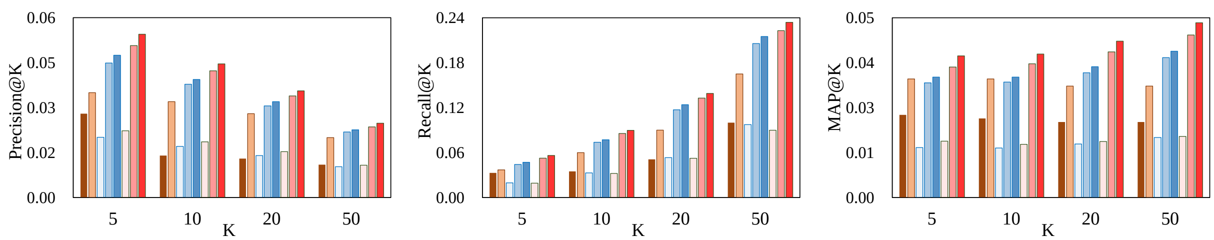
(a) Precision. (b) Recall. (c) MAP.

Figure 4: Performance comparison among DNN-based algorithms for the Gowalla dataset.[2]



(a) Precision. (b) Recall. (c) MAP.

Figure 5: Performance comparison among DNN-based algorithms for the Foursquare dataset.



(a) Precision. (b) Recall. (c) MAP.

Figure 6: Performance comparison among DNN-based algorithms for the Yelp dataset.

Method	MGMPFM	LFBCA	USG	iGSLR	LORE	IRenMF	RankGeoFM	**SAE-NAD+PREMERE**
Gowalla	0.0245	0.0610	0.0602	0.0291	0.0417	0.0663	0.0692	**0.1389**
Foursquare	0.0321	–	–	–	–	0.0604	0.0623	**0.0911**
Yelp	0.0160	0.0231	0.0264	0.0125	0.0241	0.0280	0.0331	**0.0545**

Table 3: Precision@5 results of seven non-DNN-based algorithms as well as PREMERE.

Relationship with Model Capacity As shown in Table 2, SAE-NAD generally gained a larger performance enhancement by PREMERE than CDAE owing to SAE-NAD's higher model complexity. While CDAE consists of a simple autoencoder structure, SAE-NAD additionally employs an attention mechanism (Luong, Pham, and Manning 2015; Pei et al. 2017) and geographical contexts, which offer higher capacity to understand complex user preference. Therefore, it is expected that a more powerful model can benefit from PREMERE more.

Relationship with Dataset PREMERE achieved significant improvement especially when it was incorporated into SAE-NAD and tested for the Gowalla dataset (i.e., Figure 4). This inconsistency among the datasets is attributed to the geographic distribution of check-ins. While the check-ins in Foursquare and Yelp were spread within the U.S. or a few cities in Europe, those in Gowalla were spread in the entire world. Consequently, as shown in Table 4, the proportion of other reachable POIs from each POI is the smallest in Gowalla. As SAE-NAD considers the geographic context for POI recommendation, PREMERE helps SAE-NAD effectively prune the far-away POIs, which are very unlikely to visit, by sample reweighting.

Ablation Studies

To examine the effect of temporal ensembling and PREMERE-NET components, we conducted ablation studies by incorporating the following PREMERE variants into SAE-NAD. Table 5 shows the results for the Gowalla dataset.

[2]The results of PACE and APOIR in Figures 4, 5, and 6 are borrowed from the original papers.

Distribution	Data Collection	% of POIs	
		$\leq 10\,\text{km}$	$\geq 1000\,\text{km}$
Gowalla	World-Wide	3.8%	79.4%
Foursquare	U.S. Mainland	6.1%	67.0%
Yelp	A Few Cities	10.3%	34.4%

Table 4: Geographic distribution of check-ins.

Metric	Precision@5	Recall@5	MAP@5
PREMERE	**0.1389**	**0.0923**	**0.0947**
No Self-Generation	0.1259	0.0879	0.0807
No Positive Boosting	0.0999	0.0654	0.0678
No Context Data	0.0818	0.0589	0.0536

Table 5: Ablation study results for the Gowalla dataset.

- *No meta-data self-generation*: In Definition 3, only the first case is enabled whereas the second case is disabled.
- *No positive boosting*: In Figure 3 for the PREMERE-NET architecture, the path to the positive boosting layer for Θ_p (i.e., above the dashed line) is disabled.
- *No context data*: In Figure 3, the context data is also removed from the above variant (no positive boosting). Then, this variant becomes identical to Meta-Weight-Net (Shu et al. 2019).

Effect of Meta-Data Self-Generation The first variant does not use the meta-data for negative samples, which account for 99.9% of the entire dataset. As a result, the performance degrades by 9.9%, 4.8%, and 14.8% in terms of precision@5, recall@5, and MAP@5, respectively. Thus, generating the meta-data for negative samples is important, and our self-generation approach is reliable.

Effect of Positive Sample Boosting The second variant does not deal with the rarity of positive samples. As a result, the performance degrades by 28.0%, 29.1%, and 28.4% in terms of precision@5, recall@5, and MAP@5, respectively. Thus, handling the class imbalance is essential, and the PREMERE-NET architecture is proven to be effective.

Effect of Context Data The third variant may produces a sub-optimal weight, especially when the optimal weight cannot be entirely determined by the loss, i.e., when the same loss is derived for the users or POIs of different characteristics. As a result, the performance degrades by 42.1%, 36.2%, and 43.4% in terms of precision@5, recall@5, and MAP@5, respectively. Thus, utilizing context data is important, and our design is proven to be sufficient.

Related Work

Several POI recommender algorithms, such as PACE (Yang et al. 2017), SAE-NAD (Ma et al. 2018), and APOIR (Zhou et al. 2019), have been developed using a DNN. These DNN-based algorithms are successfully shown to outperform traditional algorithms. Since we focus on sample reweighting to further improve the performance of a DNN-based algorithm, the detailed description of such algorithms is omitted. Refer to an extensive survey (Chen et al. 2020) on recent POI recommender algorithms. Meanwhile, reweighting sample importance has been an active research topic because of its importance in improving the performance.

Sample Reweighting in Recommendation

To make a recommender model understand user preference more precisely, various studies have suggested reweighting sample weights during model training (Hu, Koren, and Volinsky 2008; Ma et al. 2018; Zhang et al. 2018). Owing to the imbalance of check-ins in POI recommendation, SAE-NAD emphasizes the check-ins (positive samples) by assigning a higher weight by Eq. (12), which is one of the most popular weighting schemes (Hu, Koren, and Volinsky 2008). Recently, in movie recommendation with explicit user ratings, a self-paced learning (Kumar, Packer, and Koller 2010)-based reweighting strategy was proposed to select the samples for reweighting under a predefined weighting scheme (Zhang et al. 2018). Nevertheless, previous studies rely upon a *fixed* weighting function throughout the entire training process, which does not reflect different characteristics of users or POIs.

Meta-Learning-Based Sample Reweighting

Recently, meta-learning has been started to be adopted in adaptive sample reweighting (Ren et al. 2018; Shu et al. 2019) to overcome the limitation of the fixed weighting function. These meta-reweighting strategies alternate between ameliorating a reweighting strategy and updating a target model. In L2RW (Ren et al. 2018), the set of sample weights is defined to minimize the loss on a mini-batch of a clean validation set (meta-data). In Meta-Weight-Net (Shu et al. 2019), a multilayer perceptron (MLP) with one hidden layer is used for explicitly modeling a weighting function, where the MLP is trained using a clean validation set (meta-data). Nevertheless, previous studies *all* require meta-data to learn sample reweighting, which is practically infeasible to acquire in recommender environments.

Conclusion

In this paper, we proposed PREMERE, a novel meta-learning-based sample reweighting scheme for POI recommendation. The meta-model architecture, PREMERE-NET, is extended to use the well-designed context data and a separate flow of handling positive samples. In addition, the absence of meta-data, which was a critical problem of POI recommendation, is solved by the self-generation technique via temporal ensembling. Extensive evaluation was conducted by incorporating PREMERE into DNN-based POI recommender algorithms. The meta-reweighting of PREMERE significantly improved the recommendation performance by up to 26.9% in terms of the precision@5 compared with the heuristic reweighting. Overall, PREMERE can be applied to any DNN architecture and is expected to raise the POI recommendation performance.

Acknowledgements

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-00862, DB4DL: High-Usability and Performance In-Memory Distributed DBMS for Deep Learning).

References

Chandler, D.; and Percus, J. K. 1988. Introduction to modern statistical mechanics. *Physics Today* 41(12): 114.

Chen, L.; Shang, S.; Yang, C.; and Li, J. 2020. Spatial keyword search: A survey. *GeoInformatica* 24(1): 85–106.

Cheng, C.; Yang, H.; King, I.; and Lyu, M. R. 2012. Fused matrix factorization with geographical and social influence in location-based social networks. In *AAAI*, 17–23.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NeurIPS*, 2672–2680.

Hu, Y.; Koren, Y.; and Volinsky, C. 2008. Collaborative filtering for implicit feedback datasets. In *ICDM*, 263–272.

Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Kumar, M. P.; Packer, B.; and Koller, D. 2010. Self-paced learning for latent variable models. In *NeurIPS*, 1189–1197.

Laine, S.; and Aila, T. 2017. Temporal ensembling for semi-supervised learning. In *ICLR*.

Li, X.; Cong, G.; Li, X.-L.; Pham, T.-A. N.; and Krishnaswamy, S. 2015. Rank-GeoFM: A ranking based geographical factorization method for point of interest recommendation. In *SIGIR*, 433–442.

Lian, D.; Zhao, C.; Xie, X.; Sun, G.; Chen, E.; and Rui, Y. 2014. GeoMF: Joint geographical modeling and matrix factorization for point-of-interest recommendation. In *KDD*, 831–840.

Liu, Y.; Pham, T.-A. N.; Cong, G.; and Yuan, Q. 2017. An experimental evaluation of point-of-interest recommendation in location-based social networks. *PVLDB* 10(10): 1010–1021.

Liu, Y.; Wei, W.; Sun, A.; and Miao, C. 2014. Exploiting geographical neighborhood characteristics for location recommendation. In *CIKM*, 739–748.

Luong, M.-T.; Pham, H.; and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP*, 1412–1421.

Ma, C.; Zhang, Y.; Wang, Q.; and Liu, X. 2018. Point-of-interest recommendation: Exploiting self-attentive autoencoders with neighbor-aware influence. In *CIKM*, 697–706.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, 3111–3119.

Pei, W.; Yang, J.; Sun, Z.; Zhang, J.; Bozzon, A.; and Tax, D. M. 2017. Interacting attention-gated recurrent networks for recommendation. In *CIKM*, 1459–1468.

Ren, M.; Zeng, W.; Yang, B.; and Urtasun, R. 2018. Learning to reweight examples for robust deep learning. In *ICML*.

Shu, J.; Xie, Q.; Yi, L.; Zhao, Q.; Zhou, S.; Xu, Z.; and Meng, D. 2019. Meta-Weight-Net: Learning an explicit mapping for sample weighting. In *NeurIPS*, 1919–1930.

Song, H.; Kim, M.; Kim, S.; and Lee, J.-G. 2020a. Carpe diem, seize the samples uncertain "at the moment" for adaptive batch selection. In *CIKM*.

Song, H.; Kim, M.; and Lee, J.-G. 2019. SELFIE: Refurbishing unclean samples for robust deep learning. In *ICML*, 5907–5915.

Song, H.; Kim, S.; Kim, M.; and Lee, J.-G. 2020b. Ada-Boundary: Accelerating DNN training via adaptive boundary batch selection. *Machine Learning* 109(9): 1837–1853.

Tarvainen, A.; and Valpola, H. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 1195–1204.

Wang, H.; Terrovitis, M.; and Mamoulis, N. 2013. Location recommendation in location-based social networks using user check-in data. In *SIGSPATIAL*, 374–383.

Wu, Y.; DuBois, C.; Zheng, A. X.; and Ester, M. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *ICWSM*, 153–162.

Yang, C.; Bai, L.; Zhang, C.; Yuan, Q.; and Han, J. 2017. Bridging collaborative filtering and semi-supervised learning: A neural approach for POI recommendation. In *KDD*, 1245–1254.

Yang, D.; Zhang, D.; and Qu, B. 2016. Participatory cultural mapping based on collective behavior data in location-based social networks. *ACM Transactions on Intelligent Systems and Technology* 7(3): 30.

Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; and Hovy, E. 2016. Hierarchical attention networks for document classification. In *NAACL*, 1480–1489.

Ye, M.; Yin, P.; Lee, W.-C.; and Lee, D.-L. 2011. Exploiting geographical influence for collaborative point-of-interest recommendation. In *SIGIR*, 325–334.

Yu, H.-F.; Bilenko, M.; and Lin, C.-J. 2017. Selection of negative samples for one-class matrix factorization. In *SDM*, 363–371.

Zhang, J.-D.; and Chow, C.-Y. 2013. iGSLR: Personalized geo-social location recommendation: a kernel density estimation approach. In *SIGSPATIAL*, 334–343.

Zhang, J.-D.; Chow, C.-Y.; and Li, Y. 2014. LORE: Exploiting sequential influence for location recommendations. In *SIGSPATIAL*, 103–112.

Zhang, Y.; Wang, H.; Lian, D.; Tsang, I. W.; Yin, H.; and Yang, G. 2018. Discrete ranking-based matrix factorization with self-paced learning. In *KDD*, 2758–2767.

Zhou, F.; Yin, R.; Zhang, K.; Trajcevski, G.; Zhong, T.; and Wu, J. 2019. Adversarial point-of-interest recommendation. In *TheWebConf*, 3462–34618.