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 (POI REcommendation with MEta-learning based REweighting). 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-
We propose PREMERE, a meta-learning approach to reweighting 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 \( w \) 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 \( \Theta \). 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 \( x_{u,l} = 1 \) if \( v_{u,l} \geq 1 \) and 0 otherwise.

A recommender model, denoted as \( R \) parameterized by \( w \), 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 \( w^* \) which minimizes the loss function,

\[
\mathcal{L}(\mathbf{X}; w) = \frac{1}{N} \sum_{u=1}^{N} \ell(R(\mathbf{x}_u; w), x_u) = \frac{1}{N} \sum_{u=1}^{N} \ell(\mathbf{\hat{x}}_u, x_u)
\]

where \( x_u \) (or \( \mathbf{x}_u \)) 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 \( \|\hat{x} - x\|^2 \). After training is complete, for a user \( u \), the top-K POIs in \( \hat{x}_u \in \mathbb{R}^{M} \) 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,

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

where \( \Theta \) indicates the parameter of the meta-model \( f \), and \( I_x \) 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 \( w^* \) of \( R \) is determined as

\[
w^* = \arg \min_w \mathcal{L}'(\mathbf{X}; w, \Theta). \tag{3}
\]

**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
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

\[ w_{t+1} = \alpha w_t + (1 - \alpha)w_t, \]

where \( \alpha \) 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 \( R(x_{u,l}; w) \) be the predicted visit probability for \( x_{u,l} \) obtained by the model with \( w \). 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

\[ S_t(x_{u,l}) = \text{var}(\mathcal{H}_t(x_{u,l}; q)), \]

where \( \mathcal{H}_t(x; q) = \{R(x; w_{t-1}^{l-1}), \ldots, R(x; w_l^t)\} \).

Then, a meta-data \( X_t \) at time \( t \) is constructed depending on the value and prediction stability of each entry of \( X_t \), as stated in Definition 4.

Definition 4. A meta-data \( X_t \) is a matrix with each entry defined as

\[ x_{u,l} = \begin{cases} 1 & \text{if } x_{u,l} = 1 \text{ and } \mathcal{H}_t(x_{u,l}; q) \leq \epsilon, \\ \mathcal{H}_t(x_{u,l}; q) & \text{else,} \end{cases} \]

where \( \mathcal{H} \) is the mean of the \( q \) prediction results (i.e., meta-label), and \( \epsilon \) is the hyperparameter for stable prediction.

When a mini-batch \( M_t \) of meta-data is constructed, only the entries in \( X_t \) are considered. In addition, PREMERE iteratively refines \( X_t \) 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.
Context Data  For each training sample \( x_{u,t} \), 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, \( x^C_{u,l} \) 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), 

\[
- \sum_{l \in \Omega} P(l) \log P(l),
\]

where \( P(l) \) 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 \( u \), 

\[
\frac{1}{|\Omega(x^C_{u,l})|} \sum_{l' \in \Omega(x^C_{u,l})} \exp(-\text{dist}(l, l')^2),
\]

where \( \text{dist}(\cdot, \cdot) \) 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 \( u \), 

\[
\frac{1}{|\Omega(x^C_{u,l})|} \sum_{l' \in \Omega(x^C_{u,l})} \cos(h_l, h_{l'}),
\]

where \( \cos(\cdot, \cdot) \) is the cosine similarity between two vectors, and \( h_l \) 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 \( z^{(l)} \), which is intended to demonstrate how much a given sample contributes to the current training.

\[
z^{(l)} = \sigma(\Theta^{(l)} \cdot I_{x_{u,l}} + b^{(l)}),
\]

where 

\[
I_{x_{u,l}} = [\mathcal{L}(x_{u,d}; w_t), C_{x_{u,l}}].
\]

Here, \( \sigma \) is an activation function, \( b^{(l)} \) is the bias of the layer, and \( C_{x_{u,l}} \) is the context data for \( x_{u,t} \). Then, the embedding \( z^{(l)} \) 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

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

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

Training Algorithm Pseudocode  The unified training procedure of PREMERE is described in Algorithm 1, which is self-explanatory. It receives the user-interest matrix \( X^{\text{train}} \) for the training set (as well as the associated context data \( C \)) and produces the optimal parameter \( w^* \) 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 \( R(x_u; w^*) \), 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

Algorithm 1 PREMERE Training

INPUT: epochs, \( X^{\text{train}} \): user interest matrix, \( b \): batch size, \( c \): consistency threshold, \( q \): history length

OUTPUT: \( w^* \): trained model parameter

1: \( t \leftarrow 1; \ w_t, \Theta_t \leftarrow \text{Initialize model parameters}; \)
2: \( X_t^{\text{meta}} \leftarrow \text{Add positive samples from } X^{\text{train}}; \)
3: for \( e = 1 \) to epochs do
4:     for iteration = 1 to \( N/b \) do
5:         \(/^* \text{Meta-Model Update } */
6:         \( B_t \leftarrow \text{Sample a train mini-batch from } X^{\text{train}}; \)
7:         \( \tilde{w} \leftarrow w_t - \eta \mathcal{L}(B_t; w_t, \Theta_t); \) /* By Eq. (4) */
8:         \( M_t \leftarrow \text{Sample a meta mini-batch from } X_t^{\text{meta}}; \)
9:         \( \Theta_{t+1} \leftarrow \Theta_t - \eta \mathcal{L}(M_t; \tilde{w}); \) /* By Eq. (5) */
10: \(/^* \text{Recommender Update } */
11: \( w_{t+1} \leftarrow w_t - \eta \mathcal{L}(B_t; w_t, \Theta_{t+1}); \) /* By Eq. (6) */
12: \( t \leftarrow t + 1; \)
13: \(/^* \text{Generate Meta-Data } */
14: \( X_{t+1}^{\text{meta}} \leftarrow \text{Construct a meta-data by Def. 4}; \)
15: \( \text{return } w^* \leftarrow w_t; \)
Table 1: Profiles of the three real-world datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Users</th>
<th>POIs</th>
<th>Check-ins</th>
<th>Scarcity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gowalla</td>
<td>18,737</td>
<td>32,510</td>
<td>1,278,274</td>
<td>99.865%</td>
</tr>
<tr>
<td>Foursquare</td>
<td>24,941</td>
<td>28,593</td>
<td>1,196,248</td>
<td>99.900%</td>
</tr>
<tr>
<td>Yelp</td>
<td>30,887</td>
<td>18,995</td>
<td>860,888</td>
<td>99.860%</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Gowalla</th>
<th>Foursquare</th>
<th>Yelp</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDAE+Heuristic vs.</td>
<td>7.21%</td>
<td>26.9%</td>
<td>5.79%</td>
</tr>
<tr>
<td>SAE-NAD vs.</td>
<td>2.36%</td>
<td>7.50%</td>
<td></td>
</tr>
<tr>
<td>CDAE+PREMERE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAE-NAD+PREMERE</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

where $v_{u,t}$ is the check-in count, and $\gamma$ and $\varepsilon$ 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 hyperparameters 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 $\epsilon$ was set to be $0.25 * \log(1 + v_{u,t}/\varepsilon)$, 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.
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 components, we conducted ablation studies by incorporating the following PREMERE variants into SAE-NAD. Table 5 shows the results for the Gowalla dataset.
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 our self-generation approach is reliable.

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
Chen, L.; Shang, S.; Yang, C.; and Li, J. 2020. Spatial key-
Cheng, C.; Yang, H.; King, I.; and Lyu, M. R. 2012. Fused
matrix factorization with geographical and social influence
Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.;
Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y.
Hu, Y.; Koren, Y.; and Volinsky, C. 2008. Collaborative fil-
tering for implicit feedback datasets. In NeurIPS,
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
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 Krish-
naswamy, S. 2015. Rank-GeoFM: A ranking based geo-
graphical factorization method for point of interest recom-
mendation. In SIGIR, 433–442.
Lian, D.; Zhao, C.; Xie, X.; Sun, G.; Chen, E.; and Rui, Y.
2014. GeoMF: Joint geographical modeling and matrix fac-
torization 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 recommenda-
tion in location-based social networks. PVLDB 10(10):
1010–1021.
Liu, Y.; Wei, W.; Sun, A.; and Miao, C. 2014. Exploiting
geo-social location characteristics for location recom-
mendation. 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 autoen-
Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and
Dean, J. 2013. Distributed representations of words and phrases
Pei, W.; Yang, J.; Sun, Z.; Zhang, J.; Bozzon, A.; and Tax,
D. M. 2017. Interacting attention-gated recurrent networks
Ren, M.; Zeng, W.; Yang, B.; and Urtasun, R. 2018. Learn-
ing to reweight examples for robust deep learning. In ICM.
Shu, J.; Xie, Q.; Yi, L.; Zhao, Q.; Zhou, S.; Xu, Z.; and
diem, seize the samples uncertain “at the moment” for adap-
tive batch selection. In CIKM.
Song, H.; Kim, M.; and Lee, J.-G. 2019. SELFIE: Refur-
bishing unclean samples for robust deep learning. In ICM,
5907–5915.
Boundary: Accelerating DNN training via adaptive boundary
Tarvainen, A.; and Valpola, H. 2017. Mean teachers are
better role models: Weight-averaged consistency targets im-
prove semi-supervised deep learning results. In NeurIPS,
1195–1204.
Wang, H.; Terrovitis, M.; and Mamoulis, N. 2013. Loca-
tion recommendation in location-based social networks using
user check-in data. In SIGSPATIAL, 374–383.
Wu, Y.; DuBois, C.; Zheng, A. X.; and Estor, M. 2016. Col-
laborative 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 learn-
ing: 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 classi-
fication. In NAACL, 1480–1489.
Ye, M.; Yin, P.; Lee, W.-C.; and Lee, D.-L. 2011. Exploit-
ing geographical influence for collaborative point-of-interest
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 esti-
Zhang, J.-D.; Chow, C.-Y.; and Li, Y. 2014. LORE: Exploit-
ing 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
Zhou, F.; Yin, R.; Zhang, K.; Trajcevski, G.; Zhong, T.; and
In TheWebConf, 3462–34618.