The Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI-22)

Obtaining Calibrated Probabilities with Personalized Ranking Models

Wonbin Kweon, SeongKu Kang, Hwanjo Yu*

Pohang University of Science and Technology, South Korea
{kwb4453, seongku, hwanjoyu}@postech.ac.kr

Abstract

For personalized ranking models, the well-calibrated probability of an item being preferred by a user has great practical value. While existing work shows promising results in image classification, probability calibration has not been much explored for personalized ranking. In this paper, we aim to estimate the calibrated probability of how likely a user will prefer an item. We investigate various parametric distributions and propose two parametric calibration methods, namely Gaussian calibration and Gamma calibration. Each proposed method can be seen as a post-processing function that maps the ranking scores of pre-trained models to well-calibrated preference probabilities, without affecting the recommendation performance. We also design the unbiased empirical risk minimization framework that guides the calibration methods to the learning of true preference probability from the biased user-item interaction dataset. Extensive evaluations with various personalized ranking models on real-world datasets show that both the proposed calibration methods and the unbiased empirical risk minimization significantly improve the calibration performance.

Introduction

Personalized ranking models aim to learn the ranking scores of items, so as to produce a ranked list of them for the recommendation (Rendle et al. 2009). However, their prediction results provide an incomplete estimation of the user's potential preference for each item; the semantic of the same ranking position differs for each user. One user might like his third item with the probability of 30%, whereas the other user likes her third item with 90%. Accurately estimating the *probability* of an item being preferred by a user has great practical value (Menon et al. 2012). The preference probability can help the user choose the items with high potential preference and the system can raise user satisfaction by pruning the ranked list by filtering out items with low confidence (Arampatzis, Kamps, and Robertson 2009). To ensure reliability, the predicted probabilities need to be *calibrated* so that they can accurately indicate their ground truth correctness likelihood. In this paper, our goal is to obtain the well-calibrated probability of an item matching a user's preference based on the ranking score of the pre-trained model, without affecting the ranking performance.

While recent methods (Guo et al. 2017; Kull et al. 2019; Rahimi et al. 2020) have successfully achieved model calibration for image classification, it has remained a long-standing problem for personalized ranking. A pioneering work (Menon et al. 2012) firstly proposed to predict calibrated probabilities from the scores of pre-trained ranking models by using isotonic regression (Barlow and Brunk 1972), which is a simple non-parametric method that fits a monotonically increasing function. Although it has shown some effectiveness, there is no subsequent study about *parametric* calibration methods in the field of personalized ranking despite their richer expressiveness than non-parametric methods.

In this paper, we investigate various parametric distributions, and from which we propose two calibration methods that can best model the score distributions of the ranking models. First, we define three desiderata that a calibration function for ranking models should meet, and show that existing calibration methods have the insufficient capability to model the diverse populations of the ranking score. We then propose two parametric methods, namely Gaussian calibration and Gamma calibration, that satisfy all the desiderata. We demonstrate that the proposed methods have a larger expressive power in terms of the parametric family and also effectively handles the imbalanced nature of ranking score populations compared to the existing methods (Platt et al. 1999; Guo et al. 2017). Our methods are post-processing functions with *three* learnable parameters that map the ranking scores of pre-trained models to calibrated posterior probabilities.

To optimize the parameters of the calibration functions, we can use the log-loss on the held-out validation sets (Guo et al. 2017). The challenge here is that the user-item interaction datasets are implicit and missing-not-at-random (Schnabel et al. 2016; Saito 2019). For each user-item pair, the label is 1 if the interaction is observed, 0 otherwise. An unobserved interaction, however, does not necessarily mean a negative preference, but the item might have not been exposed to the user yet. Therefore, if we fit the calibration function with the log-loss computed naively on the implicit datasets, the mapped probabilities may indicate biased likelihoods of users' preference on items. To tackle this prob-

*Corresponding author

lem, we design an unbiased empirical risk minimization framework by adopting Inverse Propensity Scoring (Robins, Rotnitzky, and Zhao 1994). We first decompose the interaction variable into two variables for observation and preference, and adopt an inverse propensity-scored log-loss that guides the calibration functions toward the true preference probability.

Extensive evaluations with various personalized ranking models on real-world datasets show that the proposed calibration methods produce more accurate probabilities than existing methods in terms of calibration measures like ECE, MCE, and NLL. Our unbiased empirical risk minimization framework successfully estimates the ideal empirical risk, leading to performance gain over the naive log-loss. Furthermore, reliability diagrams show that Gaussian calibration and Gamma calibration predict well-calibrated probabilities across all probability range. Lastly, we provide an in-depth analysis that supports the superiority of the proposed methods over the existing methods.

Preliminary & Related Work

Personalized Ranking

Let $\mathcal{U}$ and $\mathcal{I}$ denote the user space and the item space, respectively. For each user-item pair (u, i) of $u \in \mathcal{U}$ and $i \in \mathcal{I}$, a label $Y_{u,i}$ is given as 1 if their interaction is observed and 0 otherwise. It is worth noting that unobserved interaction ($Y_{u,i} = 0$) may indicate the negative preference or the unawareness, or both. A personalized ranking model $f_\theta : \mathcal{U} \times \mathcal{I} \to \mathbb{R}$ learns the ranking scores of user-item pairs to produce a ranked list of items for each user. f_θ is mostly trained with pairwise loss that makes the model put a higher score on the observed pair than the unobserved pair:

$$\mathcal{L}_{pair} = \sum_{u \in \mathcal{U}, i, j \in \mathcal{I}} \ell(f_\theta(u, i), f_\theta(u, j)) Y_{u,i}(1 - Y_{u,j}), \quad (1)$$

where $\ell(\cdot, \cdot)$ is some convex loss function such as BPR loss (Rendle et al. 2009) or Margin Ranking loss (Weimer et al. 2007). Note that the ranking score $f_\theta(u, i) \in \mathbb{R}$ is not bounded in $[0, 1]$ and therefore cannot be used as a probability.

Calibrated Probability

To estimate $P(Y_{u,i} = 1 | f_\theta(u, i))$, which is the probability of item i being interacted with user u given the pre-trained ranking score, we need a post-processing calibration function $g_\phi(s)$ that maps the ranking score $s = f_\theta(u, i)$ to the calibrated probability p. Here, the calibration function for the personalized ranking has to meet the following **desiderata**: (1) the function $g_\phi : \mathbb{R} \to [0, 1]$ needs to take an input from the unbounded range of the ranking score to output a probability; (2) the function should be *monotonically increasing* so that the item with a higher ranking score gets a higher preference probability; (3) the function needs enough expressiveness to represent diverse score distributions.

We say the probability p is well-calibrated if it indicates the ground-truth correctness likelihood (Kull, Silva Filho, and Flach 2017):

$$\mathbb{E}[Y | g_\phi(s) = p] = p, \quad \forall p \in [0, 1]. \quad (2)$$

For example, if we have 100 predictions with $p = 0.3$, we expect 30 of them to indeed have $Y = 1$ when the probabilities are calibrated. Using this definition, we can measure the miscalibration of a model with Expected Calibration Error (ECE) (Naeini, Cooper, and Hauskrecht 2015):

$$\text{ECE}(g_\phi) = \mathbb{E}\big[|\mathbb{E}[Y | g_\phi(s) = p] - p|\big]. \quad (3)$$

However, since we only have finite samples, we cannot directly compute ECE with Eq.3. Instead, we partition the [0,1] range of p into M equi-spaced bins and aggregate the value of each bin:

$$\text{ECE}_M(g_\phi) = \sum_{m=1}^{M} \frac{|B_m|}{N} \left| \frac{\sum_{k \in B_m} Y_k}{|B_m|} - \frac{\sum_{k \in B_m} p_k}{|B_m|} \right|, \quad (4)$$

where B_m is m-th bin and N is the number of samples. The first term in the absolute value symbols denotes the ground-truth proportion of positive samples (accuracy) in B_m and the second term denotes the average calibrated probability (confidence) of B_m. Similarly, Maximum Calibration Error (MCE) is defined as follows:

$$\text{MCE}_M(g_\phi) = \max_{m \in \{1, .., M\}} \left| \frac{\sum_{k \in B_m} Y_k}{|B_m|} - \frac{\sum_{k \in B_m} p_k}{|B_m|} \right|. \quad (5)$$

MCE measures the worst-case discrepancy between the accuracy and the confidence. Besides the above calibration measures, Negative Log-Likelihood (NLL) also can be used as a calibration measure (Guo et al. 2017).

Calibration Method

Existing methods for model calibration are categorized into two groups: non-parametric and parametric methods. Non-parametric methods mostly adopt the binning scheme introduced by the histogram binning (Zadrozny and Elkan 2001). The histogram binning divides the uncalibrated model outputs into B equi-spaced bins and samples in each bin take the proportion of positive samples in the bin as the calibrated probability. Subsequently, isotonic regression (Menon et al. 2012) adjusts the number of bins and their width, Bayesian binning into quantiles (BBQ) (Naeini, Cooper, and Hauskrecht 2015) takes an average of different binning models for the better generalization. In the perspective of our desiderata, however, none of them meets all three conditions (please refer to Appendix A).

The parametric methods try to fit calibration functions that map the output scores to the calibrated probabilities. Temperature scaling (Guo et al. 2017), a well-known technique for calibrating deep neural networks, is a simplified version of Platt scaling (Platt et al. 1999) that adopts Gaussian distributions with the same variance for the positive and the negative classes. Beta calibration (Kull, Silva Filho, and Flach 2017) utilizes Beta distribution for the binary classification and Dirichlet calibration (Kull et al. 2019) generalizes it for the multi-class classification. While recent work (Rahimi et al. 2020; Mukhoti et al. 2020) is focusing on parametric methods and shows promising results for image classification, they cannot be directly adopted for the personalized ranking. Also, the above parametric methods do not

satisfy all the desiderata (please refer to Appendix A). In this paper, we propose two parametric calibration methods that satisfy all the desiderata for the personalized ranking models.

Proposed Calibration Method

Revisiting Platt Scaling

Platt scaling (Platt et al. 1999) is widely used parametric calibration method, which is a generalized form of the temperature scaling (Guo et al. 2017):

$$g_\phi^{\text{Platt}}(s) = \sigma(bs + c), \tag{6}$$

where $\phi = \{b, c\}$ are learnable parameters and $\sigma(x) = 1/(1 + \exp(-x))$ is the sigmoid function. In this section, we show that Platt scaling can be derived from the assumption that the class-conditional scores follow Gaussian distributions with the same variance.

We first set the class-conditional score distribution for the positive and the negative classes:

$$\begin{aligned} p(s|Y=0) &= (\sqrt{2\pi}\sigma_0)^{-1}\exp[-(s-\mu_0)^2/2\sigma_0^2], \\ p(s|Y=1) &= (\sqrt{2\pi}\sigma_1)^{-1}\exp[-(s-\mu_1)^2/2\sigma_1^2], \end{aligned} \tag{7}$$

where $\mu_0, \mu_1 \in \mathbb{R}$, $\sigma_0^2, \sigma_1^2 \in \mathbb{R}^+$ are the mean and the variance of each Gaussian distribution. Then, the posterior is computed as follows:

$$\begin{aligned} P(Y=1|s) &= \frac{\pi_1 p(s|Y=1)}{\pi_1 p(s|Y=1) + \pi_0 p(s|Y=0)} \\ &= \frac{1}{1 + \pi_0 p(s|Y=0)/\pi_1 p(s|Y=1)} \\ &= \frac{1}{1 + \exp\big[(\frac{1}{2\sigma_1^2} - \frac{1}{2\sigma_0^2})s^2 + (\frac{\mu_0}{\sigma_0^2} - \frac{\mu_1}{\sigma_1^2})s - c\big]} \\ &= \sigma(as^2 + bs + c), \end{aligned} \tag{8}$$

where π_0 and π_1 are the prior probability for each class, $a = (2\sigma_0^2)^{-1} - (2\sigma_1^2)^{-1}$, $b = \mu_1/\sigma_1^2 - \mu_0/\sigma_0^2$, and $c = \mu_1^2/(2\sigma_1^2) - \mu_0^2/(2\sigma_0^2) + \log(\pi_0\sigma_1) - \log(\pi_1\sigma_0) \in \mathbb{R}$. We can see that Platt scaling is a special case of Eq.8 with the assumption $a = 0$ (i.e., the same variance for both class-conditional score distributions).

Gaussian Calibration

For personalized ranking, however, the usage of the same variance for both class-conditional score distributions is not desirable, because a severe imbalance between the two classes exists in user-item interaction datasets. Since users have distinct preferences for item categories, preferred items take only a small portion ($\sim$10% in real-world datasets) of the entire itemset. Therefore, the score distribution of diverse unpreferred items and that of distinct preferred items are likely to have disparate variances.

To tackle this problem, we let the variance of each class-conditional score distribution be optimized with datasets, without any naive assumption of the same variance for both classes:

$$g_\phi^{\text{Gaussian}}(s) = \sigma(as^2 + bs + c), \tag{9}$$

where $\phi = \{a, b, c\}$ are learnable parameters and can be any real numbers. Since $a = (2\sigma_0^2)^{-1} - (2\sigma_1^2)^{-1}$ can capture the different deviations of two classes during the training, we can handle the distinct distribution of each class.

Gamma Calibration

Gamma distribution is also widely adopted to model the score distribution of ranking models (Baumgarten 1999). Unlike Gaussian distribution that is symmetric about its mean, Gamma distribution can capture the skewed population of ranking scores that might exist in the datasets. In this section, we set the class-conditional score distribution to Gamma distribution:

$$\begin{aligned} p(s|Y=0) &= \Gamma(\alpha_0)^{-1}\beta_0^{\alpha_0}s^{\alpha_0-1}\exp(-\beta_0 s), \\ p(s|Y=1) &= \Gamma(\alpha_1)^{-1}\beta_1^{\alpha_1}s^{\alpha_1-1}\exp(-\beta_1 s), \end{aligned} \tag{10}$$

where $\Gamma(\cdot)$ is the Gamma function, $\alpha_0, \alpha_1, \beta_0, \beta_1 \in \mathbb{R}^+$ are the shape and the rate parameters of each Gamma distribution. Then, the posterior is computed as follows:

$$\begin{aligned} P(Y=1|s) &= \frac{1}{1 + \pi_0 p(s|Y=0)/\pi_1 p(s|Y=1)} \\ &= \frac{1}{1 + \frac{\pi_0\beta_0^{\alpha_0}\Gamma(\alpha_1)}{\pi_1\beta_1^{\alpha_1}\Gamma(\alpha_0)}s^{\alpha_0-\alpha_1}\exp[(\beta_1-\beta_0)s]} \\ &= \frac{1}{1 + \exp\big[(\alpha_0-\alpha_1)\log s + (\beta_1-\beta_0)s - c\big]} \\ &= \sigma(a\log s + bs + c), \end{aligned} \tag{11}$$

where $a = \alpha_1 - \alpha_0$, $b = \beta_0 - \beta_1$, and $c = \log(\pi_1\beta_1^{\alpha_1}\Gamma(\alpha_0)/\pi_0\beta_0^{\alpha_0}\Gamma(\alpha_1)) \in \mathbb{R}$. Therefore, Gamma calibration can be formalized as follows:

$$g_\phi^{\text{Gamma}}(s) = \sigma(a\log s + bs + c), \tag{12}$$

where $\phi = \{a, b, c\}$ are learnable parameters. Since Gamma distribution is defined only for the positive real number, we need to shift the score to make all the inputs positive: $s \leftarrow s - s_{\text{min}}$, where s_{min} is the minimum ranking score.

Other Distributions

Besides adopting Gaussian distribution or Gamma distribution for both classes, there have been proposed other parametric distributions for modeling the ranking scores. Swets adopts two Exponential distributions (Swets 1969), Manmatha proposes Gaussian distribution for the positive class and Exponential distribution for the negative class (Manmatha, Rath, and Feng 2001), and Kanoulas proposes Gaussian distribution for the positive class and Gamma distribution for the negative class (Kanoulas et al. 2010). We also investigated these distributions, however, they either have the same form as the proposed calibration function or their posterior cannot satisfy our desiderata. Please refer to Appendix B for more information.

Monotonicity for Proposed Desiderata

The proposed calibration methods naturally satisfy the first and the third of our desiderata: (1) the proposed methods

take the unbounded ranking scores and produce calibrated probabilities; (2) the proposed methods have richer expressiveness than Platt scaling or temperature scaling, since they have a larger capacity in terms of the parametric family. The last condition that our calibration methods need to meet is that they should be monotonically increasing for maintaining the ranking order. To this end, we need linear constraints on the parameters of each method: $2as+b>0$ for Gaussian calibration and $a/s+b>0$ for Gamma calibration (derivation of these constraints can be found in Appendix C). Since these constraints are linear and we have only three learnable parameters, the optimization of constrained logistic regression is easily done in at most a few minutes with the existing module of Scipy (Pedregosa et al. 2011).

Unbiased Parameter Fitting

Naive Log-loss

After we formalize Gaussian Calibration and Gamma Calibration, we need to optimize their learnable parameters ϕ. A well-known way to fit them is to use log-loss on the held-out validation set, which can be the same set used for the hyperparameter tuning (Guo et al. 2017; Kull, Silva Filho, and Flach 2017). Since we only observe the interaction indicator $Y_{u,i}$, the naive negative log-likelihood is computed for a user-item pair as follows:

$$\mathcal{L}_{\text{naive}} = -Y_{u,i}\log(g_\phi(s_{u,i})) - (1-Y_{u,i})\log(1-g_\phi(s_{u,i})). \tag{13}$$

where $s_{u,i} = f_\theta(u,i)$ is the ranking score for the user-item pair. Note that during the fitting of the calibration function $g_\phi(s)$, the parameters of the pre-trained ranking model $f_\theta(u,i)$ are fixed.

Ideal Log-loss for Preference Estimation

The observed interaction label $Y_{u,i}$, however, indicates the presence of user-item interaction, not the user's preference on the item. Therefore, $Y_{u,i}=0$ does not necessarily mean the user's negative preference, but it can be that the user is not aware of the item. If we fit the calibration function with $\mathcal{L}_{\text{naive}}$, mapped probabilities could be biased towards the negative preference by treating the unobserved positive pair as the negative pair. To handle this implicit interaction process, we borrow the idea of decomposing the interaction variable $Y_{u,i}$ into two independent binary variables (Schnabel et al. 2016):

$$\begin{aligned} Y_{u,i} &= O_{u,i}\cdot R_{u,i}, \\ P(Y_{u,i}=1) &= P(O_{u,i}=1)\cdot P(R_{u,i}=1) \\ &= \omega_{u,i}\cdot\rho_{u,i}, \end{aligned} \tag{14}$$

where $O_{u,i}$ is a binary random variable representing whether the item i is observed by user u, and $R_{u,i}$ is a binary random variable representing whether the item i is preferred by user u. The user-item pair interacts ($Y_{u,i}=1$) when the item is observed ($O_{u,i}=1$) and preferred ($R_{u,i}=1$) by the user.

The goal of this paper is to estimate the probability of an item being *preferred* by a user, not the probability of an item being *interacted* by a user. Therefore, we need to train $g_\phi(s)$ for predicting $P(R=1|s)$ instead of $P(Y=1|s)$[1]. To this end, we need a new ideal loss function that can guide the optimization towards the true preference probability:

$$\mathcal{L}_{\text{ideal}} = -R_{u,i}\log(g_\phi(s_{u,i})) - (1-R_{u,i})\log(1-g_\phi(s_{u,i})). \tag{15}$$

The ideal loss function enables the calibration function to learn the unbiased preference probability. However, since we cannot observe the variable $R_{u,i}$ from the training set, the ideal log-loss cannot be computed directly.

Unbiased Empirical Risk Minimization

In this section, we design an unbiased empirical risk minimization (UERM) framework to obtain the ideal empirical risk minimizer. We deploy the Inverse Propensity Scoring (IPS) estimator (Robins, Rotnitzky, and Zhao 1994), which is a technique for estimating the counterfactual outcome of a subject under a particular treatment. The IPS estimator is widely adopted for the unbiased rating prediction (Schnabel et al. 2016; Wang et al. 2019) and the unbiased pairwise ranking (Joachims, Swaminathan, and Schnabel 2017; Saito 2019). For a user-item pair, the inverse propensity-scored log-loss for the unbiased empirical risk minimization is defined as follows:

$$\mathcal{L}_{\text{UERM}} = -\frac{Y_{u,i}}{\omega_{u,i}}\log(g_\phi(s_{u,i})) - (1-\frac{Y_{u,i}}{\omega_{u,i}})\log(1-g_\phi(s_{u,i})), \tag{16}$$

where $\omega_{u,i}=P(O_{u,i}=1)$ is called *propensity score*.

Proposition 1. *$\hat{\mathcal{R}}_{\text{UERM}}(g_\phi|\omega)$, which is the empirical risk of $\mathcal{L}_{\text{UERM}}$ on validation set with true propensity score ω, is equal to $\hat{\mathcal{R}}_{\text{ideal}}(g_\phi)$, which is the ideal empirical risk.*

The proof can be found in Appendix D. This proposition shows that we can get the unbiased empirical risk minimizer by $\phi^{\text{UERM}} = \text{argmin}_\phi\{\hat{\mathcal{R}}_{\text{UERM}}(g_\phi|\omega)\}$ when only $Y_{u,i}$ is observed.

The remaining challenge is to estimate the propensity score $\omega_{u,i}$ from the dataset. There have been proposed several techniques for estimating the propensity score such as Naive Bayes (Schnabel et al. 2016) or logistic regression (Rosenbaum 2002). However, the Naive Bayes needs unbiased held-out data for the missing-at-random condition and the logistic regression needs additional information like user demographics and item categories. In this paper, we adopt a simple way that utilizes the popularity of items as done in (Saito 2019): $\hat{\omega}_{u,i} = (\sum_{u\in\mathcal{U}} Y_{u,i}/\max_{i\in\mathcal{I}}\sum_{u\in\mathcal{U}} Y_{u,i})^{0.5}$. While one can concern that this estimate of propensity score may be inaccurate, Schnabel (Schnabel et al. 2016) shows that we merely need to estimate better than the naive uniform assumption. We provide an experimental result that demonstrates our estimate of the propensity score shows comparable performance with Naive Bayes and Logistic Regression that use additional information (Appendix F).

For deeper insights into the variability of the estimated empirical risk, we investigate the bias when the propensity scores are inaccurately estimated.

[1] We can replace $Y_{u,i}$ with $R_{u,i}$ in Eq.2∼12.

Proposition 2. *The bias of* $\hat{\mathcal{R}}_{\text{UERM}}(g_\phi|\hat{\omega})$ *induced by the inaccurately estimated propensity scores* $\hat{\omega}$ *is* $\frac{1}{|\mathcal{D}_{\text{val}}|}\sum_{(u,i)\in\mathcal{D}_{\text{val}}}\rho_{u,i}\left(\frac{\omega_{u,i}}{\hat{\omega}_{u,i}}-1\right)\log\left(\frac{g_\phi(s_{u,i})}{1-g_\phi(s_{u,i})}\right)$.

The proof can be found in Appendix D. Obviously, the bias is zero when the propensity score is correctly estimated. Furthermore, we can see that the magnitude of the bias is affected by the inverse of the estimated propensity score. This finding is consistent with the previous work (Su et al. 2019) that proposes to adopt a propensity clipping technique to reduce the variability of the bias. In this work, we use a simple clipping technique $\hat{\omega}_{u,i} \leftarrow \max\{\hat{\omega}_{u,i}, 0.1\}$ that can prevent the item with extremely low popularity from amplifying the bias (Saito 2019).

Experiment

Experimental Setup

We concisely introduce our experimental settings in this section. For more details, please refer to Appendix E. Our source code is publicly available[2].

Datasets To evaluate the calibration quality of predicted preference probability, we need an unbiased test set where we can directly observe the preference variable $R_{u,i}$ without any bias from the observation process $O_{u,i}$. To the best of our knowledge, there are two real-world datasets that have separate unbiased test sets where the users are asked to rate uniformly sampled items (i.e., $O_{u,i} = 1$ for test sets). Note that in the training set, we only observe the interaction $Y_{u,i}$. **Yahoo!R3**[3] has over 300K *interactions* in the training set and 54K *preferences* in the test set from 15.4K users and 1K songs. **Coat** (Schnabel et al. 2016) has over 7K interactions in the training set and 4.6K preferences in the test set from 290 users and 300 coats. We hold out 10% of the training set as the validation set for the hyperparameter tuning of the base models and the optimization of the calibration methods.

Base models For rigorous evaluation, we apply the calibration methods on several widely-used personalized ranking models with various model architectures and loss functions: Bayesian Personalized Ranking (BPR) (Rendle et al. 2009), Neural Collaborative Filtering (NCF) (He et al. 2017), Collaborative Metric Learning (CML) (Hsieh et al. 2017), Unbiased BPR (UBPR) (Saito 2019), and LightGCN (LGCN) (He et al. 2020). The details for the training of these base models can be found in Appendix E.

Calibration methods compared We evaluate the proposed calibration methods with various calibration methods. For the naive baseline, we adopt the minmax normalizer and the sigmoid function which simply re-scale the scores into [0,1] without calibration. For non-parametric methods, we adopt Histogram binning (Zadrozny and Elkan 2001), Isotonic regression (Menon et al. 2012), and BBQ (Naeini, Cooper, and Hauskrecht 2015). For parametric methods, we adopt Platt scaling (Platt et al. 1999) and Beta calibration (Kull, Silva Filho, and Flach 2017). Note that we do not compare recent work designed for multi-class classification (Kull et al. 2019; Rahimi et al. 2020), since they are either the generalized version of Beta calibration or cannot be directly adopted for the personalized ranking models.

Evaluation metrics We adopt well-known calibration metrics like ECE, MCE with $M = 15$, and NLL as done in recent work (Kull et al. 2019; Rahimi et al. 2020). We also plot the reliability diagram that shows the discrepancy between the accuracy and the average calibrated probability of each probability interval. Note that evaluation metrics are computed on $R_{u,i}$ which is observed only from the test set.

Evaluation process We first train the base personalized ranking model $f_\theta(u,i)$ with $Y_{u,i}$ on the training set. Second, we compute ranking score $s_{u,i} = f_\theta(u,i)$ for user-item pairs in the validation set. Third, we optimize the calibration method $g_\phi(s)$ on the validation set with the computed $s_{u,i}$ and the estimated $\hat{\omega}_{u,i}$, with $f_\theta(u,i)$ fixed. Lastly, we evaluate the calibrated probability $p = g_\phi(s_{u,i})$ with $R_{u,i}$ from the unbiased test set by using the above evaluation metrics.

Comparing Calibration Performance

Table 1 shows ECE of each calibration method applied on the various personalized ranking models (MCE and NLL can be found in Appendix F). ECE indicates how well the calibrated probabilities and ground-truth likelihoods match on the test set across all probability ranges. First, the minmax normalizer and the sigmoid function produce poorly calibrated preference probabilities. It is obvious because the ranking scores do not have any probabilistic meaning and naively re-scaling them cannot reflect the score distribution.

Second, the parametric methods better calibrate the preference probabilities than the non-parametric methods in most cases. This is consistent with recent work (Guo et al. 2017; Kull et al. 2019) for image classification. The non-parametric calibration methods lack rich expressiveness since they rely on the binning scheme, which maps the ranking scores to the probabilities in a discrete manner. On the other hand, the parametric calibration methods fit the continuous functions based on the parametric distributions. Therefore, they have a more granular mapping from the ranking scores to the preference probabilities.

Third, every parametric calibration method benefits from adopting $\mathcal{L}_{\text{UERM}}$ instead of $\mathcal{L}_{\text{naive}}$ for the parameter fitting. The naive log-loss treats all the unobserved pairs as negative pairs and makes the calibration methods produce biased preference probabilities. On the contrary, inverse propensity-scored log-loss handles such problem and enables us to compute the ideal empirical risk indirectly. As a result, ECE decreases by 7.40%$\sim$76.52% for all parametric methods compared to when the naive log-loss is used for the optimization.

Lastly, Gaussian calibration and Gamma calibration with $\mathcal{L}_{\text{UERM}}$ show the best calibration performance across all base models and datasets. Platt scaling can be seen as a special case of the proposed methods with $a = 0$, so it has less expressiveness in terms of the capacity of parametric family.

[2]https://github.com/WonbinKweon/CalibratedRankingModels_AAAI2022

[3]http://research.yahoo.com/Academic_Relations

		Yahoo!R3					Coat				
Type	Methods	BPR	NCF	CML	UBPR	LGCN	BPR	NCF	CML	UBPR	LGCN
uncalibrated	MinMax	0.4929	0.4190	0.3152	0.3004	0.2258	0.1790	0.4624	0.1834	0.1920	0.2350
	Sigmoid	0.3065	0.0729	0.0526	0.2516	0.3024	0.2196	0.1422	0.0647	0.1415	0.0508
non-parametric	Hist	0.0161	0.0133	0.0641	0.0130	0.0194	0.0552	0.0230	0.0161	0.0514	0.0470
	Isotonic	0.0146	0.0130	0.0635	0.0127	0.0154	0.0474	0.0159	0.0160	0.0490	0.0453
	BBQ	0.0136	0.0137	0.0634	0.0140	0.0165	0.0552	0.0178	0.0198	0.0459	0.0494
parametric w/ $\mathcal{L}_{\text{naive}}$	Platt	0.0126	0.0146	0.0515	0.0107	0.0099	0.0441	0.0245	0.0203	0.0423	0.0407
	Beta	0.0127	0.0144	0.0504	0.0105	0.0150	0.0451	0.0258	0.0270	0.0416	0.0407
	Gaussian	0.0129	0.0104	0.0486	0.0105	0.0073	0.0436	0.0264	0.0245	0.0410	0.0404
	Gamma	0.0108	0.0145	0.0512	0.0107	0.0098	0.0424	0.0239	0.0208	0.0405	0.0406
parametric w/ $\mathcal{L}_{\text{UERM}}$	Platt	0.0106	0.0129	0.0303	0.0100	0.0070	0.0411	0.0120	0.0155	0.0354	0.0224
	Beta	0.0109	0.0132	0.0305	0.0094	0.0076	0.0414	0.0075	0.0183	0.0375	0.0266
	Gaussian	0.0106	**0.0096**	**0.0285**	**0.0070**	**0.0061**	0.0393	0.0062	**0.0147**	**0.0323**	**0.0208**
	Gamma	**0.0100**	0.0117	0.0287	0.0085	0.0065	**0.0390**	**0.0061**	0.0148	0.0326	0.0215
	Improv	5.85%	25.35%	5.94%	25.85%	12.86%	5.21%	18.67%	5.41%	8.81%	7.14%

Table 1: Expected Calibration Error of each calibration method applied on five personalized ranking models. Numbers in boldface are the best results and *Improv* denotes the improvement of the best proposed method over the best competitor (Platt or Beta with $\mathcal{L}_{\text{UERM}}$).

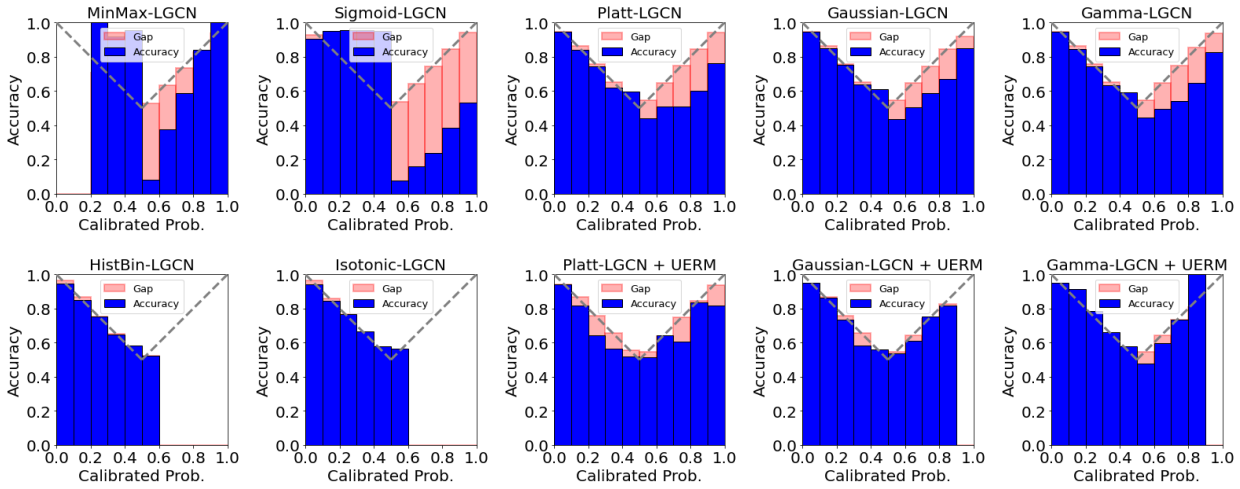

Figure 1: Reliability diagram of each calibration method. Gap denotes the discrepancy between the accuracy and the average calibrated probability for each bin. The grey dashed line is a diagonal function that indicates the ideal reliability line where the blue accuracy bar should meet.

Beta distribution is only defined in [0,1], so it cannot represent the unbounded ranking scores. To adopt Beta calibration, we need to re-scale the ranking score, however, it is not verified for the optimality (Menon et al. 2012). As a result, our calibration methods improve ECE by 5.21%∼25.85% over the best competitor. Also, since our proposed models have a larger capacity of expressiveness, they show larger improvement on Yahoo!R3, which has more samples to fit the parameters than Coat.

Reliability Diagram

Figure 1 shows the reliability diagram (Guo et al. 2017) for each calibration method applied on LGCN for Yahoo!R3. We partition the calibrated probabilities $g_\phi(s)$ into 10 equispaced bins and compute the accuracy and the average calibrated probability for each bin (i.e., the first and the second term in Eq.4, respectively). The accuracy is the same with the ground-truth proportion of positive samples for the positive bins (i.e., probability over 0.5) and the ground-truth proportion of negative samples for the negative bins (i.e., probability under 0.5). Note that the bar does not exist if the bin does not have any prediction in it.

First, the non-parametric calibration methods do not produce the probability over 0.6. It is because they can easily be overfitted to the unbalanced user-item interaction datasets since they do not have any prior distribution. On the other

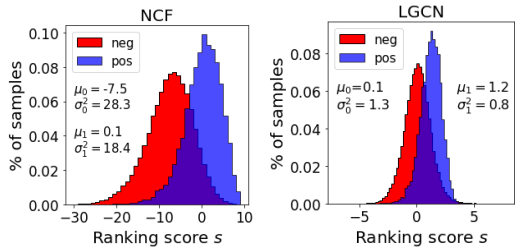

Figure 2: Ranking score distributions of negative and positive pairs.

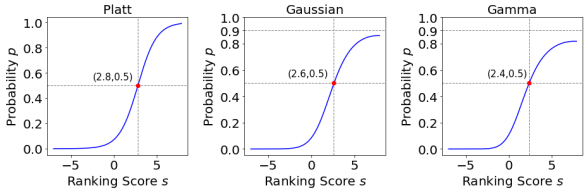

Figure 3: Fitted function of each calibration method.

hand, the parametric calibration methods produce probabilities across all ranges by avoiding such overfitting problem with the prior parametric distributions.

Second, the parametric calibration methods with UERM produce well-calibrated probabilities especially for the positive preference ($p > 0.5$). The naive log-loss makes the calibration methods biased towards the negative preference, by treating all the unobserved pairs as the negative pairs. As a result, the parametric methods with the naive log-loss (upper-right three diagrams of Figure 1) show large gaps in the positive probability range ($p > 0.5$). On the contrary, UERM framework successfully alleviates this problem and produces much smaller gaps for the positive preference (lower-right three diagrams of Figure 1). Lastly, it is quite a natural result that parametric methods with UERM do not produce the probability over 0.9, considering that the users prefer only a few items among a large number of items.

Score Distribution & Fitted Function

Figure 2 shows the distribution of ranking scores trained by NCF and LGCN on Yahoo!R3. We can see that the class-conditional score distributions have different deviations ($\sigma_0 > \sigma_1$) and skewed shapes (left tails are longer than the right tails). This indicates that Platt scaling (or temperature scaling) assuming the same variance for both classes cannot effectively handle these score distributions. Figure 3 shows the fitted calibration function of each parametric method adopted on LGCN and optimized with UERM. Since most of the user-item pairs are negative in the interaction datasets, all three functions are fitted to produce the low probability under 0.1 for a wide bottom range to reflect the dominant negative preferences. Platt scaling is forced to have the symmetric shape due to its parametric family, so it produces the high probability over 0.9 which is symmetrical to that of under 0.1. On the other hand, Gaussian calibration and Gamma calibration, which have a larger expressive power, learn asymmetric shapes tailored to the score distributions having different deviations and skewness. This result shows that they effectively handle the imbalance of user-item interaction datasets and supports the experimental superiority of the proposed methods.

u_{4506}	i_{126}	i_{319}	i_{55}	i_{811}	i_{580}
s	7.67	7.51	6.52	6.47	6.36
$g_\phi(s)$	0.83	0.82	0.81	0.81	0.81
u_{8637}	i_{831}	i_{154}	i_{398}	i_{579}	i_{304}
s	0.68	0.58	0.54	0.54	0.48
$g_\phi(s)$	0.13	0.13	0.12	0.12	0.12
u_{2940}	i_{126}	i_{169}	i_{319}	i_{303}	i_{98}
s	6.70	5.25	4.96	4.26	3.88
$g_\phi(s)$	0.82	0.76	0.74	0.68	0.63

Figure 4: Case study. Top-5 items for each user with ranking score s and calibrated probability $g_\phi(s)$.

Case Study

Figure 4 shows the case study on Yahoo!R3 with Gaussian calibration adopted on LGCN. The personalized ranking model first learns the ranking scores and produces a top-5 ranking list for each user. Then, Gaussian calibration transforms the ranking scores to the well-calibrated preference probabilities. For the first user u_{4506}, the method produces high preference probabilities for all top-5 items. In this case, we can recommend them to him with confidence. On the other hand, for the second user u_{8637}, all top-5 items have low preference probabilities, and the last user u_{2940} has a wide range of preference probabilities. For these users, merely recommending all the top-ranked items without consideration of potential preference degrade their satisfaction. It is also known that the unsatisfactory recommendations even make the users leave the platform (McNee, Riedl, and Konstan 2006). Therefore, instead of recommending items with low confidence, the system should take other strategies, such as requesting additional user feedback (Kweon et al. 2020).

Conclusion

In this paper, we aim to obtain calibrated probabilities with personalized ranking models. We investigate various parametric distributions and propose two parametric calibration methods, namely Gaussian calibration and Gamma calibration. We also design the unbiased empirical risk minimization framework that helps the calibration methods to be optimized towards true preference probability with the biased user-item interaction dataset. Our extensive evaluation demonstrates that the proposed methods and framework significantly improve calibration metrics and have a richer expressiveness than existing methods. Lastly, our case study shows that the calibrated probability provides an objective criterion for the reliability of recommendations, allowing the system to take various strategies to increase user satisfaction.

Acknowledgments

This work was supported by the NRF grant funded by the MSIT (South Korea, No.2020R1A2B5B03097210), the IITP grant funded by the MSIT (South Korea, No.2018-0-00584, 2019-0-01906), and the Technology Innovation Program funded by the MOTIE (South Korea, No.20014926).

References

Arampatzis, A.; Kamps, J.; and Robertson, S. 2009. Where to stop reading a ranked list? Threshold optimization using truncated score distributions. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 524–531.

Barlow, R. E.; and Brunk, H. D. 1972. The isotonic regression problem and its dual. *Journal of the American Statistical Association*, 67(337): 140–147.

Baumgarten, C. 1999. A probabilistic solution to the selection and fusion problem in distributed information retrieval. In *Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval*, 246–253.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*, 1321–1330.

He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; and Wang, M. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, 639–648.

He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, 173–182.

Hsieh, C.-K.; Yang, L.; Cui, Y.; Lin, T.-Y.; Belongie, S.; and Estrin, D. 2017. Collaborative metric learning. In *Proceedings of the 26th international conference on world wide web*, 193–201.

Joachims, T.; Swaminathan, A.; and Schnabel, T. 2017. Unbiased learning-to-rank with biased feedback. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, 781–789.

Kanoulas, E.; Dai, K.; Pavlu, V.; and Aslam, J. A. 2010. Score distribution models: assumptions, intuition, and robustness to score manipulation. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, 242–249.

Kull, M.; Perello Nieto, M.; Kängsepp, M.; Silva Filho, T.; Song, H.; and Flach, P. 2019. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with Dirichlet calibration. In *Advances in Neural Information Processing Systems*, 12316–12326.

Kull, M.; Silva Filho, T.; and Flach, P. 2017. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In *International Conference on Artificial Intelligence and Statistics*, 623–631.

Kweon, W.; Kang, S.; Hwang, J.; and Yu, H. 2020. Deep Rating Elicitation for New Users in Collaborative Filtering. In *Proceedings of The Web Conference 2020*, 2810–2816.

Manmatha, R.; Rath, T.; and Feng, F. 2001. Modeling score distributions for combining the outputs of search engines. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, 267–275.

McNee, S. M.; Riedl, J.; and Konstan, J. A. 2006. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI'06 extended abstracts on Human factors in computing systems*, 1097–1101.

Menon, A. K.; Jiang, X. J.; Vembu, S.; Elkan, C.; and Ohno-Machado, L. 2012. Predicting accurate probabilities with a ranking loss. In *International Conference on Machine Learning*, 703–710.

Mukhoti, J.; Kulharia, V.; Sanyal, A.; Golodetz, S.; Torr, P.; and Dokania, P. 2020. Calibrating Deep Neural Networks using Focal Loss. In *Advances in Neural Information Processing Systems*.

Naeini, M. P.; Cooper, G.; and Hauskrecht, M. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830.

Platt, J.; Smola, A.; Bartlett, P.; Scholkopf, B.; and Schuurmans, D. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3): 61–74.

Rahimi, A.; Shaban, A.; Cheng, C.-A.; Hartley, R.; and Boots, B. 2020. Intra order-preserving functions for calibration of multi-class neural networks. In *Advances in Neural Information Processing Systems*, 13456–13467.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 452–461.

Robins, J. M.; Rotnitzky, A.; and Zhao, L. P. 1994. Estimation of regression coefficients when some regressors are not always observed. *Journal of the American statistical Association*, 89(427): 846–866.

Rosenbaum, P. R. 2002. Overt bias in observational studies. In *Observational studies*, 71–104. Springer.

Saito, Y. 2019. Unbiased Pairwise Learning from Implicit Feedback. In *NeurIPS 2019 Workshop on Causal Machine Learning*.

Schnabel, T.; Swaminathan, A.; Singh, A.; Chandak, N.; and Joachims, T. 2016. Recommendations as treatments: Debiasing learning and evaluation. In *International conference on machine learning*, 1670–1679.

Su, Y.; Wang, L.; Santacatterina, M.; and Joachims, T. 2019. Cab: Continuous adaptive blending for policy evaluation and

learning. In *International Conference on Machine Learning*, 6005–6014.

Swets, J. A. 1969. Effectiveness of information retrieval methods. *American Documentation*, 20(1): 72–89.

Wang, X.; Zhang, R.; Sun, Y.; and Qi, J. 2019. Doubly robust joint learning for recommendation on data missing not at random. In *International Conference on Machine Learning*, 6638–6647.

Weimer, M.; Karatzoglou, A.; Le, Q.; and Smola, A. 2007. Cofirank-maximum margin matrix factorization for collaborative ranking. In *Advances in Neural Information Processing Systems*, 222–230.

Zadrozny, B.; and Elkan, C. 2001. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *International Conference on Machine Learning*, 609–616.