LCD: Adaptive Label Correction for Denoising Music Recommendation

Quanyu Dai*
Huawei Noah's Ark Lab
daiquanyu@huawei.com

Yalei Lv*
Tsinghua University
lyl20@mails.tsinghua.edu.cn

Jieming Zhu
Huawei Noah's Ark Lab
jamie.zhu@huawei.com

Junjie Ye
Huawei Noah's Ark Lab
yejunjie4@huawei.com

Zhenhua Dong
Huawei Noah's Ark Lab
dongzhenhua@huawei.com

Rui Zhang
www.ruizhang.info
rayteam@yeah.net

Shu-Tao Xia
Tsinghua University
xiast@sz.tsinghua.edu.cn

Ruiming Tang
Huawei Noah's Ark Lab
tangruiming@huawei.com

ABSTRACT

Music recommendation is usually modeled as a Click-Through Rate (CTR) prediction problem, which estimates the probability of a user listening a recommended song. CTR prediction can be formulated as a binary classification problem where the played songs are labeled as positive samples and the skipped songs are labeled as negative samples. However, such naively defined labels are noisy and biased in practice, causing inaccurate model predictions. In this work, we first identify serious label noise issues in an industrial music App, and then propose an adaptive Label Correction method for Denoising (LCD) music recommendation by ensembling the noisy labels and the model outputs to encourage a consensus prediction. Extensive offline experiments are conducted to evaluate the effectiveness of LCD on both industrial and public datasets. Furthermore, in a one-week online AB test, LCD also significantly increases both the music play count and time per user by 1% to 5%.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

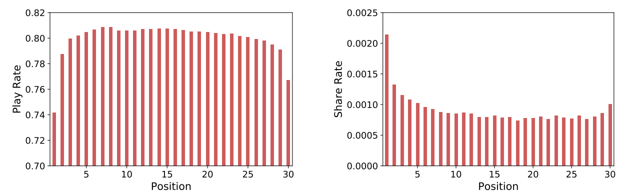
Music Recommendation, CTR Prediction, Label Noise, Denoising

ACM Reference Format:
Quanyu Dai*, Yalei Lv*, Jieming Zhu, Junjie Ye, Zhenhua Dong, Rui Zhang, Shu-Tao Xia, and Ruiming Tang. 2022. LCD: Adaptive Label Correction for Denoising Music Recommendation. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3511808.3557625

* Both authors contributed equally to this work.

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557625



(a) Play rate (b) Share rate

Figure 1: Play rates and share rates at different positions of the recommendation list.

1 INTRODUCTION

With the rapid development of digital music, huge amount of music data makes music recommendation more and more important [19, 29]. Usually, music recommendation can be performed via a Click-Through Rate (CTR) prediction task, which is essentially a binary classification problem that predicts the probability of a user listening a song [6, 21]. To train a CTR prediction model, each sample of the training data should be assigned a binary label indicating a user's interest on an item for supervision. Usually, such labels are naively defined according to user-item interactions in the system, which can be easily affected by noisy feedback, thus resulting in serious label noise issues which can damage model training. Here, we use an example from an industrial music app to illustrate the serious label noise problem in music recommendation.

In the music app, a list of 30 songs is recommended to users everyday based on their preferences. Imagine that a user opens the App in the morning and chooses to listen to the recommended list. At first, her attention is totally on the playing songs. If she likes the song, she will probably give some explicit feedback, such as clicking the *Like* or *Share* button. If she does not like the song, she may switch to the next one immediately. As time goes by, her attention may gradually shift to her work, and thus the playing music just becomes the background sound. Thus, the recommended songs are likely to be fully played one by one without any explicit feedback, neither positive feedback like sharing nor negative feedback like dislike or switching. Therefore, as shown in Figure 1, we observe that the play rate at the first position is the lowest while the share rate is the highest, indicating that users are more likely to pay attention to the music at the beginning and take active actions such as switching to another song or sharing the song, whereas users are less likely to continuously pay attention to the music after the first song. Furthermore, the play rate increases and stabilizes gradually while the share rate decreases and stabilizes, which indicates that users' attention on the playing songs drops gradually.

These unique characteristics make the false-positive issue in music recommendation much more serious than other applications, such as short video or news recommendation which requires users' consistent attention. Moreover, there also exists false-negative issue. Usually, the songs exposed to users but not played are labeled as negative samples, while users may actually like the recommended songs but do not listen to them in many scenarios, e.g., they do not notice the exposed songs or they just shut down the music app after listening the first few songs. Therefore, serious label noise issues exist in music recommendation.

In this paper, we aim to design label correction method for simultaneously tackling the positive and negative label noise issues in music recommendation. There are two challenging problems. Firstly, how to obtain useful information for label correction is nontrivial, since little context information of user playing music can be leveraged. Secondly, how to differentiate the correct labels from the corrupted ones and perform an adaptive and dynamic label correction in an instance-wise manner is difficult.

To address these challenges, we propose an adaptive Label Correction method for Denoising (LCD) data by dynamically updating the target labels according to the current state of the model during training to introduce a consensus prediction. Firstly, previous studies [13, 25] point out that model predictions could magnify useful underlying information in data, and thus can be leveraged to mitigate the damage of the noisy labels since the noisy labels may end up being very inconsistent with model predictions. Therefore, we adaptively adjust the target label based on a convex combination of sample labels and model predictions. Secondly, we design a dynamic weighting scheme based on model loss to perform more effective label correction, since the model loss can help effectively differentiate between correct labels and corrupted ones as discovered by existing work [1, 4, 15]. Our LCD can calibrate the training process by model predictions to reduce the damage of label noise, and improve model performance.

Extensive experiments are conducted on large-scale industrial and public datasets with state-of-the-art (SOTA) CTR models to evaluate our proposed LCD. Furthermore, LCD also significantly improves both the music play count and time per user by 1%-5% in a one-week online AB test of an industrial system as shown in Figure 4. We would like to highlight that label denoising is an extremely important problem in music recommendation, and LCD as a simple, effective and model-agnostic method can easily improve industrial models and serve as a good baseline for future research.

2 METHOD

2.1 Task Formulation

Let $\mathcal{X} \in \mathbb{R}^d$ be the feature space, $\mathcal{Y} = \{0, 1\}$ be the ground-truth label space, and $s = (x, y)$ be samples obtained from a joint distribution over $\mathcal{X} \times \mathcal{Y}$. $y = 0$ indicates that s is unclicked and labeled as a negative sample; $y = 1$ indicates that s is clicked and labeled as a positive sample. Given a noisy training dataset $\tilde{D} = \{(x_i, \tilde{y}_i)\}_{i=1}^N$, where $\tilde{y}$ is a *noisy* label that may be inaccurate, and N is the total number of training samples. The goal of music recommendation training is to learn user preferences from a set of user-item interactions. Formally, the goal is to learn the mapping function $f(\cdot; \Theta) : \mathcal{X} \to \mathcal{Y}$ of the CTR prediction model parameterized by Θ. The optimal parameters Θ^* is obtained by minimizing the binary cross-entropy loss over $\tilde{D}$:

$$L = -\frac{1}{N} \sum_{(x,\tilde{y}) \in \tilde{D}} (\tilde{y} \log p(x) + (1 - \tilde{y}) \log(1 - p(x))) \quad (1)$$

where $p(x) = f(x; \Theta)$ is the predicted CTR value of the model.

Due to the existence of noisy labels in $\tilde{D}$, existing models cannot learn user preferences accurately, resulting in poor performance. Therefore, the goal of our LCD is to correct the noisy labels $\tilde{y}$ during training, and then training on the corrected labels should be approximately equivalent to training on clean labels.

2.2 Adaptive Label Correction for Denoising

Previous works [11, 13, 25] point out that model predictions could magnify useful potential information in data, which can help mitigate the damage of the noisy labels. A straight-forward way to combine model predictions during training is to use a convex combination of noisy labels and model predictions as the adjusted target labels. Bootstrapping [23] is the first robust training approach that proposes to update the target labels based on model predictions. Bootstrapping augments the prediction objective with a notion of consistency, which improves the ability to measure the perceptual consistency of noisy labels. The adjusted label y^* of Bootstrapping is calculated as follows:

$$y^* = \lambda \tilde{y} + (1 - \lambda) p(x) \quad (2)$$

where $\tilde{y}$ is the noisy label, $p(x)$ is the model prediction, and $\lambda \in [0, 1]$ is used to balance the noisy label and the model prediction.

Bootstrapping backpropagates the loss of the adjusted labels rather than the noisy ones, leading to a certain robustness to the label noise issue. Essentially, the updated objective augments the original one with a minimum entropy regularization, which encourages the model to have a high confidence in predicting labels [10, 23]. However, Bootstrapping still has several drawbacks: (1) Bootstrapping starts label correction since the first iteration, which may bring in an instability of the adjusted labels due to the inaccurate model predictions at the beginning of the training. (2) When the model prediction is more accurate than the noisy label, we expect to assign nearly 100% weight on the correct label while we can only assign at most $(1 - \lambda)$ weight on the correct label. (3) Samples that need correction usually have large losses at an early stage of the training, but Bootstrapping corrects every sample with the same weight equally, thus fails to leverage such informative signal to design the weighting scheme.

To overcome the above drawbacks, we propose a novel instance-dependent label correction approach with an adaptive weighting scheme based on the model loss. The intuition is that the model loss of each sample contains rich information and can help effectively differentiate between clean samples and corrupted ones as discovered by existing work [1, 4, 15]. Specifically, in each training

Table 1: Statistics of the experimental datasets.

Dataset	#User	#Item	#Train	#Validation	#Test	#Feature
Product	3.64 M	117 K	77.11 M	10.89 M	10.96 M	69
Last.fm	985	584 K	10.10 M	1.26 M	1.26 M	87

Note: "M" means million, and "K" means thousand.

step, the label correction scheme of a given sample s is as follows:

$$y^* = \omega \times \tilde{y} + (1 - \omega) \times p(x) \tag{3}$$

$$\omega = F(t) \text{ with } t = (l - l_{min})/(l_{max} - l_{min}) \tag{4}$$

where l is the training loss of the sample s, l_{min} and l_{max} are the minimum and maximum sample losses respectively in the batch, and $F(\cdot)$ is the transformation function for obtaining the final weighting value. $F(\cdot)$ can be set to various functions according to the necessity of real scenarios. When it is an identity function, ω is simply the normalization of sample loss; when it is set to a parameterized function, such as some complex distributions with configurable parameters or a differentiable multi-layer perceptrons, it enables more flexible label correction due to higher model capacity. During training, we perform our label correction approach only when the base CTR model is trained sufficiently well, so that the useful underlying information in data can be exploited by model predictions to mitigate the label noise issue.

Discussion. In our experiments, the transformation function $F(\cdot)$ is set as the cumulative distribution function of a Beta distribution $Beta(\alpha, \beta)$, i.e., $F(\cdot; \alpha, \beta)$. When we set $\alpha = \beta = 1$ for F, ω just equals t and increases linearly with the loss l. In this case, our LCD can be reduced to an existing method [4]. However, such linear transformation of the model loss l for label correction is insufficient to capture the complex relationship between the desired adaptive weight ω and l. $F(t; \alpha, \beta)$ enables a more flexible weighting scheme. For example, when $\alpha = \beta < 1$, the model prediction of samples with large loss will be trusted more, while those with small loss will be trusted less; when $\alpha = \beta > 1$, the situation is opposite. Generally, samples that need correction have large losses, but this does not mean that the weight assigned to the model prediction $p(x)$ must be higher for a larger loss. When model predictions are inaccurate, assigning large weight to $p(x)$ for samples with large loss can seriously harm modeling training. Hence, we design our weighting scheme of LCD in Equation (3) and (4). Its effectiveness is validated empirically by experiments in Section 3.3.1.

3 EXPERIMENTS

3.1 Experimental Setup

3.1.1 Datasets. We conduct experiments on an industrial dataset from a music app denoted as Product, and a public music dataset Last.fm [3]. Table 1 summarizes the statistics. In Product, the training set contains one week data, and both the validation and testing sets are composed of one day data. The Last.fm dataset, collected between July 2005 and 2009, is randomly decomposed into a training set, a validation set and a testing set in 8 : 1 : 1.

3.1.2 Base Models and Implementations. We apply our proposed LCD to 8 SOTA CTR models to validate its effectiveness, including factorization machines (FM) [24], wide & deep learning model (Wide&Deep) [6], YoutubeNet [7], DeepFM [12], extreme deep FM (xDeepFM) [20], deep & cross network (DCN) [27], automatic feature interaction learning network (AutoInt) [26], and the improved Deep & Cross Network (DCN-V2) [28]. For all models, the size of embedding vectors is set to 8. Hidden layers of multi-layer perceptron are set to $256 \times 128 \times 64$. The batch size is set to 8192. For LCD, (α, β) is tuned in $\{(0.5, 0.5), (1.0, 1.0), (2.0, 2.0)\}$.

3.1.3 Evaluation Metrics. We adopt AUC [8] and GAUC [35] as evaluation metrics. Meanwhile, we introduce RealImp [30] metric to measure the relative improvement.

Table 2: Experimental results on two datasets.

Dataset	Model	AUC			GAUC		
		Raw	+LCD	RealImp	Raw	+LCD	RealImp
Product	FM	0.7542	0.7577	1.41%	0.5843	0.5872	3.41%
	YoutubeNet	0.7809	0.7854	1.61%	0.5951	0.5985	3.53%
	Wide&Deep	0.7813	0.7841	0.97%	0.5978	0.5994	1.64%
	DeepFM	0.7803	0.7840	1.31%	0.5963	0.5999	3.79%
	xDeepFM	0.7814	0.7823	0.33%	0.5971	0.5987	1.63%
	DCN	0.7827	0.7883	2.00%	0.5970	0.6016	4.82%
	AutoInt	0.7872	0.7891	0.64%	0.5993	0.6015	2.27%
	DCN-V2	0.7911	0.7926	0.50%	0.6045	0.6067	2.16%
Last.fm	FM	0.7990	0.8080	1.12%	0.7594	0.7638	0.57%
	YoutubeNet	0.7927	0.8034	1.35%	0.7558	0.7607	0.65%
	Wide&Deep	0.7954	0.8141	2.35%	0.7580	0.7728	1.95%
	DeepFM	0.8200	0.8225	0.31%	0.7806	0.7821	0.20%
	xDeepFM	0.8208	0.8243	0.43%	0.7797	0.7828	0.40%
	DCN	0.8151	0.8168	0.21%	0.7696	0.7744	0.62%
	AutoInt	0.8114	0.8202	1.08%	0.7680	0.7739	0.76%
	DCN-V2	0.8057	0.8088	0.38%	0.7637	0.7654	0.23%

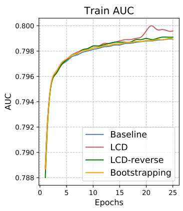
3.2 Performance Comparison

Table 2 shows the results of 8 base models equipped with and without LCD on Product and Last.fm. For both metrics, models with LCD consistently achieve better performance than those without LCD. For example, on the Product, the average relative improvements of AUC and GAUC are 1.10% and 2.91%, respectively. The significant improvements demonstrate the effectiveness and generality of our LCD method, which is mainly because our LCD can encourage perceptual consistency by ensembling the noisy labels and the model prediction, thus reducing the damage of noisy labels. Moreover, the consistent performance gains confirm that our LCD is a model-agnostic denoising method.

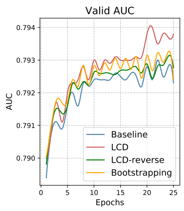
3.3 Ablation Study

3.3.1 Study of Label Correction. We compare LCD with Bootstrapping [23] and a reverse version of LCD (LCD-Re), i.e., $y^* = (1 - \omega) \times \tilde{y} + \omega \times p(x)$. We take DCN-V2 [28] as our baseline, and integrate it with Bootstrapping, LCD or LCD-Re. All models are trained for 25 epochs. We start Bootstrapping from the first iteration as [23], and apply LCD from the 20th epoch. Figure 2(a) and 2(b) show the training and validation curves of AUC w.r.t. the training epoch on Product dataset. We can see that after introducing label correction, the AUC of LCD increases sharply and achieves consistently better results than other methods. For the testing, LCD also outperforms other methods as shown in Figure 2(c).

We conclude that: (1) Compared to the reverse version of LCD, assigning a small weight $1-\omega$ on $p(x)$ for large-loss samples is more practical, which verifies our assertion in the discussion of Section 2. (2) As for Bootstrapping, on one hand, the inaccuracy of model predictions in the early stage of training introduces instability. On the other hand, the static weighting scheme cannot utilize the underlying information of loss.



(a) Training



(b) Validation

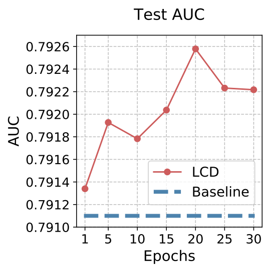
Test results.

Method	AUC	GAUC
Baseline	0.7911	0.6045
LCD	**0.7926**	**0.6067**
LCD-Re[1]	0.7912	0.6039
Boots[2]	0.7914	0.6044

[1] Reverse version of LCD.
[2] Bootstrapping.

(c) Testing

Figure 2: Study of different label correction strategies.



(a) Apply LCD at different epochs.

Test results of AUC.

Model	1-1	0.5-0.5	2-2[1]
FM	0.7574	0.7573	**0.7577**
YoutubeNet	0.7823	**0.7854**	0.7847
Wide&Deep	0.7839	0.7834	**0.7841**
DeepFM	0.7829	0.7833	**0.7840**
xDeepFM	**0.7823**	0.7811	0.7810
DCN	0.7851	0.7877	**0.7883**
AutoInt	0.7886	**0.7891**	0.7885
DCN-V2	**0.7926**	0.7921	0.7922

[1] 2-2 means $\alpha = 2, \beta = 2$.

(b) Setting of Beta distribution.

Figure 3: Study of hyper-parameter settings in LCD.

3.3.2 Hyper-parameter Sensitivity. We analyse the "good" time to introduce LCD by fixing the overall training epochs as 35 and tuning the epoch for applying LCD in {1, 5, 10, 15, 20, 25, 30}. We show the test results of AUC on `Product` dataset in Figure 3(a). We can find that: (1) Applying label correction at any epoch can bring in improvement, demonstrating the robustness and superior denoising ability of LCD. (2) Introducing label correction in a rather late stage, when the base model is sufficiently trained, performs the best. The reason is that model predictions cannot exploit much underlying information in data in an early stage, while the model probably overfits the noisy data at the end of the training.

We also investigate the sensitivity of α and β of the Beta distribution in our weighting scheme. We set $\alpha = \beta \in \{1, 0.5, 2\}$ respectively and apply LCD to 8 CTR models. Figure 3(b) shows the results on `Product` dataset. We can observe that: (1) In most cases, the inferior results of $\alpha = \beta = 1$ indicate that the linear transformation of the model loss is insufficient to capture the complex relationship between the desired weight and the model loss. (2) Our adaptive weighting scheme enjoys high flexibility, since different models can choose its own settings of α and β to achieve good performance.

3.4 Online AB testing

We also deploy our LCD method on a real product and conduct online AB testing to verify its performance. For the control group, the users are provided with recommendations generated by a highly-optimized deep CTR model without LCD. For the experimental group, the users are presented with the recommendations generated by the same CTR model with LCD. As shown in Figure 4, the model trained with LCD contributes up to 1% to 5% promotion in terms of the average music play count and time per user, respectively. Such significant improvements demonstrate the effectiveness of our proposed method.

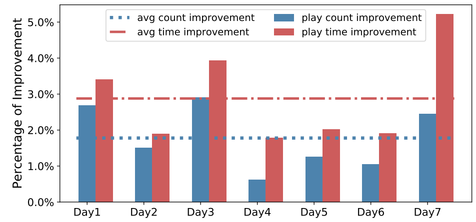


Figure 4: Results from a one-week online AB testing.

4 RELATED WORK

Large amounts of efforts have been made on mining user preferences from various perspectives in CTR prediction [31, 36, 37], such as feature interaction modeling [16, 26], user behavior modeling [33, 34], and learning strategy [2]. However, these existing studies just assume the sample labels to be clean, which is impractical in real scenarios. Meanwhile, a series of methods have been proposed to learn from noisy labels in image classification task, such as designing robust architecture [5, 9], applying robust regularization [17, 22], devising robust loss [4, 23] and selecting clean samples [14, 18]. Among these methods, label correction methods [4, 23], which aims to adjust the training loss using corrected labels, is a simple and effective denoising solution.

In this work, we aim to study the challenging label noise problem in music recommendation, which is parallel to the majority of researches in this field. The proposed method can be applied to a series of existing models to improve their performance. Besides, by designing an adaptive weighting strategy, our proposed denoising method also enables a more flexible label correction scheme compared with existing approaches [4, 23]. One concurrent work [32] also aims to tackle the label noise issue in music recommendation, but it models the problem in an online learning manner and can only deal with false-positive samples.

5 CONCLUSION

Labels are often noisy and naively defined in real recommender systems, resulting in poor generalization performance of existing models. However, little work on music recommendation takes label noise issues into consideration. In this work, we proposed a novel adaptive label correction method for denoising music recommendation by combining the noisy labels and model outputs to encourage a consensus prediction. Particularly, we design a dynamic weighting scheme based on the training loss to assign instance-dependent weights to model predictions. Extensive experiments on both large-scale industrial and public datasets show that our proposed method achieves consistently better results on 8 state-of-the-art CTR prediction models, validating its generality and effectiveness. Moreover, in a one-week online AB testing, our method achieves significant 1% to 5% improvements over the base model in terms of the average music play count and time.

ACKNOWLEDGEMENTS

We appreciate the support from Mindspore[1], which is a new deep learning computing framework.

[1] https://www.mindspore.cn

REFERENCES

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin McGuinness. 2019. Unsupervised label noise modeling and loss correction. In *International Conference on Machine Learning*. PMLR, 312–321.

[2] Guohao Cai, Jieming Zhu, Quanyu Dai, Zhenhua Dong, Xiuqiang He, Ruiming Tang, and Rui Zhang. 2022. ReLoop: A Self-Correction Continual Learning Loop for Recommender Systems. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2692–2697.

[3] Òscar Celma. 2010. *Music Recommendation and Discovery - The Long Tail, Long Fail, and Long Play in the Digital Music Space.* Springer.

[4] Pengfei Chen, Guangyong Chen, Junjie Ye, Pheng-Ann Heng, et al. 2020. Noise against noise: stochastic label noise helps combat inherent label noise. In *International Conference on Learning Representations*.

[5] Xinlei Chen and Abhinav Gupta. 2015. Webly supervised learning of convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 1431–1439.

[6] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.

[7] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.

[8] Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern recognition letters* 27, 8 (2006), 861–874.

[9] Jacob Goldberger and Ehud Ben-Reuven. 2016. Training deep neural-networks using a noise adaptation layer. (2016).

[10] Yves Grandvalet and Yoshua Bengio. 2004. Semi-supervised Learning by Entropy Minimization. In *Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada]*. 529–536.

[11] Melody Guan, Varun Gulshan, Andrew Dai, and Geoffrey Hinton. 2018. Who said what: Modeling individual labelers improves classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[12] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

[13] Bo Han, Quanming Yao, Tongliang Liu, Gang Niu, Ivor W Tsang, James T Kwok, and Masashi Sugiyama. 2020. A survey of label-noise representation learning: Past, present and future. *arXiv preprint arXiv:2011.04406* (2020).

[14] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *arXiv preprint arXiv:1804.06872* (2018).

[15] Lang Huang, Chao Zhang, and Hongyang Zhang. 2020. Self-adaptive training: beyond empirical risk minimization. *Advances in Neural Information Processing Systems* 33 (2020).

[16] Tongwen Huang, Zhiqi Zhang, and Junlin Zhang. 2019. FiBiNET: combining feature importance and bilinear feature interaction for click-through rate prediction. In *Proceedings of ACM Conference on Recommender Systems (RecSys)*. 169–177.

[17] Simon Jenni and Paolo Favaro. 2018. Deep bilevel learning. In *Proceedings of the European conference on computer vision (ECCV)*. 618–633.

[18] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*. PMLR, 2304–2313.

[19] Miao Jiang, Ziyi Yang, and Chen Zhao. 2017. What to play next? A RNN-based music recommendation system. In *2017 51st Asilomar Conference on Signals, Systems, and Computers*. IEEE, 356–358.

[20] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1754–1763.

[21] H. Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, Sharat Chikkerur, Dan Liu, Martin Wattenberg, Arnar Mar Hrafnkelsson, Tom Boulos, and Jeremy Kubica. 2013. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 1222–1230.

[22] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548* (2017).

[23] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. 2014. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596* (2014).

[24] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International conference on data mining*. IEEE, 995–1000.

[25] David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. 2017. Deep learning is robust to massive label noise. *arXiv preprint arXiv:1705.10694* (2017).

[26] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1161–1170.

[27] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*. 1–7.

[28] Ruoxi Wang, Rakesh Shivanna, Derek Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed Chi. 2021. DCN V2: Improved Deep amp; Cross Network and Practical Lessons for Web-Scale Learning to Rank Systems. In *Proceedings of the Web Conference 2021* (Ljubljana, Slovenia) *(WWW '21)*. Association for Computing Machinery, New York, NY, USA, 1785–1797. https://doi.org/10.1145/3442381.3450078

[29] Champika H. P. D. Wishwanath, Supuni N. Weerasinghe, Kanishka H. Illandara, A. S. T. M. R. D. S. Kadigamuwa, and Supunmali Ahangama. 2020. A Personalized and Context Aware Music Recommendation System. In *Social Computing and Social Media. Participation, User Experience, Consumer Experience, and Applications of Social Computing*, Gabriele Meiselwitz (Ed.). Springer International Publishing, Cham, 616–627.

[30] Ling Yan, Wu-jun Li, Gui-Rong Xue, and Dingyi Han. 2014. Coupled group lasso for web-scale ctr prediction in display advertising. In *International Conference on Machine Learning*. PMLR, 802–810.

[31] Weinan Zhang, Jiarui Qin, Wei Guo, Ruiming Tang, and Xiuqiang He. 2021. Deep Learning for Click-Through Rate Estimation. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*. 4695–4703.

[32] Xiao Zhang, Sunhao Dai, Jun Xu, Zhenhua Dong, Quanyu Dai, and Ji-Rong Wen. 2022. Counteracting User Attention Bias in Music Streaming Recommendation via Reward Modification. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2504–2514.

[33] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2018. Deep Interest Evolution Network for Click-Through Rate Prediction. *CoRR* abs/1809.03672 (2018).

[34] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1059–1068.

[35] Han Zhu, Junqi Jin, Chang Tan, Fei Pan, Yifan Zeng, Han Li, and Kun Gai. 2017. Optimized cost per click in taobao display advertising. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2191–2200.

[36] Jieming Zhu, Quanyu Dai, Liangcai Su, Rong Ma, Jinyang Liu, Guohao Cai, Xi Xiao, and Rui Zhang. 2022. BARS: Towards Open Benchmarking for Recommender Systems. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*. ACM, 2912–2923.

[37] Jieming Zhu, Jinyang Liu, Shuai Yang, Qi Zhang, and Xiuqiang He. 2021. Open Benchmarking for Click-Through Rate Prediction. In *The 30th ACM International Conference on Information and Knowledge Management (CIKM)*. 2759–2769.