A Guided Learning Approach for Item Recommendation via Surrogate Loss Learning

Ahmed Rashed
ahmedrashed@ismll.uni-hildesheim.de
Information Systems and Machine Learning Lab, University of Hildesheim, Germany

Josif Grabocka
grabocka@informatik.uni-freiburg.de
Department of Computer Science, University of Freiburg, Germany

Lars Schmidt-Thieme
schmidt-thieme@ismll.uni-hildesheim.de
Information Systems and Machine Learning Lab, University of Hildesheim, Germany

Problem Definition

In item recommendation tasks, there exist a set of users U $:= \{u_1, u_2, \ldots, u_N\}$, a set of itemsI $:= \{i_1, i_2, \ldots, i_M\}$, and a sparse binary interaction matrix$R \in \mathbb{R}^{N \times M}$ that indicate user's implicit preferences on items. The main goal of the item recommendation task is to present user with short personalized ranked lists of relevant items based on their likelihood scores.

To do so, models have to be optimized with respect to maximizing the ranked list's NDCG quality measure

$$\mathrm{DCG@k} = \sum_{p=1}^{k} \frac{rel_p}{\log_2(p+1)} \tag{1}$$

$$\mathrm{NDCG@k} = \frac{\mathrm{DCG}_k}{\mathrm{IDCG}_k} \tag{2}$$

where rel_p is the relevance of the item at position p. IDCG is the best possible value for $IDCG_k$ for the best possible ranking of relevant items at threshold k.

Challenges and Contributions

- As NDCG is non-differentiable, it cannot be optimized by gradient-based optimization procedures directly.

- Surrogate losses have been engineered that aim to make learning recommendation and ranking models that optimize NDCG possible. However, this approach suffers from stability issues similar GAN.

- An alternative guided learning approach for surrogate losses are proposed.

- Learning the surrogate loss in parallel to learning the recommender model using the original logistic loss to stabilize the learning phase.

Methodology

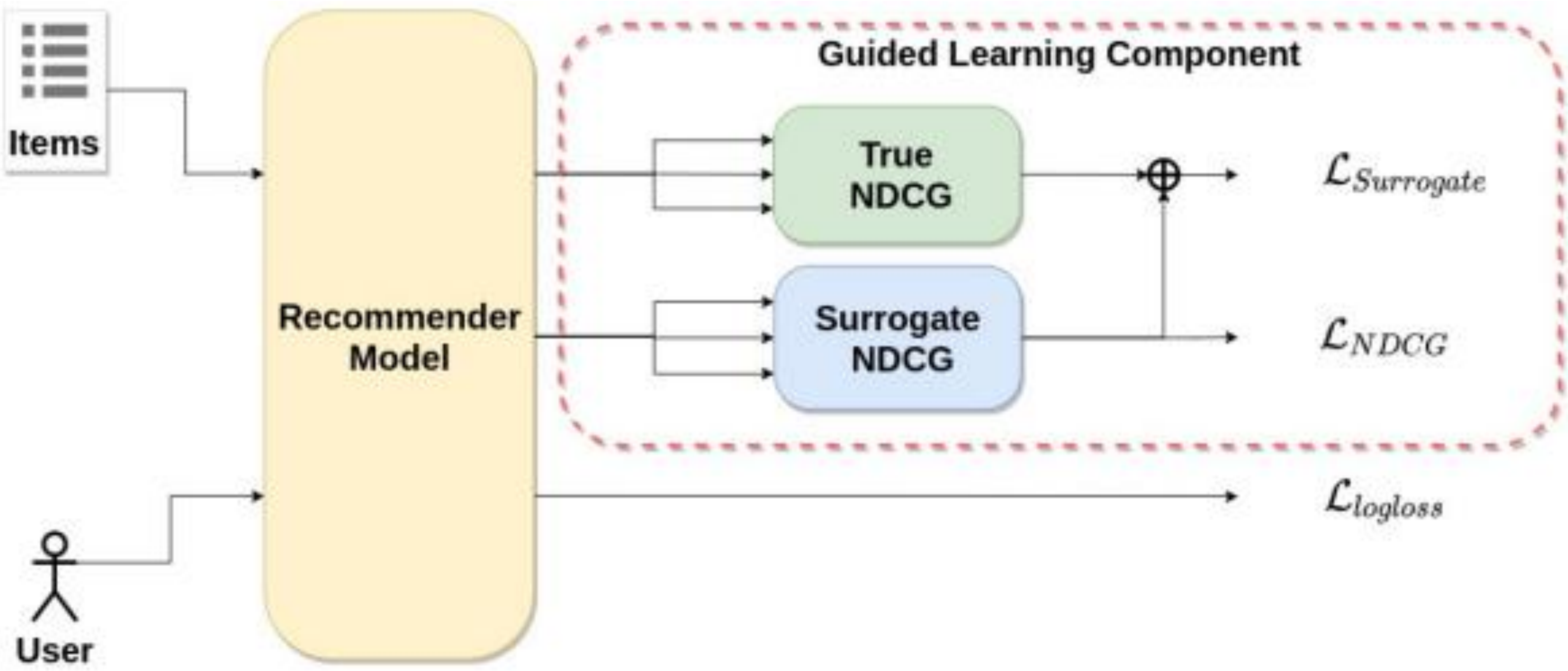

Figure 1: GuidedRec Architecture

Logloss

- In the first step the recommender model is optimized by minimizing its original log loss function.

$$\mathcal{L}_{\text{logloss}}(\mathcal{U},\mathcal{I},\mathcal{R}) = -\beta\left(\sum_{(u,i)\in(\mathcal{U},\mathcal{I},\mathcal{R})} r_{ui}\log(\hat{r}_{ui}) + (1-r_{ui})\log(1-\hat{r}_{ui})\right) \qquad (9)$$

$$\hat{r}_{ui} = f(u,i;\theta_f)$$

where f is the recommender system model with parameters θ_f. β controls the importance of the log loss function to the overall learning procedure.

Surrogate Loss Model

Given the list-wise predicted scores $\hat{y} \in \hat{\mathcal{Y}}$ and ground-truth scores$y \in \mathcal{Y}$, we define the add-on surrogate loss model that estimate the NDCG of the predicted scores as follows

$$x = zip(y, \hat{y}) = \{(r_1, \hat{r}_1), ..., (r_K, \hat{r}_K)\}, x \in R^{K \times 2} \tag{4}$$

$$\phi_{\mathrm{MLP}}(x) = v(Flatten(x); \theta_v) \tag{5}$$

$$z = g(x; \theta_g) \tag{6}$$

$$\phi_{\mathrm{NFM}}(x) = e\left(\frac{1}{K} z z^T \mathbf{1}; \theta_e\right) \tag{7}$$

$$\hat{\mathrm{NDCG}}(y, \hat{y}) = h(\phi_{\mathrm{NFM}}(x) \odot \phi_{\mathrm{MLP}}(x); \theta_h) \tag{8}$$

where g, v, e and h are a series of non-linear fully connected layers with weights $\theta_g, \theta_v, \theta_e$ and θ_h respectively, and $\odot$ denotes the element-wise product of two vectors.

Surrogate Loss Model

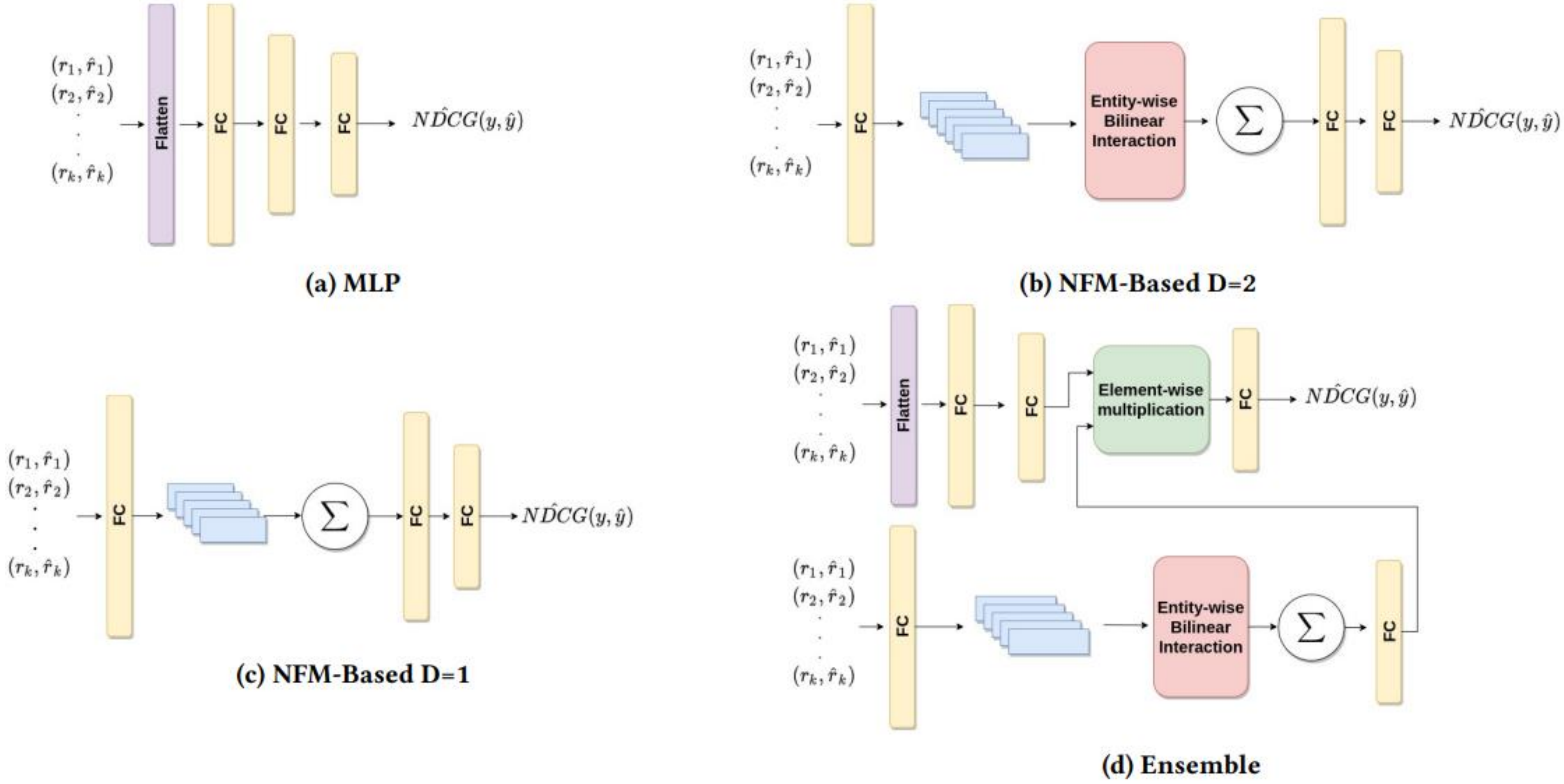

Figure 3: Different candidate architectures for the surrogate loss model

Surrogate Loss and NDCG Loss

- **Surrogate Loss**

$$\mathcal{L}_{Surrogate}(\mathcal{Y}, \hat{\mathcal{Y}}) = \sum_{(y,\hat{y}) \in zip(\mathcal{Y},\hat{\mathcal{Y}})} (NDCG(y, \hat{y}) - N\widehat{DC}G(y, \hat{y}))^2 \qquad (10)$$

- **NDCG Loss**

$$\mathcal{L}_{NDCG}(\mathcal{Y}, \hat{\mathcal{Y}}) = \sum_{(y,\hat{y}) \in zip(\mathcal{Y},\hat{\mathcal{Y}})} -N\widehat{DC}G(y, \hat{y}) \qquad (11)$$

The Pseudo-code of GuidedRec

Algorithm 1: GuidedRec ($\mathcal{U}, \mathcal{I}, \mathcal{R}$)

input : A set of users $\mathcal{U}$, a set of items $\mathcal{I}$ and their ground-truth relevance scores $\mathcal{R}$

output : The predicted items scores $\hat{\mathcal{Y}}$

Initialize the surrogate model and the recommender system model parameters Θ and θ_f

for *E epochs* **do**

  Draw a list-wise batch ($\mathcal{U}, \mathcal{T}, \mathcal{Y}$)

  Convert the list-wise batch to pair-wise batch using Q_{in}

  Predict items scores $\hat{\mathcal{R}}$ using Eq. (3)

  Update θ_f by mnimizing Eq. (9)

  Convert the pair-wise predicted scores $\hat{\mathcal{R}}$ to list-wise scores $\hat{\mathcal{Y}}$ using Q_{out}

  Calculate the average true NDCG value using Eq. (1) and (2)

  Predict the surrogate NDCG value using Eq. (8)

  Update Θ by minimizing Eq. (10) while fixing θ_f

  Update θ_f by maximizing Eq. (11) while fixing Θ

end

Experiments

Table 2: Performance comparison of the GuidedRec learning approach

Loss Function	MovieLens 100k HR@10	MovieLens 100k NDCG@10	MovieLens 1M HR@10	MovieLens 1M NDCG@10	MovieTweeting HR@10	MovieTweeting NDCG@10
Logloss	0.628	0.368	0.681	<u>0.413</u>	0.723	0.567
Gumbel-NDCG [1]	0.633	0.372	0.684	0.410	0.722	0.567
Approx-NDCG [2, 15]	0.618	0.355	0.684	0.409	0.714	0.559
ListMLE [18]	0.618	0.366	0.675	0.401	0.723	0.567
Softmax [14]	0.620	0.346	0.656	0.374	0.724	<u>0.568</u>
Pair-wise logloss [14]	<u>0.645</u>	0.377	<u>0.688</u>	0.409	0.724	<u>0.568</u>
Neural-Sort [5]	<u>0.645</u>	<u>0.379</u>	0.663	0.385	<u>0.726</u>	0.567
GuidedRec w/o logloss	0.437	0.241	0.292	0.170	0.724	0.567
GuidedRec w/ logloss	**0.661****	**0.386****	**0.695***	**0.422****	**0.732****	**0.571****
Benchmark Models						
TopPopular	0.406	0.219	0.467	0.260	0.724	0.565
NeuMF [7]	0.621	0.356	0.660	0.393	0.724	0.567

Significantly outperforms the best baseline at the: ** 0.01 and * 0.05 levels.

Experiments

Table 4: Performance comparison between different architectures for the surrogate loss on the MovieLens 100K

Model	HR@10	NDCG@10
MLP	0.641	0.375
NFM-Based D=1	0.647	0.377
NFM-Based D=2	0.646	0.378
Ensemble	**0.661****	**0.386****

Significantly outperforms the best baseline at the: ** 0.01 and * 0.05 levels.

Table 6: Runtime comparison between GuidedRec and other proxy loss functions

Loss Function	Average batch runtime in seconds
Logloss	4.185
Gumbel-NDCG [1]	12.451
Approx-NDCG [2, 15]	4.709
ListMLE [18]	4.557
Softmax [14]	4.680
Pair-wise logloss [14]	5.567
Neural-Sort [5]	4.812
GuidedRec w/ logloss	16.525