Pairwise loss regularization for recommendations explanation

Alexandre Chanson
University of Tours
alexandre.chanson@univ-tours.fr

Nicolas Labroche
University of Tours
nicolas.labroche@univ-tours.fr

Patrick Marcel
University of Tours
patrick.marcel@univ-tours.fr

Willeme Verdeaux
Kalidea Up, University of Tours
willeme.verdeaux@up.coop

ABSTRACT

Recommender systems are notoriously complex systems for which providing a local explanation on why a certain item is proposed to a specific user is still a challenging task. Most explanation approaches focus on predicting the rating of items thereby minimizing some discrepancy with the real ratings by means of traditional loss functions (e.g., sum of squares error). However, most of the times, ratings may not fully embrace user preferences concerning the ranking of items. To better embrace user preferences, methods based on ranking losses have been proposed either to recommend or to explain why an item is recommended. Although effective at identifying the most prominent items, these methods fail to capture a realistic value for the rating attached to their explanation. This loss attached to the semantic of the recommendation can in turn arm the trust of a user in the explanation. In this paper, we propose and discuss experimental results of a simple yet effective novel loss schema that balances ranking and rating losses to provide a best of both world explanation.

1 INTRODUCTION

Recommender systems (RS) are decision systems that help users to decide between several options or items. In their classical form, RS are formalized as a function which inputs a user and an item and whose final objective is to produce an ordered list of items, otherwise called a permutation (or ranking), for this particular user. As stated in [2], the scoring function at the core of each RS can be expressed as follows:

$$f : U \times I \to \mathbb{R} \qquad (1)$$

where U is the set of users, I is the set of items and $\mathbb{R}$ is the definition domain of the scores as a totally ordered set.

These systems are ubiquitous in our everyday life and they span numerous domains from entertainment [7, 10, 15] to more high stakes domain such as medicine [19]. However, as these systems rely on complex models such as matrix factorization or deep learning methods optimizing complex loss functions [14, 22], it is very difficult for the end user to get the rationale for a particular recommendation.

As a consequence, as noted by [24], this non-transparency of RS (termed as black-boxes) and new regulation such as GDPR and its "right to explanation" call for the development of explainable recommender systems (denoted as XRS hereafter). [20] identify 6 main objectives for XRS as it might: improve the adequacy of retrieved items to user interests and facilitate the whole process ((i) effectiveness, (ii) efficiency, (iii) satisfaction), persuade users

to access / buy new items ((iv) persuasiveness), reinforce confidence in the RS ((v) trust), and finally make the rationale of each recommendation more explicit to the user ((vi) transparency).

There exists a large field of study for XRS [23] ranging from intrinsic methods - improving a RS by enriching its ordered list of items with an explanation - to post-hoc model agnostic approaches - whose goal is solely to generate explanations based on the output of an external RS.

In the present work, we are interested in post-hoc model agnostic explanation models, following the extension of the traditional LIME approach [18] to the RS domain [4, 6, 17]. These systems build a local surrogate model of a recommender system by determining a simple (generally a linear) relation between features of an interpretable space composed of additional features qualifying users and/or items. All these surrogate models are trained based on a loss that minimizes the rating prediction error between the ground truth as represented by the original user-item matrix and the output of the surrogate model. In the end, the weights attached to the interpretable features of the surrogate model are the expected local explanation for a specific user-item rating.

Interestingly, we have shown in [4] that it is possible to improve the relevance of these explainer systems, i.e. their ability to discover the important interpretable features, by replacing the rating error loss by a *pairwise ranking loss*. This loss is based on a cross-entropy measure that better preserves the order of the items for a specific user rather than the actual ratings.

However, the observed discrepancy in the predicted ratings, even if the identified features are correct, induces a loss in the semantic attached to the recommendation and thus may hamper the trust in the system. This mechanism is studied in [12] where an explanation should be faithful to the model it explains (in our case minimizing the rating error) while being plausible to the users (in our case preserving ranking of items). Moreover, in [4], we show that a simple sigmoid-based normalization of the ratings, although effective, does not produce acceptable errors compared to other post-hoc explainers based on rating loss.

For these reasons, we study in this paper the possibility to improve the pairwise ranking loss and we contribute with a simple yet effective regularization term to achieve a "best of both world" optimization that maintains the relevance of interpretable features as well as a reduced rating error.

The paper is organized as follows: Section 2 introduces the conclusions of [4] needed to understand the contribution presented in Section 3. Finally, Section 4 presents promising preliminary results and Section 5 concludes and discusses future works.

2 PREREQUISITES

In this section, we briefly describe the main principles behind the post-hoc explanation approach for RS that we developed in [4], and illustrated in Figure 1. We also position the contribution with regard to recent counterfactual works on RS.

2.1 Problem statement

In [4, 6], we consider the case of an explanation instance $\langle u, i, f(u, i)\rangle$ where the aim is to explain why the rating $f(u, i)$ is produced by the black-box f for the user $u \in U$ and item $i \in I$.

The idea of post-hoc explanation methods consists in training a surrogate model g defined in an interpretable feature space Z.

When g is a simple linear model, explaining $\langle u, i, f(u, i)\rangle$ reduces to a regression problem where a loss $\mathcal{L}$ is minimized between the black-box output $f(u, i)$ and the surrogate model $g = \mathbf{w}.\phi(u, i)$, where $\phi : U \times I \to Z$ projects any user-item (u, i) into the interpretable feature space Z.

Considering that a training set $\mathcal{T}_{train}^{(u)} \subset Z$ is available, the problem of determining the best explanation $e_f(u, i)$ is formalized as follows in [4]:

$$e_f(u, i) = \underset{\mathbf{w} \in \mathbb{R}^n}{\operatorname{argmin}} \sum_{z \in \mathcal{T}_{train}^{(u)}} \left(f(\phi^{-1}(z)) - \mathbf{w}.z \right)^2 + \lambda \|\mathbf{w}\|_1 \quad (2)$$

where the last term $\|\mathbf{w}\|_1$ achieves the simplest explanation by only retaining the most interesting features, acting as in a LASSO regression model.

The core idea of [4] consists in replacing the pointwise loss of Equation 2 by a pairwise loss that better captures the preferences over items ranking rather than ratings for user u.

2.2 Introducing a pairwise loss

The main intuition of the pairwise loss proposed in [4] is that if it is possible for a surrogate model g to estimate the ratings, then it is possible, considering the pair of training instances z_i and z_j, $i \neq j$ from the training set $\mathcal{T}_{train}^{(u)}$ to train g so that the ground truth in user-item matrix R is preserved by the ratings $g(z_i) = \mathbf{w} \cdot z_i$ and $g(z_j) = \mathbf{w} \cdot z_j$ as estimated by the surrogate model.

More precisely, it is possible to build a preference ground truth for training instances z_i and z_j by defining the target probability that item i is preferred to item j for the black-box f as:

$$P(z_i \succ_f z_j) = \begin{cases} 1 & \text{if } f(\phi^{-1}(z_i)) > f(\phi^{-1}(z_j)) \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

Similarly, it is possible to express the probability that reflects the preference over items $i \neq j$ for the same training instances z_i and z_j for the surrogate model as follows:

$$P(z_i \succ_{\mathbf{w}} z_j) = \frac{1}{1 + e^{(\mathbf{w} \cdot z_i - \mathbf{w} \cdot z_j)}} \quad (4)$$

where the computation of estimated ratings $\mathbf{w} \cdot z_i$ for instance z_i and $\mathbf{w} \cdot z_j$ for instance z_j is similar to the linear explanation model presented in the previous section.

Based on the previous two equations, it is possible to define a new loss function that measures the discrepancy between the ground truth probabilities and the surrogate probabilities, for any recommender system f and surrogate model g over a training set $\mathcal{T}_{train}^{(u)}$ as follows:

$$\mathcal{L}(f, g, \mathcal{T}_{train}^{(u)}) = \sum_{(z_i, z_j) \in \mathcal{T}_{train}^{(u)}} C(P(z_i \succ_g z_j), P(z_i \succ_f z_j)) \quad (5)$$

with z_i and z_j are two instances drawn from $\mathcal{T}_{train}^{(u)}$ and C is a cross-entropy defined as follows:

$$C(S_{i,j}, Y_{i,j}) = -Y_{i,j} log(S_{i,j}) - (1 - Y_{i,j}) log(1 - S_{i,j}) \quad (6)$$

where, for the sake of readability, we set $S_{i,j} = P(z_i \succ_{\mathbf{w}} z_j)$ and $Y_{i,j} = P(z_i \succ_f z_j)$.

2.3 Implementation of the explainable RS

We detail here the main steps presented in [4] to implement such an explanation system.

Definition of an interpretable feature space. This feature space is a reduction of the original user-item ratings space. Here the interpretable feature space is the set of items $I \setminus \{i\}$ as illustrated in Figure 1- (1). As such, this space allows for a simple representation of any user over a set of directly understandable dimensions, without the need for external metadata.

Setting a training set for the surrogate model. Although several types of locality definitions are envisioned such as clustering or gradual perturbations on ratings (Figure 1- (2)), the pointwise and pairwise methods, named LIRE-P and LIRE-PP respectively, (see Figure 1- (3)) relies on feature perturbations.

These perturbations are defined for an explanation instance $\langle u, i, f(u, i)\rangle$ as random modifications of the values of the vector, denoted $\mathbf{t^u}$, that corresponds to a projection of the row u of the user-item matrix R in the interpretable space $I \setminus \{i\}$. Perturbations are based on a Gaussian distribution $\mathcal{N}(0, \sigma_j)$. The value of σ_j for each item $j \neq i \in I$ is computed as the average observed deviation of all ratings in the training sample. Then, we model as a Bernoulli process the chance to modify each non-zero element of $\mathbf{t^u}$.
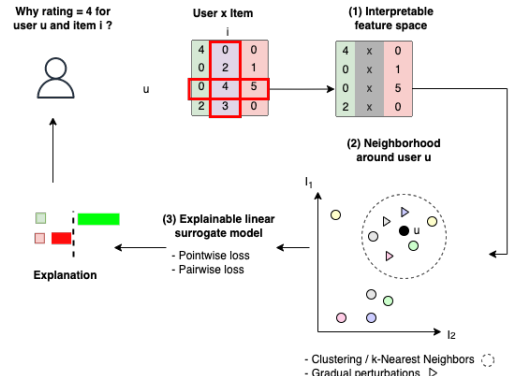

Figure 1: LIRE explanations [4] are presented in an interpretable feature space (1), constructed locally (2) and using a pairwise loss (3).

Implementation of the pairwise loss. In [4], the loss function in Equation 5 is implemented and optimized as an improved 1-layer RankNet [5] model without bias.

This RankNet architecture, during its training, inputs two instances z_i and z_j from the surrogate training set $\mathcal{T}_{train}^{(u)}$, then re-adjusts its parameters $\mathbf{w}$ which are feature weights so that the corresponding score for each instance, $\mathbf{w} \cdot z_i$ and $\mathbf{w} \cdot z_j$, does not violate the pairwise preference relation implied by the black-box ratings.

Interestingly, in [4], we add to this generic 1-layer RankNet model a sigmoid activation function ensuring that the computed scores $\mathbf{w} \cdot z_i$ and $\mathbf{w} \cdot z_j$ are constrained in the range of the use case.

This activation function $g(r)$ is traditionally defined as follows:

$$g(r) = \frac{1}{1 + e^{-r}}, \forall r \in \mathbb{R} \qquad (7)$$

2.4 Positionning

Explanations for RS should help users better understand why an item has been recommended, or qualities of items to decide whether to accept them or not [20]. One way to achieve the best faithfulness to RS is to develop intrinsic models [23]. Such as [1], tailored to match more specifically matrix factorization models , or [8] focusing on Bayesian Pairwise Recommendations, while taking care of the exposure bias of these models.

In this paper we focus on model-agnostic post-hoc local explanation models such as LIME-RS [3, 17], a direct adaptation of LIME [18] to RS, where each cell of the user-item matrix is encoded as a 1-hot vector decomposed in three parts: user, item and metadata information. This design causes multiple weaknesses, noticeably results depend on the availability and quality of metadata information. In our method, we only rely on the user-item matrix used to train the RS without the need for external data. Moreover, in order to be also plausible [12] to the user, we aim at optimizing the ranking of all items while simultaneously improving faithfulness to the original ratings.

One way to further explain RS is to produce counterfactual explanations [21], i.e. to determine the minimal set of scored items or actions that if not undertaken by the user, would have changed the recommendation. [9] present a method that operates on heterogeneous information networks (HIN), that are graphs representing users and items (nodes) and several types of actions (edges) and for which explanations are computed as random walk inspired by PageRank-like scores. This approach needs all RS training data (all available user item interactions) and can only work within the HIN formalism. In [13], authors propose to limit the accessed data and adapt the counterfactual explanation to any RS as in our case. Their model relies on heuristics to explore efficiently the powerset of user actions based on a time budget. However, in collaborative RS settings, their solution should also take into account actions of other users to explain why a recommendation was made to a user. In this case, the size of the set of possible interventions explodes becoming intractable.

This still justifies the need for approximate methods like ours that provide a rationalization of plausible explanations. These approximations can be more efficiently computed and already take into consideration possible relations between users and items and are thus more aligned with collaborative RS.

3 CONTRIBUTION

In [4], it is shown that, while normalization greatly helps to reduce the error on the predicted ratings, this error remains too high to fully capture the semantics attached to the ratings.

Our contribution aims at preserving the high interpretable feature relevance of the pairwise explainer proposed in [4], while improving its ability to correctly estimate the ground truth ratings so as to maintain the semantic attached to the recommendation.

Interestingly, in [16], regularization of neural networks architecture is discussed as a mean to preserve generalization and avoid overfitting. According to the findings of this paper, it seems more beneficial to control regularization during training of the neural network. The general schema of such regularization is to add a penalty term $\mathcal{L}_{\mathcal{H}}$ to the standard loss function $\mathcal{L}$. The modified cost function $\mathcal{L}_{\mathcal{I}}$ is as follows:

$$\mathcal{L}_{\mathcal{I}} = \mathcal{L} + \lambda \mathcal{L}_{\mathcal{H}} \qquad (8)$$

where λ is a scalar that determines the influence of $\mathcal{L}_{\mathcal{H}}$.

In this, our primary objective is not to reduce overfitting but to gain an additional property to the produced explanation: (i) we want our solution to rely primarily on the pairwise loss defined in Equation 5, to preserve the ranking of interpretable features, but (ii) we also aim at minimizing errors in the rating predictions so as to reduce the semantic loss attached to our surrogate model.

For these reasons, we propose to set $\mathcal{L}$ as in Equation 5 and to regularize with a penalty term derived from Equation 2, that accounts for the error in the ratings. More precisely, as the main term $\mathcal{L}$ of the loss relates to pairwise preferences, the second term $\mathcal{L}_{\mathcal{H}}$ that accounts for errors in ratings should take care of minimizing this error for both training instances z_i and z_j as follows:

$$\mathcal{L}_{\mathcal{H}} = \left(f(\phi^{-1}(z_i)) - \mathbf{w}.z_i \right)^2 + \left(f(\phi^{-1}(z_j)) - \mathbf{w}.z_j \right)^2 \qquad (9)$$

Experiments section analyses to which extent, introducing a regularization term can be beneficial to produce more meaningful explanations.

4 EXPERIMENTS

This section describes the experiments conducted to assess the effectiveness of our approach in terms of explanations quality and the way our regularization impacts the behaviour of the pairwise explanation algorithm. To do so, we have set up several experiments reported hereafter that answer the following research questions:

- Q1 How our approach compares in terms of explanation "quality" with the approaches detailed in [4]? Noticeably, do we reach a better balance between predicted rating error and predicted interpretable features ranking as expected by the introduction of a regularization term?
- Q2 Does this approach still scale to larger datasets? As in [4], we run a test on MovieLens 20M entries to evaluate the impact of the regularization term on the performances.
- Q3 How the value of hyper-parameter λ in Equation 8 impacts the results? We run comparative experiments with parameter $\lambda \in \{100, 50, 10, 5, 1, 0.8, 0.6, 0.4, 0.2, 0.010, 0.008, 0.006, 0.004, 0.002, 0\}$ that ranges from a strong regularization of the optimization towards the respect of rating errors, to a situation where the optimization is only based on respect of ranking preferences similar to the LIRE-PP algorithm [4] described in Section 2.2.

We present hereafter our experimental protocol: the datasets, the black-boxes, and finally the evaluation metrics.

4.1 Experimental protocol

Datasets. To ease the comparisons with [4, 6], we consider the same 2 well-known datasets from MovieLens. Each dataset describes 5-star ratings and free-text tagging activity from Movie-Lens website. We limit our main tests to the $100K$ MovieLens dataset [11] with 610 users and $9,724$ items, as we aim at testing several scenarios and parameters. An evaluation on the MovieLens 20M entries dataset ($20,000,263$ ratings generated by $138,493$ users for $27,278$ movies) is done to attest that our newly regularized approach can scale to larger volumes of data.

Black-boxes. In our setup, we want to determine to which extent our method is able to identify correctly the most important interpretable features as well as reducing the discrepancy in the ratings prediction. To this end, we consider a linear white-box model playing the role of the complex predictor. Similar to [4, 6], the idea is to simulate a linear black-box recommender system, for which we know by advance the relative weights of the features and that can rate an item based on a weighted linear combinations of scores on other items. Then, we expect a good surrogate model to be able to learn a linear model very close to this simulated "black-box". Here, the objective is to challenge a surrogate model in a controlled environment where it is possible to precisely estimate to which extent relevant features are identified. Interestingly, the task of predicting ratings from a linear black-box model, even if it is simpler than estimating the output of a complex black-box RS, is still challenging for the pairwise approach as reported by [4, 6]. More precisely, for a given explanation instance $\langle u, i, f(u, i) \rangle$, we pick at random 10 items that are evaluated by user u (excluding item i). These items are assigned random non-zero weights uniformly chosen in $]0, 1]$.

Evaluation metrics. Similar to [4, 6], in our tests, for each configuration of parameter λ, 50 explanations are produced to estimate the evaluation metrics.

- the **accuracy** to the "black-box" model is computed as a Mean Absolute Error (MAE) between the prediction of the black-box and the prediction of the surrogate model in the interpretable space;
- the **recall** is expressed as the ratio of features from the white-box model f that are discovered by the surrogate model g (i.e. features whose weights exceed 0 in the model g). With $\mathcal{F}$ being the set of features of a model, then the $recall(f, g)$ is defined as:

$$recall(f, g) = \frac{\mathcal{F}(f) \cap \mathcal{F}(g)}{\mathcal{F}(f)} \tag{10}$$

- the **feature ranking quality** considers the ordering of the relevant features in the explanation. To achieve the comparison between the orders of features in the white-box model and in the surrogate model, we use the Normalized Discounted Cumulative Gain (NDCG) measure at rank ρ (NDCG@ρ) for values of $\rho \in \{3, 5, 10\}$.
- the **computation time** is estimated in seconds as the average duration for one explanation. All experiments were conducted on a 512GB RAM RTX A6000 GPU paired with dual xeon Gold 6240R processors.

4.2 Sensitivity to λ parameter value

Results of the first experiment are reported in Table 1 that presents the average recall, ndcg@10, MAE and computation times when varying the weight λ of the regularization term. These scores are estimated over 50 distinct explanation instances.

First, it can be seen, as expected, that preference-based measure scores (recall and ndcg@10) increase as λ decreases. Indeed, the higher λ, the stronger the regularization of the ranking loss based on the respect of true rating values. Note that when $\lambda = 0$, the algorithm is similar to LIRE-PP. As a consequence, without any regularization, our algorithm shows an even stronger limit in capturing a credible rating, hence exhibiting a very large MAE error around 6.671 as shown in Table 2.

Conversely, the value of λ does not seem to have a strong impact on the MAE score (unless $\lambda = 0$, see Table 2). Interestingly,

this means that even a small regularization weight on the rating errors is enough to counterbalance the influence of the pairwise ranking term of the loss in Equation 8. This could be explained by the fact that, as the ranking loss term is expressed as a cross-entropy, its values are generally close to 0 (even if in theory not bounded on its highest achievable scores). As a consequence, any λ value for the second loss term, different from 0, may bring some effective regularization.

Finally, computation times are also mostly invariant in our tests when varying λ. Interestingly, larger values of $\lambda = 10$ or $\lambda = 100$ lead to an increase in the observed computation times. This can be due to the loss function complexity that exhibits more conflicting objectives for higher values of λ. This, in turns, may cause the optimization process to get stuck more often into local solutions and thereby would need more restarts to find an appropriate solution. We leave as future work an in depth study of the relation between the loss function and λ as observed results pertain to the data, the optimizer (ADAM in our case) and the way it balances loss terms according to λ.

From all these tests, we observed that introducing our new regularization term can lead to substantial improvements of the solution. For example, when $\lambda = 0.0002$ our approach captures correctly user preference with very high scores ($recall = 0.794$ and $ndcg@10 = 0.931$) while minimizing the MAE error ($mae = 0.079$). In following experiments, we set the value of $\lambda = 0.0002$.

4.3 Comparative experiments

Table 2 details comparative experiments between the pointwise loss algorithm LIRE-P [4, 6] and 3 distinct pairwise loss configurations: (i) with no sigmoid normalization denoted LIRE-PP-NS, (ii) with sigmoid normalization denoted LIRE-PP and finally, (iii) with our new error rating regularization term denoted LIRE-PP-R. Interestingly, only LIRE-P and LIRE-PP are detailed originally in [4], with only recall and ndcg@10 scores provided. As the focus of the paper is on pairwise approaches, results from [4] for LIRE-PP have been completed and re run on the same conditions so as to assess a fair comparison. Scenarios (i) and (iii) are new to this paper.

First, it can be seen from Table 2 that pointwise loss method is unable to capture the user preferences as measured by recall and ndcg@10 scores. Only pairwise methods achieve very good performance in this regards.

Second, it can be observed that LIRE-PP-NS and LIRE-PP do not achieve a good estimation of the real ratings that the black-box outputs, hence causing very large MAE scores. Of course the sigmoid normalization helps keeping rating errors under control with a MAE of around 6.671 that is much below the score of around 15 otherwise, but this is not enough to capture realistic rating values. Only our novel LIRE-PP-R manages to minimize the MAE score around 0.079 thanks to the new normalization.

Although, we notice the introduction of regularization drastically reduces standard deviation around MAE while preserving low deviation around recall and ndcg@10 scores.

Finally, introducing a regularization term does not seem to affect computation times as attested by the last column of Table 2.

4.4 Large scale analysis

Finally, Table 2 reports experiments on MovieLens with 20 millions ratings and for 50 explanations. It can be seen that, similarly to LIRE-P and LIRE-PP [4], LIRE-PP-R can also run effectively on larger size datasets. Moreover, the expected balance between

λ	100	10	1	0.2	0.01	0.002	0
Recall	0.516 ± 0.180	0.528 ± 0.170	0.604 ± 0.171	0.614 ± 0.157	0.794 ± 0.115	0.794 ± 0.113	0.810 ± 0.140
ndcg@10	0.687 ± 0.173	0.667 ± 0.148	0.742 ± 0.161	0.747 ± 0.141	0.869 ± 0.092	0.931 ± 0.054	0.933 ± 0.050
MAE	0.138 ± 0.123	0.125 ± 0.107	0.113 ± 0.108	0.202 ± 0.281	0.105 ± 0.110	0.079 ± 0.063	6.671 ± 4.563
Time (s.)	42.070 ± 0.953	38.841 ± 4.052	33.410 ± 1.999	33.944 ± 2.212	35.082 ± 2.649	34.405 ± 2.403	34.947 ± 2.425

Table 1: Sensitivity of our approach to the λ parameter value. Evaluation metrics are preference-based (recall and ndcg@10), rating-based (MAE rating error) and time-based. Provided results are averaged over 50 runs.

Methods	Recall	ndcg@10	MAE	Time (s.)
Lire-P	0.212 ± 0.206	0.255 ± 0.250	NA	NA
LIRE-PP-NS	0.780 ± 0.231	0.861 ± 0.230	15.009 ± 4.304	37.079 ± 10.852
LIRE-PP [4]	0.810 ± 0.140	0.933 ± 0.050	6.671 ± 4.563	34.947 ± 2.425
LIRE-PP-R	0.794 ± 0.113	0.931 ± 0.054	0.079 ± 0.063	34.405 ± 2.403
LIRE-PP-R 20M	0.862 ± 0.109	0.942 ± 0.072	0.078 ± 0.150	51.173 ± 9.584

Table 2: Upper part: comparative results on the single white-box experiment. New proposed method LIRE-PP-R provides a better balance between all evaluation criteria. Lower part: performance of our regularized LIRE-P Pairwise on Movielens 20M for $\lambda = 0.002$. Provided results are averaged over 50 runs.

accuracy and preference preservation is still achieved with very good scores. Computation times show that real time explanation may not yet be achieved by our method, since, on our test machine, approximately 51 seconds are needed on average to converge for a single explanation.

5 CONCLUSION

This paper improves an existing posthoc explainer for RS, named LIRE-PP, based on a pairwise loss that preserves users relative preferences between items at the cost of an incorrect estimation of the true ratings for these items. We propose a new pairwise regularization term to enrich the original loss. Experiments show that the new RS explainer is now efficient in both item ranking and item rating preservation and can still scale to larger datasets. Future work should evaluate other regularization terms, test other optimizers for the ranknet structure and other datasets.

REFERENCES

[1] Behnoush Abdollahi and Olfa Nasraoui. 2016. Explainable Matrix Factorization for Collaborative Filtering. In *Proc. of WWW Conf., Montreal, Canada, April 11-15, 2016.* 5–6. https://doi.org/10.1145/2872518.2889405

[2] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE TKDE* 17, 6 (2005), 734–749.

[3] Vito Walter Anelli, Alejandro Bellogín, Tommaso Di Noia, Francesco Maria Donini, Vincenzo Paparella, and Claudio Pomo. 2021. Adherence and Constancy in LIME-RS Explanations for Recommendation. In *ComplexRec workshop co-located with RecSys 2021, Amsterdam (CEUR Workshop Proceedings, Vol. 2960).*

[4] Léo Brunot, Nicolas Canovas, Alexandre Chanson, Nicolas Labroche, and Willeme Verdeaux. 2022. Preference-based and local post-hoc explanations for recommender systems. *Inf. Syst.* 108 (2022), 102021. https://doi.org/10.1016/j.is.2022.102021

[5] Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N. Hullender. 2005. Learning to rank using gradient descent. In *Proc. of ICML, Bonn, Germany, August 7-11, 2005.* 89–96. https://doi.org/10.1145/1102351.1102363

[6] Alexandre Chanson, Nicolas Labroche, and Willeme Verdeaux. 2021. Towards Local Post-hoc Recommender Systems Explanations. In *Proc. of DOLAP EDBT/ICDT Workshop, Nicosia, Cyprus, March 23, 2021.* 41–50.

[7] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep Neural Networks for YouTube Recommendations. In *Proc. of ACM RecSys Conf., New York, NY, USA, 2016.*

[8] Khalil Damak, Sami Khenissi, and Olfa Nasraoui. 2021. Debiased Explainable Pairwise Ranking from Implicit Feedback. In *Proc. of RecSys '21, Amsterdam, The Netherlands, September 27 - October 1, 2021.* 321–331. https://doi.org/10.1145/3460231.3474274

[9] Azin Ghazimatin, Oana Balalau, Rishiraj Saha Roy, and Gerhard Weikum. 2020. PRINCE: Provider-side Interpretability with Counterfactual Explanations in Recommender Systems. In *Proc. of ACM WSDM '20, Houston, TX, 2020.* 196–204. https://doi.org/10.1145/3336191.3371824

[10] Carlos A. Gomez-Uribe and Neil Hunt. 2016. The Netflix Recommender System: Algorithms, Business Value, and Innovation. 6, 4, Article 13 (dec 2016), 19 pages. https://doi.org/10.1145/2843948

[11] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4, Article 19 (dec 2015), 19 pages. https://doi.org/10.1145/2827872

[12] Alon Jacovi and Yoav Goldberg. 2021. Aligning Faithful Interpretations with their Social Attribution. *Trans. Assoc. Comput. Linguistics* 9 (2021), 294–310. https://doi.org/10.1162/tacl_a_00367

[13] Vassilis Kaffes, Dimitris Sacharidis, and Giorgos Giannopoulos. 2021. Model-Agnostic Counterfactual Explanations of Recommendations. In *Proc. of ACM UMAP 2021, Utrecht, The Netherlands, June, 21-25, 2021.* 280–285. https://doi.org/10.1145/3450613.3456846

[14] Alexandros Karatzoglou, Linas Baltrunas, and Yue Shi. 2013. Learning to rank for recommender systems. In *Proc. of ACM RecSys Conf., Hong Kong, China, October 12-16, 2013.* 493–494. https://doi.org/10.1145/2507157.2508063

[15] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* 7, 1 (2003), 76–80.

[16] Seán F. McLoone and George W. Irwin. 2001. Improving neural network training solutions using regularisation. *Neurocomputing* 37, 1-4 (2001), 71–90. https://doi.org/10.1016/S0925-2312(00)00314-3

[17] Caio Nóbrega and Leandro Balby Marinho. 2019. Towards explaining recommendations through local surrogate models. In *Proc. of ACM SAC Conf., Limassol, Cyprus, April 8-12, 2019.* 1671–1678. https://doi.org/10.1145/3297280.3297443

[18] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " Why should i trust you?" Explaining the predictions of any classifier. In *Proc. of ACM SIGKDD Conf., San Francisco, USA, June 26 - July 1, 2016.* 1135–1144.

[19] Benjamin Stark, Constanze Knahl, Mert Aydin, and Karim O. Elish. 2019. A Literature Review on Medicine Recommender Systems. *International Journal of Advanced Computer Science and Applications* 10, 8 (2019).

[20] Nava Tintarev and Judith Masthoff. 2015. Explaining Recommendations: Design and Evaluation. In *Recommender Systems Handbook.* 353–382. https://doi.org/10.1007/978-1-4899-7637-6_10

[21] Willème Verdeaux, Clément Moreau, Nicolas Labroche, and Patrick Marcel. 2020. Causality based explanations in multi-stakeholder recommendations. In *Proc. of the Workshops of the EDBT/ICDT 2020 Joint Conference.* Copenhagen, Denmark.

[22] Markus Weimer, Alexandros Karatzoglou, Quoc Le, and Alex Smola. 2007. COFI RANK - Maximum Margin Matrix Factorization for Collaborative Ranking. In *Adv. in Neural Information Proc. Syst.*, J. Platt, D. Koller, Y. Singer, and S. Roweis (Eds.), Vol. 20. Curran Associates, Inc.

[23] Yongfeng Zhang and Xu Chen. 2020. Explainable Recommendation: A Survey and New Perspectives. *Found. Trends Inf. Retr.* 14, 1 (2020), 1–101. https://doi.org/10.1561/1500000066

[24] Jinfeng Zhong and Elsa Negre. 2022. Towards improving user-recommender systems interactions. In *IEEE/SICE Int. Symposium on System Integration, SII 2022, Narvik, Norway, January 9-12, 2022.* 816–820. https://doi.org/10.1109/SII52469.2022.9708869