Explainable Recommendation with Comparative Constraints on Product Aspects

Trung-Hoang Le
Singapore Management University
Singapore
thle.2017@smu.edu.sg

Hady W. Lauw
Singapore Management University
Singapore
hadywlauw@smu.edu.sg

ABSTRACT

To aid users in choice-making, explainable recommendation models seek to provide not only accurate recommendations but also accompanying explanations that help to make sense of those recommendations. Most of the previous approaches rely on evaluative explanations, assessing the quality of an individual item along some aspects of interest to the user. In this work, we are interested in comparative explanations, the less studied problem of assessing a recommended item in comparison to another reference item.

In particular, we propose to anchor reference items on the previously adopted items in a user's history. Not only do we aim at providing comparative explanations involving such items, but we also formulate comparative constraints involving aspect-level comparisons between the target item and the reference items. The framework allows us to incorporate these constraints and integrate them with recommendation objectives involving both types of subjective and objective aspect-level quality assumptions. Experiments on public datasets of several product categories showcase the efficacies of our methodology as compared to baselines at attaining better recommendation accuracies and intuitive explanations.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; **Collaborative filtering**.

KEYWORDS

Explainable Recommendation; Comparative Constraints

ACM Reference Format:
Trung-Hoang Le and Hady W. Lauw. 2021. Explainable Recommendation with Comparative Constraints on Product Aspects. In *Proceedings of the Fourteenth ACM International Conference on Web Search and Data Mining (WSDM '21), March 8–12, 2021, Virtual Event, Israel.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3437963.3441754

1 INTRODUCTION

In this digital era, most of our activities and interactions are taking place online. This trend is accelerating further with recent events. The online marketplaces offer us choices that are orders-of-magnitude greater, necessitating the increasing use of recommender systems to help us navigate these choices. There is almost nary a site online that does not offer recommendation feature to its users.

To many, searching for products and making choices are often learning experiences in their own right. Many of the products we encounter in the search process are new to us. Therefore, while recommendations may help to focus our attention and narrow our search, these recommendations may not always immediately make sense to us. This is where explanations would go a long way in persuading users to understand and accept the recommendations.

Propitiously, in recent past, we begin to see a build-up of interest in explainable recommendations [34, 38]. The core of many models lies in anchoring the explanations on product aspects that have been mentioned by users in online reviews. For instance, a pioneering work EFM [38] produces an explanation in the form of *"You might be interested in [aspect], on which this product performs [well/poorly]."*. In turn, another well-known model MTER [34] produces an explanation in the form of *"Its [aspect] is [opinion phrase]."*. In these explanation templates, variables enclosed in square brackets are to be substituted with the relevant aspects, sentiments, or opinion phrases. Note how such explanations are *evaluative* by nature, assessing the quality of a single product in and of itself.

Comparative Explanation. We posit that users are inherently interested in choice-making, gaining information from relative comparisons. To this extent, binary sentiments are of insufficient precision in differentiating items (many of which may be equally *'positive'*, or *'negative'*). Neither is it easy to compare opinion phrases such as *'light'* vs. *'portable'*, or *'affordable'* vs. *'value for money'*. Thus, we seek a comparative explanation for a recommended item, with respect to another reference item, as illustrated below.

```
[recommended item] is better at [an aspect] than
[reference item], but worse at [another aspect].
```

One question is which items should serve as reference to a recommended item. There are several reasonable options. One could be a comparable substitute under consideration, e.g., a buyer of washing machines may wish to know how other washers in the market compare to the recommended one. Another could be a previously purchased item by the target user. Our focus is on the latter. For one reason, this is a comparison that the target user would likely find *understandable*, given her familiarity with the previous item. For another, the user may find it more *actionable* if it confers a perception of gain in improving upon one's past purchase. This would also characteristically be a *personalized* form of explanation,

WSDM '21, March 8–12, 2021, Virtual Event, Israel

ACM ISBN 978-1-4503-8297-7/21/03…$15.00
https://doi.org/10.1145/3437963.3441754

as it relates directly to the target user's past actions. The disadvantage lies in the classic cold-start scenario when the user has not previously purchased a similar product, in which case we could always fall back to an evaluative explanation for such scenarios.

Comparative Constraints. If we presuppose that a user generally tries to make better decisions over time by improving upon past purchases, then hypothetically the stereotypical users may already exhibit this behavior in their past purchase histories, at least to a certain extent. In other words, an explainable recommendation model that expects this behavior when learning the model has the potential of producing recommendations that are more reflective of user behavior and hopefully more accurate as well.

Therefore, we formulate comparative constraints relating historical purchases that can be incorporated into explainable recommendation models. Borrowing a terminology from the skyline literature [2], we say that a product y *dominates* another product x, if the former is at least as good as the latter in all aspects and better in at least one aspect. Now, it may not necessarily be the case that a later purchase y must always dominate a previous purchase x. On the other hand, it may be reasonable to assume that most of the time, y is not dominated by x. For example, while a user may go on to buy a cheaper model after finding that she does not need all the bells and whistles that come with a previously purchased more expensive model, it bears pointing out that the very fact that the later purchase is cheaper means that it is still superior in one way (i.e., value) and thus is not dominated by the earlier purchase.

In the explainable recommendation literature [34, 38], product aspects are often extracted from review text. When a later adoption is observed to be better at some aspect than a previous adoption by the same user, we formulate an aspect-level comparative constraint that seeks to preserve the same comparison in the modeled latent aspect sentiments. Moreover, our approach is to model these aspect-level comparative constraints as a framework, allowing specific instantiations built on two lines of explainable recommendation models, namely one that allows *subjective* aspect-level quality (user-specific) and another that accommodates *objective* aspect-level quality.

Contributions. First, we propose user's history of adoptions as basis for comparison for recommended items. Second, in Section 4, we describe a framework called *Comparative Explainable Recommendation* or ComparER, which incorporates comparative constraints into explainable recommendation models. Third, in Section 6, we conduct experiments on publicly available datasets of several product categories that show the efficacy of comparative constraints for top-k recommendations. In Section 7, we illustrate the comparative explanations through case study and user study.

2 PRODUCT RATINGS OVER TIME

To gain insights in developing the comparative constraints using previous items as references, we conduct an empirical analysis of ratings that users give to products from a broad spectrum of categories over time. For this purpose, we use the public[1] Amazon.com dataset [9]. In this dataset, only the act of rating, rather than purchase, is visible. Note also that this discussion is of motivational rather than conclusive nature, and a fuller investigation of this hypothesis at aspect and category-levels will be done in Section 6.

[1]http://jmcauley.ucsd.edu/data/amazon/

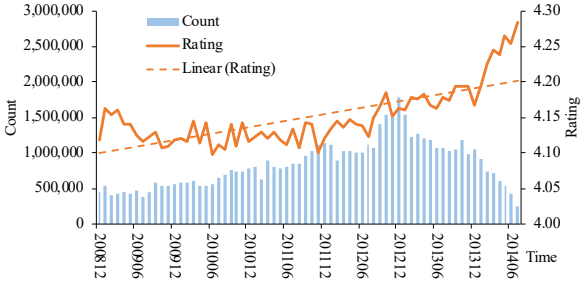

Figure 1: Average rating of products launched over time

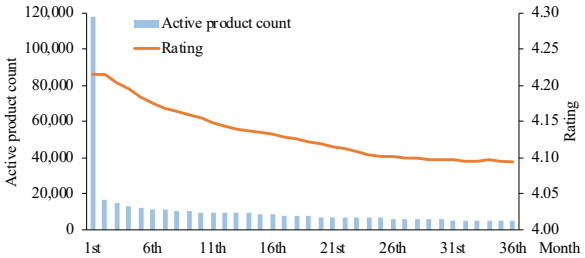

Figure 2: Average rating of products since launch time

Different Products Launched Across Time. First, we consider whether more recently launched products tend to induce greater 'consumer satisfaction', which may imply that generally speaking products tend to improve over time (with better features, ease of use, etc.). For this analysis, we associate every product with two attributes. One is its 'estimated quality' (i.e., average rating). The other is its 'estimated launch time' (i.e., time of its first review). These are but approximations, which may suffice as our intent is to study general trends involving many products.

We group all products 'launched' in the same month, and tracks their average 'quality' over time in Figure 1. It is evident from the line graph that, minor fluctuations notwithstanding, the general trend is that products launched later tend to have higher average rating over its 'lifetime'. The dotted line provides the best-fitting line, which has a positive gradient. The basis for this analysis is a large number of reviews, as shown by the histogram in Figure 1. The rating count initially goes up, probably as the popularity of Amazon goes up. The later downturn is because products 'launched' in recent years have not reached their full potentials in terms of rating counts by the cut-off date in the dataset. Even the lowest bar of the histogram (July 2014) has been supported by 258K ratings.

Same Product Over Time. Second, we now group all the products together, but slice the set of ratings by the distance between the launch time and the time in which the rating is assigned. This may give us a sense of how the quality of a product is generally perceived over time. The line graph in Figure 2 shows that the tendency is for the average rating to be the highest at launch and thereafter to fall over time. There could be various explanations, such as excitement about the product winds down, product flaws are discovered over time, etc. However, taking Figure 2 and Figure 1 together may suggest that satisfaction with earlier launched products decreases as other newly launched products appear.

Table 1: Main Notations

$\mathcal{U}$, $\mathcal{P}$, $\mathcal{A}$, $\mathcal{O}$	set of all users, products, aspects, and opinions
$\mathcal{L}$	sentiment lexicon
$\mathcal{S}$	set of all purchased sequences
$S_i \in \mathcal{S}$	a purchased sequence by user i
N	the highest overall rating in the target domain
Q	user-product-aspect quality tensor
X, Q'	user-aspect attention matrix and product-aspect quality matrix
λ_x, λ_y	coefficients weigh the relative important of aspects vs. ratings
σ	logistic function
λ_d	trade-off parameter of ComparER
α	trade-off between rating and aspect scores for ranking

Figure 2 also shows a histogram of products still 'actively' receiving ratings months from their launch. For many products, their 'lifetime' are rather short, as many are no longer active after one month. A cynical view is ratings decrease because inactive products are 'better' than long-surviving products. What we find more persuasive is that those products are inactive because they have been replaced by newer products, while others still survive as alternatives to the newer products but suffer from weaker perception.

Observations from these empirical analyses are consistent, at least not in conflict, with our hypothesis of a previously purchased product as reference. Capitalizing on these insights, we instantiate concrete formulations to build explainable recommendation models that support comparative explanations at aspect level.

3 NOTATION AND FORMULATION

The notations are summarized in Table 1. $\mathcal{P}$ denotes the universal set of products of a specific category, e.g., washing machines. The set of users is denoted $\mathcal{U}$. A user $i \in \mathcal{U}$ assigns to a product $j \in \mathcal{P}$ a rating $r_{ij} \in \mathbb{R}_+$. Each user is also associated with S_i, which is a temporally ordered list of products rated/adopted by user i.

Let $\mathcal{A}$ be the set of aspects, and $\mathcal{O}$ be the set of opinion phrases. In the review accompanying a rating r_{ij}, the user may express several opinion phrases with regards to aspects. From such expressions, we extract (a, o, ρ) tuples, each representing sentiment polarity $\rho \in \{-1, +1\}$ for aspect $a \in \mathcal{A}$ with opinion phrase $o \in \mathcal{O}$. A sentence may support a tuple. Tuples across sentences within a review are to be aggregated to get user's aspect-level sentiments (see Section 4). The collection of unique tuples extracted from reviews make up a contextual sentiment lexicon $\mathcal{L}$. To build $\mathcal{L}$, we leverage opinion lexicon from [12] and aspect lexicon from Microsoft Concept Graph[2] (see Section 6).

The problem can thus be stated as follows. We receive as input the set of users $\mathcal{U}$, products $\mathcal{P}$, ratings $\mathcal{R}$, sequences $\mathcal{S}$, and contextual sentiment lexicon $\mathcal{L}$. From these inputs, we seek a model that is capable of producing top-k personalized ranking list of products as well as an explanation associated with each recommended item. The explanation will express the tradeoff in aspects between the recommended item and a reference item (previously purchased).

4 METHODOLOGY

We propose *Comparative Explainable Recommendation* (ComparER). The gist is to transform observed aspect-level quality into a set of comparative constraints relating an item and previous items in the user's adoption history. In particular, we describe two variants of this approach owing to the two modes of expressing aspect-level quality common to the explainable recommendation literature. In one mode, aspect-level quality is subjective, i.e., the perception of a product for an aspect may vary across users. In another mode, aspect-level quality is objective. It is expressed for each product.

4.1 Subjective Aspect-Level Quality

Aspect-Level Quality. Let Q be a tensor of dimensionality $|\mathcal{U}| \times |\mathcal{P}| \times |\mathcal{A}|$. $q_{ijk} \in Q$ represents the quality score of aspect $k \in \mathcal{A}$ that user $i \in \mathcal{U}$ assigns to product $j \in \mathcal{P}$. However, the explicit quality scores given by users are usually unavailable. Instead, they can be estimated based on sentiments extracted from textual reviews [34].

Let s_{ijk} be the aggregate (e.g., sum) of the sentiment polarity scores (the aforementioned ρ) for aspect k extracted from user i's review of item j. The higher it is, the more positive i assesses j on k. For instance, [34] defines a non-linear mapping from s_{ijk} to q_{ijk}.

$$q_{ijk} = \begin{cases} 0, \text{ if aspect } k \text{ is not mentioned when } i \text{ reviews } j \\ 1 + \frac{N-1}{1+e^{-s_{ijk}}}, \text{ otherwise} \end{cases} \quad (1)$$

where N is the highest rating score in the target domain. Realistically, Q is only partially observed. The crux of the model lies in predicting the missing values in Q.

Comparative Constraint. Let us take two products rated by user i, namely j and j' where $j \prec j'$, i.e., j is earlier in the sequence of adoption than j'. We hypothesize that for many such pairs, j' is not dominated by j. In other words, $\forall k \in \mathcal{A}, q_{ijk} \leq q_{ij'k}$ or $\exists k \in \mathcal{A}, q_{ijk} < q_{ij'k}$. Note that this is not necessarily true all the time, we study such 'violations' in Section 6. However, it holds frequently enough that we would like to impose a constraint to their corresponding predictions $\hat{q}_{ijk}$ and $\hat{q}_{ij'k}$ (to be learnt by the prediction model) to preserve instances where $q_{ijk} < q_{ij'k}$ holds.

To this end, we favor the aspect quality comparisons where the more recently adopted product achieves superiority in some aspect. In particular, we formulate the following loss for comparative aspects, where we try to maximize the difference in the aspect quality scores between the more recent product and the earlier product.

$$L_{\text{ComparER}_{sub}} = -\sum_{i \in \mathcal{U}} \sum_{(j \prec j') \in S_i} \sum_{\{k | q_{ijk} < q_{ij'k}\}} \ln \sigma(\hat{q}_{ij'k} - \hat{q}_{ijk}) \quad (2)$$

Joint Model. We seek to minimize our proposed comparative constraint loss jointly with the recommendation objective. Without loss of generality, we adopt the recommendation objective of MTER [34]. It models user-product-aspect interactions with ratings jointly as a tensor $G \in \mathbb{R}_+^{|\mathcal{U}| \times |\mathcal{P}| \times (|\mathcal{A}|+1)}$. The rating r_{ij} is appended as an additional aspect to the tensor G, i.e., $g_{ij(|\mathcal{A}|+1)} = r_{ij}$. And the aspect-level quality scores are $g_{ij(:|\mathcal{A}|)} = q_{ij(:|\mathcal{A}|)}$.

G is decomposed by minimizing the following loss function[3]:

$$L_{\text{MTER}} = ||\hat{G} - G|| - \lambda_b \sum_{i \in \mathcal{U}} \sum_{\{(i,j,l) | r_{ij} > r_{il}\}} \ln \sigma(\hat{g}_{ij(|\mathcal{A}|+1)} - \hat{g}_{il(|\mathcal{A}|+1)}) \quad (3)$$

[2] https://concept.research.microsoft.com/

[3] The full objective also includes decomposition of user-aspect-opinion and item-aspect-opinion tensors. For simplicity, we only show the decomposition of user-item-aspect tensor G as the other tensors are for predicting opinion phrases. In our experiment, we use the full version of the objective function.

The first component $||\hat{G} - G||$ is due to Tucker decomposition [15] on the observed elements of the tensor, where $\hat{G}$ is the tensor reconstruction of G. The second component is due to applying the Bayesian Personalized Ranking (BPR) principle [29] to the rating component of the tensor to preserve the triples (i, j, l) where we observe the rating r_{ij} to be higher than r_{il}. λ_b is a trade off parameter to balance the two types of loss. Towards joint modeling, we integrate the loss due to the comparative constraints as follows:

$$L = L_{\text{MTER}} + \lambda_d L_{\text{ComparER}_{sub}} \tag{4}$$

where λ_d controls the contribution of comparative constraints.

Parameter Learning. Let Θ be the set of all learning parameters[4]. We optimize for $L_{\text{ComparER}_{sub}}$ by minimizing the following $-\ln \sigma(\hat{q}_{ij'k} - \hat{q}_{ijk})$. The corresponding gradient is:

$$-\frac{\nabla}{\nabla\Theta} \ln \sigma(\hat{q}_{ij'k} - \hat{q}_{ijk}) \propto \frac{e^{\hat{q}_{ijk} - \hat{q}_{ij'k}}}{1 + e^{\hat{q}_{ijk} - \hat{q}_{ij'k}}} \frac{\nabla}{\nabla\Theta}(\hat{q}_{ij'k} - \hat{q}_{ijk}) \tag{5}$$

The complexity of enumerating comparable product pairs for each sequence is $O(|S_i|^2)$. Iterating through all aspects requires $O(|\mathcal{A}|)$. Thus, given the set of training sequences $\mathcal{S}_{\text{train}}$ with an average sequence length of $\bar{S}$, the overall complexity of ComparER on a training epoch is $O(|\mathcal{S}_{\text{train}}| \cdot |\bar{S}|^2 \cdot |\mathcal{A}|)$. Nevertheless, in practice, the average sequence length[5] and the number of aspects in each product are relatively small.

Top-k Recommendation. With the learnt parameters, we compute the ranking score for a recommended item j to user i as follows:

$$\text{RankingScore}_{ij} = \alpha \cdot \frac{\sum_{k \in C_{ij}} \hat{q}_{ijk}}{|C_{ij}|} + (1 - \alpha) \cdot \hat{r}_{ij} \tag{6}$$

Here, $\hat{r}_{ij} = \hat{g}_{ij(|\mathcal{A}|+1)}$ is the predicted rating for the item, which is weighted by $(1-\alpha)$. We also consider the effects of aspect sentiments. Let $|C_{ij}|$ be the specified number of top aspects to be considered in the prediction. Correspondingly, $C_{ij} \subseteq \mathcal{A}$ is the set of top aspects of interest by user i on item j in terms of highest $\hat{q}_{ijk}$. We then average these scores in C_{ij} and incorporate it with weight α. This combination is useful as we will investigate in Section 6.

Explanation. For each item j' in the top-k recommendation list, we provide an explanation with respect to a reference item from the user i's previous adoption history. The choice of which item j from the user's history is to be used as reference is left as an application device. Possible heuristics could be most recent, similar, substitutable, equally-priced, etc. It could also be user-specified. The former aspect k would be one where $\hat{q}_{ijk} < \hat{q}_{ij'k}$, whereas the latter aspect k' would be one where $\hat{q}_{ijk'} > \hat{q}_{ij'k'}$ (if any). When there are more than one choice of aspect, we select randomly, though other selection criteria could also apply (e.g., highest difference).

4.2 Objective Aspect-Level Quality

Aspect-Level Quality. The second mode of expressing aspect-level quality is through a quality matrix $Q' \in \mathbb{R}_+^{|\mathcal{P}| \times |\mathcal{A}|}$, which is user-independent as proposed by [38]. Each element $q'_{jk} \in Q'$ may be obtained from sentiments extracted from reviews as follows:

$$q'_{jk} = \begin{cases} 0, \text{ if aspect } k \text{ is not discussed in reviews of } j \\ 1 + \frac{N-1}{1+e^{-s'_{jk}}}, \text{ otherwise} \end{cases} \tag{7}$$

where s'_{jk} is the aggregate (e.g., sum) of sentiment-scores of aspect k across the reviews of product j (by any reviewer).

Comparative Constraint. Let us take two products, namely j and j' where $\exists i \in \mathcal{U}, (j \prec j') \in S_i$, i.e., j' is later in the sequence of adoption than j for at least one user. We would like to impose a constraint to their corresponding predictions $\hat{q'}_{jk}$ and $\hat{q'}_{j'k}$ (to be learnt by the prediction model) to preserve instances where $q'_{jk} < q'_{j'k}$ holds. However, not all such constraints would be equal. Some are supported by many more sequences (users) than others. Let $c_{jj'}$ be the count of users that support the $(j \prec j')$ sequence. Intuitively, the greater $c_{jj'}$ is, the more weight it should carry in the optimization objective. Thus we apply a scaling factor of $1 + \ln(c_{jj'})$, which satisfies this objective. In particular,

$$L_{\text{ComparER}_{obj}} = -\sum_{(j,j') \in \cup_{i \in \mathcal{U}} S_i} (1 + \ln(c_{jj'})) \sum_{\{k | q'_{jk} < q'_{j'k}\}} \ln \sigma(\hat{q'}_{j'k} - \hat{q'}_{jk}) \tag{8}$$

where q'_{jk} and $q'_{j'k}$ are the product aspect-level quality score of a previously bought product j and a later bought product j'.

Joint Model. To integrate the proposed comparative constraint with a compatible recommendation objective, we extend the recommendation objective of EFM [38]. In addition to product-aspect quality matrix, it models user-aspect attention matrix X by projecting the frequency t_{ik} of an aspect k mentioned by a user i.

$$x_{ik} = \begin{cases} 0, \text{ if aspect } k \text{ is not mentioned by } i \\ 1 + (N-1)\left(\frac{2}{1+e^{-t_{ik}}} - 1\right), \text{ otherwise} \end{cases} \tag{9}$$

X and Q' are reconstructed along with ratings R by multi-matrix factorization with shared factors, minimizing the following:

$$L_{\text{EFM}} = ||UP^T - R||^2 + \lambda_x ||\eta_1 \psi^T - X||^2 + \lambda_y ||\eta_2 \psi^T - Q'||^2 \tag{10}$$

where $U = [\eta_1 \phi_1]$ and $P = [\eta_2 \phi_2]$ are users' and products' latent factors respectively. Each is the concatenation of aspect-based factor (η_1, η_2) influenced by X, Q' and hidden factors (ϕ_1, ϕ_2) influenced by ratings. ψ are the latent factors of aspects. Coefficients λ_x and λ_y weigh the relative importance of aspects vs. ratings.

We integrate the loss functions as follows:

$$L = L_{\text{EFM}} + \lambda_d L_{\text{ComparER}_{obj}} \tag{11}$$

Parameter Learning. We optimize for $L_{\text{ComparER}_{obj}}$ where the corresponding gradient for a comparative pair (j, j') is:

$$\begin{aligned} &-(1 + \ln(c_{jj'})) \sum_{\{k | q'_{j'k} > q'_{jk}\}} \frac{\nabla}{\nabla\Theta} \ln \sigma(\hat{q'}_{j'k} - \hat{q'}_{jk}) \\ &\propto (1 + \ln(c_{jj'})) \sum_{\{k | q'_{j'k} > q'_{jk}\}} \frac{e^{\hat{q'}_{jk} - \hat{q'}_{j'k}}}{1 + e^{\hat{q'}_{jk} - \hat{q'}_{j'k}}} \frac{\nabla}{\nabla\Theta}(\hat{q'}_{j'k} - \hat{q'}_{jk}) \end{aligned} \tag{12}$$

Because aspect score comparisons are done at product level, instead of user level, the complexity is smaller than before: $O(|\mathcal{U}| \times |\bar{S}|^2)$ for computing the counts $c_{jj'}$ and $O(|\mathcal{P}|^2 \cdot |\mathcal{A}|)$ for parameter learning (in practice the number of compared pairs are much less than $|\mathcal{P}|^2$ due to data sparsity).

[4]For simplicity of presentation, we hide all regularization from the notation. In our experiments, we use L_2-norm for every factor as regularization.

[5]If efficiency is a concern for very long sequences, optimization strategies such as using windows or subsequences may be applicable.

Ranking Score. The ranking score is measured as follows:

$$\text{RankingScore}_{ij} = \alpha \cdot \frac{\sum_{k \in C_i} \hat{x}_{ik} \hat{q'}_{jk}}{|C_i|N} + (1 - \alpha) \cdot \hat{r}_{ij} \quad (13)$$

where α is the control factor, C_i of a specified size is the set of most-cared aspects of user u_i in terms of $\hat{x}_{ik}$ values. Other details such as top-k recommendations and explanation are similar to those described earlier in Section 4.1.

5 RELATED WORK

Explainable Recommendation. Other than EFM [38], MTER [34] mentioned earlier, other approaches to explainable recommendations include [1, 31, 38] based on matrix factorization, [5, 34] based on tensor factorization, [24, 36] combining matrix factorization with topic modeling. Others enhance explainable models using graphs [10] or trees [8]. In virtually all cases, explainability comes from pairing ratings with reviews, knowledge bases, etc.

[26] analyses the role of attributes in product quality comparisons. [4] studies a related form of comparative explanation, called tradeoff-oriented explanation, which is validated to be useful via a user study. However, their work is aimed at comparing product clusters. Moreover, they focus on validating the interface, rather than on the underlying explainable recommendation model. [33] studies multivariate rankings, how to rank multiple aspects jointly. Their focus is not on the explanation perspective that sets us apart.

Other than template-based explanation, there are alternative explanations. [11] works with content-based collaborative filtering. [23, 31, 35] explains recommendation by learning explainable rules. [6, 30] interprets from learnt topics, whereas [28] uses social network. [3, 24] suggests helpful reviews. Others attempt to provide recommendations along with the generated review. [18] conditions its review generation on latent factors, [22] extends on review textual features, whereas [7] conditions on aspects. These works are not comparable, as they focus on a different type of explanation.

Another line of research decouples the recommendation explanation from explainable recommendation, focusing only on generating explanation for a given recommendation [17, 27].

Comparison Mining. There are various problems related to comparisons. One is determining which of two products is better overall [19, 39]. For instance, it could be based on how two named entities are compared within the same sentence [32]. Another line is in finding substitute and/or complementary products. [25] relies on discovering topics in product reviews and networks of products derived from browsing and co-purchasing logs. Yet another related problem is competitor mining [13, 16, 37], finding which products are most likely to be comparable to a target product. Our work is orthogonal to these directions. Rather than focusing on the selection of which products to compare to, we assume they are given and focus on finding the appropriate recommendation explanation.

Skyline Queries. While we borrow the concept of when a product dominates another from the literature on skyline queries, the connection is incidental. Skyline computation identifies a set of points that are not dominated by any other point. It is frequently expressed as a query processing issue, optimizing for the computational efficiency in which such points can be retrieved [14, 20]. A similar concept is applied to recommendation [21], recommending a skyline group that are not dominated by any other group. It is not our interest to find such a skyline set. Instead, we focus on improving explainable recommendations using comparative constraints as well as on exploring comparative forms of explanations.

Table 2: Data Statistics

Dataset	#User	#Product	#Rating	#Aspect	#Opinion
Electronic	45,225	57,873	759,016	445	4,232
Toy	4,188	10,512	70,944	428	2,559
Clothing	5,200	17,895	68,262	422	1,748
Cellphone	3,216	7,807	44,492	423	2,032
Music	1,763	3,383	40,675	416	3,065

6 EXPERIMENT

As experimental objectives, we investigate whether incorporating the comparative constraints leads to improved recommendation accuracy. We also consider the resulting comparative explanations through case study and user study. Comparisons between methods are tested with one-tailed paired-sample Student's t-test at 0.05 level. Computational efficiency is not the focus of this paper. Most recommendation algorithms are learnt offline. While ranking score computation is online, its computational time is practically identical across methods being compared. Experiments were run on machine with Intel Xeon E5-2650v4 2.20 GHz CPU and 256GB RAM.

6.1 Setup

Datasets. For experiments, we rely on the publicly available Amazon datasets from the same source as the one used in Section 2 for preliminary empirical analysis. However, due to the comparative nature of the hypothesis, the modeling and learning are more appropriately conducted for distinct categories separately, as one probably does not compare a toy and a phone. Therefore, we conduct five experiments with the following categories respectively: *Electronics* (Electronic), *Toys and Games* (Toy), *Clothing* (Clothing), *Cell Phones and Accessories* (Cellphone), *Digital Music* (Music). Table 2 summarizes basic statistics of the datasets. For statistical sufficiency, we retain users with at least 10 ratings. Each user's rating sequence is then split into train, validation, test with ratio $0.64 : 0.16 : 0.2$ chronologically. Unknown products are excluded from validation and test sets, a uniform practice across all methods.

To extract aspects and opinions from reviews, we adopt the frequency-based approach of [8]. Using Microsoft Concepts as aspects, we retrieve top-2000 most frequently mentioned in reviews, sort them by their correlations with the ratings, and keep only top-500 (after filtering unseen aspects in validation/testing, the number comes to 400+). We select opinions associated with these aspects to construct (a, o, ρ) tuples, using the opinion lexicon from [12].

Methods and Baselines. There are two instantiations of our method, namely: ComparER$_{obj}$ (see Section 4.2) and ComparER$_{sub}$ (see Section 4.1). We note that while in Section 4 we describe joint models that incorporate comparative constraints with a base recommendation objective, our approach can be seen as a framework as these comparative constraints could potentially be applicable to other base recommendation objectives[6]. Therefore, the most appropriate choice of baselines would be the base recommendation objectives that we use, specifically EFM [38] for ComparER$_{obj}$ and

[6]To maintain focus, we keep such explorations to future work.

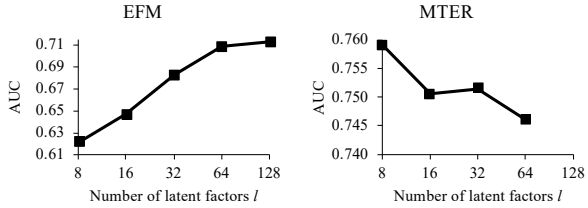

Figure 3: AUC performance of EFM and MTER while varying number of latent factors l on Electronic data

Table 3: Performance of EFM and ComparER$_{obj}$

Dataset	Model	AUC	Recall@k%			NDCG@k%		
			1	5	10	1	5	10
Electronic	EFM	0.717	0.107	0.277	0.393	0.022	0.044	0.057
	ComparER$_{obj}$	**0.759**§	**0.176**§	**0.362**§	**0.474**§	**0.038**§	**0.062**§	**0.074**§
Toy	EFM	0.580	0.030	0.106	0.189	0.009	0.021	0.033
	ComparER$_{obj}$	**0.656**§	**0.042**§	**0.157**§	**0.268**§	**0.014**§	**0.033**§	**0.049**§
Clothing	EFM	0.579	0.036	0.114	0.189	0.009	0.020	0.028
	ComparER$_{obj}$	**0.611**§	**0.059**§	**0.154**§	**0.233**§	**0.016**§	**0.030**§	**0.039**§
Cellphone	EFM	0.652	0.045	0.162	0.266	0.012	0.032	0.046
	ComparER$_{obj}$	**0.701**§	**0.069**§	**0.214**§	**0.334**§	**0.022**§	**0.046**§	**0.062**§
Music	EFM	0.641	0.040	0.158	0.257	0.019	0.046	0.064
	ComparER$_{obj}$	**0.678**§	**0.060**§	**0.200**§	**0.320**§	**0.029**§	**0.061**§	**0.083**§

§ denotes statistically significant improvements. Highest values are in **bold**

MTER [34] for ComparER$_{sub}$, for these would directly evaluate whether the comparative constraints produce a positive effect.

Measures. Each method produces a ranked list of recommended items. The length of each list is relative to corresponding dataset size, and is expressed as the top-k% of items in terms of the ranking score, for various $k \in \{1, 5, 10\}$. As evaluation measures, we employ multiple standard ranking metrics, such as Area Under the ROC Curve (AUC), Recall at k percentage (Recall@k%), and Normalized Discount Cumulative Gain at k percentage (NDCG@k%). For these metrics, a higher value indicates better performance.

Learning Details. We use grid search to find the optimal hyperparameters for the baselines EFM and MTER. For ComparER, we then apply the same hyperparameters as the coresponding base model for parity. We fix $\lambda_x = \lambda_y = 1$ (as in the author implementation in Librec), and search for the latent dimensions $l \in \{8, 16, 32, 64, 128\}$. We further tune the coefficient λ_d in the candidate set of $\{0.01, 0.1, 1, 10, 100\}$. For each method, the setting with the best AUC on validation set is selected.

Figure 3 illustrates the performance of the base models while varying the latent dimensionality l in terms of AUC on the largest Electronic data (we observe similar trends on other datasets as well). EFM achieves better AUC with greater dimensionality l. The opposite is true for MTER, yet it requires much more time for training. So we set $l = 128$ for EFM and $l = 8$ for MTER as default, which we apply to our methods as well. To speed up training, we load pretrained weights from the respective base model and continue training with the added constraints. For parity purpose, we further verify that as the base models have indeed converged, further continuing their training does not add any value.

6.2 Ranking Performance

First, we investigate whether adding the comparative constraints improve the ranking performance of the base models. Table 3 shows the results for ComparER$_{obj}$ and its baseline EFM. We observe that on all metrics, across all datasets, ComparER$_{obj}$ improves upon the ranking performance of EFM consistently and in a statistically significant manner. We attribute this to the contribution of the comparative constraints based on historical reference. In turn, Table 4 shows the ranking performance of ComparER$_{sub}$ and its baseline MTER. It substantially echoes the observations above that supports the outperformance of ComparER$_{sub}$ over its baseline. Much of the outperformance are also statistically significant, save for a couple of pockets (e.g., Recall@1% on the smaller datasets Clothing and Music) where the differences still exist but in a smaller way.

Table 4: Performance of MTER and ComparER$_{sub}$

Dataset	Model	AUC	Recall@k%			NDCG@k%		
			1	5	10	1	5	10
Electronic	MTER	0.759	0.157	0.337	0.448	0.035	0.058	0.070
	ComparER$_{sub}$	**0.797**§	**0.185**§	**0.398**§	**0.520**§	**0.041**§	**0.069**§	**0.083**§
Toy	MTER	0.727	0.066	0.217	0.359	0.020	0.044	0.064
	ComparER$_{sub}$	**0.747**§	**0.093**§	**0.278**§	**0.422**§	**0.029**§	**0.059**§	**0.079**§
Clothing	MTER	0.671	0.069	0.189	0.287	**0.017**	0.034	0.045
	ComparER$_{sub}$	**0.680**§	**0.071**	**0.200**§	**0.297**§	**0.017**	**0.035**§	**0.046**§
Cellphone	MTER	0.757	0.113	0.296	0.425	0.036	0.066	0.084
	ComparER$_{sub}$	**0.787**§	**0.129**§	**0.337**§	**0.474**§	**0.043**§	**0.077**§	**0.095**§
Music	MTER	0.844	0.128	0.380	0.548	0.057	0.114	0.146
	ComparER$_{sub}$	**0.848**§	**0.130**	**0.391**§	**0.564**§	**0.059**	**0.119**§	**0.151**§

§ denotes statistically significant improvements. Highest values are in **bold**

Table 5: Constraint violations analysis. Counting function $V(\cdot)$ takes all pairs and the aspect quality weights as input and reports number of pairs violating the constraint

Dataset	#Pairs	$V(Q')$	$\frac{V(Q')}{\#\text{Pairs}}$	$V(Q)$	$\frac{V(Q)}{\#\text{Pairs}}$
Electronic	4,692,596	57,668	1.23%	30,867	0.66%
Toy	479,786	11,602	2.42%	10,476	2.18%
Clothing	258,357	14,081	5.45%	13,226	5.11%
Cellphone	219,851	8,024	3.65%	5,751	2.61%
Music	509,302	6,515	1.28%	2,481	0.49%
Total	6,159,892	97,890	1.59%	62,801	1.02%

6.3 Comparative Constraints

Constraint Violation. Earlier, we motivate the comparative constraints with the hypothesis whereby it is unlikely that an item j' that a user rates later is 'dominated' by another item j rated earlier. Seeking some measure of validation, we now analyze the number of occurrences in which this hypothesis is violated. To define violation, we use the ground-truth matrix Q' for objective aspect-level quality. For subjective aspect-level quality, a matrix is obtained from flattening Q by averaging the values across users.

Table 5 shows the total number pairs involving two products, one of which is rated later than the other by a user. Suppose that $V(\cdot)$ is a counting function that takes in all these pairs and the aspect quality weights as input, and reports the number of violating pairs. We express these numbers both as absolute count as well as a percentage of the total number of pairs. Interestingly, the stated hypothesis seems to hold for the vast majority of pairs. The violations amount to a single-digit percentage value, which across all datasets come to less than 2% of all pairs.

Effect of Constraint Coefficient λ_d. We incorporate ComparER constraints with a coefficient weight λ_d to learn model parameters that would preserve the constraint for as many pairs

Table 6: Effect of Constraint Coefficient λ_d on ComparER$_{obj}$

Dataset	λ_d	AUC	Recall@k%			NDCG@k%			$V(\hat{Q}')$
			1	5	10	1	5	10	
Electronic	0.01	0.755	**0.182**	0.357	0.463	**0.042**	**0.065**	**0.077**	91,669
	0.1	**0.759**	0.176	**0.362**	**0.474**	0.038	0.062	0.074	75,528
	1	0.749	0.151	0.348	0.459	0.029	0.055	0.067	45,982
	10	0.735	0.103	0.296	0.412	0.021	0.046	0.059	48,401
Toy	0.01	0.647	**0.047**	**0.160**	0.259	**0.015**	**0.033**	0.047	10,772
	0.1	**0.656**	0.042	0.157	0.268	0.014	**0.033**	**0.049**	10,111
	1	0.649	0.033	0.156	**0.269**	0.011	0.032	0.047	9,919
	10	0.624	0.036	0.139	0.233	0.013	0.030	0.043	9,474
Clothing	0.01	0.606	0.053	0.151	0.229	0.015	0.028	0.038	14,053
	0.1	0.611	**0.059**	**0.154**	0.233	**0.016**	**0.030**	**0.039**	12,043
	1	**0.612**	0.054	0.153	**0.237**	0.015	0.028	0.038	12,294
	10	0.605	0.051	0.150	**0.237**	0.012	0.026	0.036	11,505
Cellphone	0.01	0.697	**0.072**	**0.218**	0.331	**0.023**	**0.047**	**0.062**	7,698
	0.1	**0.701**	0.069	0.214	**0.334**	0.022	0.046	**0.062**	6,739
	1	0.698	0.053	0.202	0.320	0.017	0.041	0.057	6,442
	10	0.689	0.044	0.169	0.294	0.014	0.034	0.051	5,598
Music	0.01	0.672	0.056	0.195	0.309	0.025	0.057	0.078	5,634
	0.1	**0.678**	**0.060**	**0.200**	**0.320**	**0.029**	**0.061**	**0.083**	5,074
	1	0.675	0.044	0.179	0.308	0.023	0.054	0.078	3,948
	10	0.648	0.036	0.142	0.248	0.016	0.041	0.060	3,875

Better values are in **bold**

Table 7: Effects of Constraint Coefficient λ_d on ComparER$_{sub}$

Dataset	λ_d	AUC	Recall@k%			NDCG@k%			$V(\hat{Q})$
			1	5	10	1	5	10	
Electronic	0.1	0.759	0.150	0.331	0.445	0.033	0.056	0.069	2,029,774
	1	0.774	0.163	0.356	0.476	0.036	0.061	0.074	2,138,013
	10	0.794	0.180	0.388	0.512	0.040	0.067	0.081	1,062,211
	100	**0.797**	**0.185**	**0.398**	**0.520**	**0.041**	**0.069**	**0.083**	814,568
Toy	0.1	0.725	0.065	0.214	0.357	0.019	0.043	0.063	246,307
	1	0.733	0.072	0.232	0.373	0.022	0.048	0.068	229,863
	10	**0.747**	**0.093**	**0.278**	**0.422**	**0.029**	**0.059**	**0.079**	106,607
	100	**0.747**	0.082	0.263	0.410	0.024	0.054	0.074	70,228
Clothing	0.1	0.666	0.069	0.187	0.287	**0.017**	0.034	**0.046**	131,862
	1	0.670	0.069	0.192	0.293	**0.017**	0.034	**0.046**	130,323
	10	**0.680**	**0.071**	**0.200**	**0.297**	**0.017**	**0.035**	**0.046**	82,484
	100	0.678	0.065	0.190	**0.297**	0.016	0.033	**0.046**	61,998
Cellphone	0.1	0.751	0.112	0.282	0.409	0.036	0.064	0.081	117,560
	1	0.762	0.129	0.313	0.432	0.042	0.072	0.089	109,877
	10	0.783	**0.156**	**0.359**	**0.477**	**0.053**	**0.086**	**0.102**	53,091
	100	**0.787**	0.129	0.337	0.474	0.043	0.077	0.095	22,119
Music	0.1	0.840	0.127	0.381	0.554	0.059	0.117	0.149	257,273
	1	0.844	0.129	0.386	0.560	**0.060**	0.118	**0.151**	249,079
	10	**0.848**	**0.130**	**0.391**	**0.564**	0.059	**0.119**	**0.151**	101,376
	100	0.833	0.126	0.367	0.531	0.058	0.113	0.144	124,574

Better values are in **bold**

as possible. Table 6 tabulates the ranking performance and violation count of ComparER$_{obj}$ at various values of λ_d. When λ_d is zero, we are optimizing only for the recommendation objective. Interestingly, as we increase λ_d, the number of violations (last column) generally decreases, which means the imposed constraints are taking effect. The ranking performance also initially improves, though with too high λ_d it may hurt ranking performance as it downweighs the recommendation objective. Table 7 presents the results for ComparER$_{sub}$ with largely the same conclusion as well.

To see how ComparER retains the original violations as in the base models, not only in terms of the violation counts, but also whether it is identifying the 'correct' violations, Table 8 and Table 9 show that ComparER achieves lower number of constraint violations than the baselines that do not optimize for this directly.

To further clarify the degree of agreement between the estimated scores obtained after training ($\hat{Q}'$ or $\hat{Q}$) and the ground-truth scores

Table 8: Constraint Violations: EFM vs. ComparER$_{obj}$

Dataset	Model	$V(\hat{Q}')$	Recall	Precision	F-Measure
Electronic	EFM	92,491	0.810	0.505	0.622
	ComparER$_{obj}$	75,528	**0.875**	**0.668**	**0.758**
Toy	EFM	17,799	**0.888**	0.579	0.701
	ComparER$_{obj}$	10,111	0.857	**0.983**	**0.916**
Clothing	EFM	21,718	0.870	0.564	0.684
	ComparER$_{obj}$	12,294	**0.872**	**0.999**	**0.931**
Cellphone	EFM	11,523	**0.878**	0.611	0.721
	ComparER$_{obj}$	6,739	0.825	**0.982**	**0.897**
Music	EFM	10,090	**0.838**	0.541	0.658
	ComparER$_{obj}$	5,074	0.750	**0.962**	**0.843**

Better values are in **bold**

Table 9: Constraint Violations: MTER vs. ComparER$_{sub}$

Dataset	Model	$V(\hat{Q})$	Recall	Precision	F-Measure
Electronic	MTER	2,033,981	**0.976**	0.015	0.029
	ComparER$_{sub}$	814,568	0.867	**0.033**	**0.063**
Toy	MTER	229,456	**0.971**	0.044	0.085
	ComparER$_{sub}$	106,607	0.892	**0.088**	**0.160**
Clothing	MTER	123,797	**0.944**	0.101	0.182
	ComparER$_{sub}$	82,484	0.877	**0.141**	**0.242**
Cellphone	MTER	99,014	**0.946**	0.055	0.104
	ComparER$_{sub}$	22,119	0.648	**0.168**	**0.267**
Music	MTER	198,437	**0.800**	0.010	0.020
	ComparER$_{sub}$	101,376	0.541	**0.013**	**0.026**

Better values are in **bold**

Table 10: Aspects in Ranking Score: α and #Top Aspects

Dataset	ComparER$_{obj}$		ComparER$_{sub}$	
	α	#Top Aspects	α	#Top Aspects
Electronic	0.8	300	0.4	200
Toy	0.7	20	0.5	all aspects
Clothing	0.8	10	0.3	10
Cellphone	0.6	10	0.3	300
Music	0.7	20	0.3	400

in training data (Q' or Q), we evaluate the Recall = $\frac{V(Q')\cap V(\hat{Q}')}{V(Q')}$, Precision = $\frac{V(Q')\cap V(\hat{Q}')}{V(\hat{Q}')}$, F-Measure (similar formula applies to Q).

Given the large number constraint violations by the base models, perhaps it is not surprising that they have higher recall. However, much of the recovered violations may not be correct, as reflected by their lower precision as compared to ComparER. When we take the recall and precision together, their harmonic mean or F-measure shows that ComparER performs better in recovering the violations.

6.4 Incorporating Aspects in Ranking Scores

To verify that the aspects do participate meaningfully in the recommendation, we tune different values α in the range of $[0, 1]$ with step size 0.1 and the number of top aspects in a candidate set of $\{10, 20, 30, 50, 100, 200, 300, 400,$ all aspects$\}$. Table 10 shows the setting with the best performance on various datasets for ComparER$_{sub}$ and ComparER$_{obj}$. Evidently, for all datasets, we have $\alpha > 0$, which means that aspects are indeed helpful. The number of aspects to take into account in the prediction is dataset- and method-dependent.

7 COMPARATIVE EXPLANATION

One of the objectives is to explain a recommended item by way of comparison to a reference item (another item previously rated or purchased by the user). In this section, we show a couple of examples of such explanations and discuss a user study.

User: A15N56ZCTHRB73
Recommended product: B00F6AVFK8
The Oontz XL - Cambridge SoundWorks Most Powerful Portable, Wireless, Bluetooth Speaker

Previously bought product: B00AI5V3CQ
The OontZ Angle Ultra Portable Wireless Bluetooth Speaker - Better Sound, Better Volume, Incredible Online Price - The Perfect Speaker to take everywhere with you this summer (Blue)

Explanation:
EFM: You might be interested in **sound**, on which this product performs well.
You might be interested in **purpose**, on which this product performs poorly.
ComparER$_{obj}$: Product B00F6AVFK8 is better at **quality** than B00AI5V3CQ. But worse at **sound**.

Figure 4: Example Explanations by EFM and ComparER$_{obj}$

User: ACO3U8DT64IV6
Recommended product: B00GN6QZ0Y
Mpow 3.1Amps 15.5W Dual Port Backlight USB Car Charger for iPhone 5s 5c 5 4s 4 iPad 1 2 3 5 Air Mini Samsung Galaxy S4 S3 S2 Galaxy Note 3 2 HTC One X V S and More (White and Blue)

Previously bought product: B000S5Q9CA
Motorola Vehicle Power Adapter micro-USB Rapid Rate Charger

Explanation:
MTER: Its **phone** is mistakenly. Its **case** is mistakenly.
ComparER$_{sub}$: Product B00GN6QZ0Y is better at **design** than B000S5Q9CA. But worse at **quality**.

Figure 5: Example Explanations by MTER and ComparER$_{sub}$

7.1 Case Study

For the first example in Figure 4, we recommend a bluetooth speaker *Oontz XL* to the user. The explanation generated by the baseline EFM for this product is evaluative, speaking of the aspects *sound* which is positive and *purpose* which is negative. It offers no hint as for how this product may compare to any other. A comparative explanation relies on a reference item, which we propose to be a previously rated product. In this particular case, one of the previously rated products in the category was another bluetooth speaker *Oontz Angle*, which is a smaller model than the recommended item. Using our approach ComparER$_{obj}$, we identify *quality* as an aspect for which the recommended item is better, and *sound* as an aspect for which it is worse than the reference item. Since the user would have been familiar with the reference item (previously rated), this may offer more information than a standalone explanation.

For a second example involving ComparER$_{sub}$ and its baseline MTER, Figure 5 shows the case of a user being recommended a car charger of *Mpow* brand. MTER's explanation is based on opinion phrases. In this case, it identifies two pertinent aspects: *phone* and *case* and the opinion phrase *mistakenly*. In contrast, our approach is to present a reference item, which is a previously purchased car charger of *Motorola* brand. The explanation by ComparER$_{sub}$ alludes to the recommended item being better at *design* (it is more compact and cableless) but worse at *quality* (it is of a less well-known brand than the reference item).

Table 11: Analysis of User Study

+ reference product	Method	Score	Method	Score
No	EFM (original)	2.12	MTER (original)	2.06
Yes	EFM (enhanced)	2.24	MTER (enhanced)	2.05
Yes	ComparER$_{obj}$	**3.29**§	ComparER$_{sub}$	**3.06**§

$^{\S}p$-value < 0.01. Highest value are in **bold**

7.2 User Study

We conduct a user study with 25 examples (5 product recommendations from each category). Since the focus in this section is on the explanation, rather than the relative accuracy of methods, we consider recommended items from users' test data, with reference items from their training data. We generate explanation for the recommended product by ComparER and its base model for every example. As seen in the case studies, the original versions of EFM and MTER generate explanation for only the recommended product. To give them the benefit of comparison, we further include enhanced versions of these baselines by showing explanations for the reference products as well, as in our approach.

We then conduct independent surveys, each containing 25 examples of different recommendation explanations generated from different models (selected randomly and presented blindly). The study was completed by 20 annotators, who are neither the authors nor having any knowledge of the objective of the study. We ask every annotator to rate their opinion on the generated explanation with the following question (adopted from [34]):

> *Does the explanation help you know more about the recommended product?*

Each explanation is seen by at least 3 different people. A participant chooses from five-point Likert scale, from 1 (strongly disagree) to 5 (strongly agree). The average scores are reported in Table 11. Both ComparER variants received significantly better scores than the original and the enhanced versions of the base methods.

While we are aware of general limitations of user studies, we presume that similar limitations apply to both our approach and the baselines. The consistency in which users find favor with the proposed explanations provide some evidence for the promising nature of ComparER at providing comparative explanations.

8 CONCLUSION

We approach explainable recommendation from the perspective where an explanation compares the recommended item with a reference item (previously adopted product). The proposed ComparER incorporates comparative constraints into explainable recommendation models. Experiments on datasets of five categories show that ComparER enhances the performance of ranking prediction.

There are several avenues for future work to build on this promising line of research: exploring comparisons to other definitions of reference items such as substitutes, incorporating the comparative constraints into other recommendation models/objectives, testing multi-way comparisons to multiple products simultaneously, etc.

ACKNOWLEDGMENTS

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its NRF Fellowship Programme (Award No. NRF-NRFF2016-07).

REFERENCES

[1] Konstantin Bauman, Bing Liu, and Alexander Tuzhilin. 2017. Aspect Based Recommendations: Recommending Items with the Most Valuable Aspects Based on User Reviews. In *KDD (KDD'17)*. ACM, 717–725. https://doi.org/10.1145/3097983.3098170

[2] Stephan Borzsony, Donald Kossmann, and Konrad Stocker. 2001. The Skyline operator. In *ICDE*. IEEE, 421–430.

[3] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural Attentional Rating Regression with Review-Level Explanations. In *WWW (WWW'18)*. 1583–1592. https://doi.org/10.1145/3178876.3186070

[4] Li Chen and Feng Wang. 2017. Explaining Recommendations Based on Feature Sentiments in Product Reviews. In *IUI (IUI'17)*. ACM, 17–28. https://doi.org/10.1145/3025171.3025173

[5] Xu Chen, Zheng Qin, Yongfeng Zhang, and Tao Xu. 2016. Learning to Rank Features for Recommendation over Multiple Categories. In *SIGIR (SIGIR'16)*. ACM, 305–314. https://doi.org/10.1145/2911451.2911549

[6] Xu Chen, Yongfeng Zhang, and Zheng Qin. 2019. Dynamic Explainable Recommendation Based on Neural Attentive Models. In *AAAI*. AAAI Press, 53–60. https://doi.org/10.1609/aaai.v33i01.330153

[7] Zhongxia Chen, Xiting Wang, Xing Xie, Tong Wu, Guoqing Bu, Yining Wang, and Enhong Chen. 2019. Co-Attentive Multi-Task Learning for Explainable Recommendation. In *IJCAI*, Sarit Kraus (Ed.). ijcai.org, 2137–2143. https://doi.org/10.24963/ijcai.2019/296

[8] Jingyue Gao, Xiting Wang, Yasha Wang, and Xing Xie. 2019. Explainable Recommendation through Attentive Multi-View Learning. In *AAAI*. AAAI Press, 3622–3629. https://doi.org/10.1609/aaai.v33i01.33013622

[9] Ruining He and Julian McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering *(WWW'16)*. 507–517. https://doi.org/10.1145/2872427.2883037

[10] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. 2015. TriRank: Review-Aware Explainable Recommendation by Modeling Aspects. In *CIKM (CIKM'15)*. ACM, 1661–1670. https://doi.org/10.1145/2806416.2806504

[11] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. 2004. Evaluating Collaborative Filtering Recommender Systems. *ACM Trans. Inf. Syst.* 22, 1 (Jan. 2004), 5–53. https://doi.org/10.1145/963770.963772

[12] Minqing Hu and Bing Liu. 2004. Mining and Summarizing Customer Reviews. In *KDD (KDD'04)*. ACM, 168–177. https://doi.org/10.1145/1014052.1014073

[13] Myungha Jang, Jin-woo Park, and Seung-won Hwang. 2012. Predictive mining of comparable entities from the web. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

[14] Kazuki Kodama, Yuichi Iijima, Xi Guo, and Yoshiharu Ishikawa. 2009. Skyline Queries Based on User //locations and Preferences for Making //location-Based Recommendations. In *LBSN (LBSN'09)*. ACM, 9–16. https://doi.org/10.1145/1629890.1629893

[15] Tamara G Kolda and Brett W Bader. 2009. Tensor decompositions and applications. *SIAM review* 51, 3 (2009), 455–500.

[16] Theodoros Lappas, George Valkanas, and Dimitrios Gunopulos. 2012. Efficient and domain-invariant competitor mining. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 408–416.

[17] Trung-Hoang Le and Hady W. Lauw. 2020. Synthesizing Aspect-Driven Recommendation Explanations from Reviews. In *IJCAI (IJCAI'20)*. ijcai.org, 2427–2434. https://doi.org/10.24963/ijcai.2020/336

[18] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. Neural Rating Regression with Abstractive Tips Generation for Recommendation. In *SIGIR (SIGIR'17)*. ACM, 345–354. https://doi.org/10.1145/3077136.3080822

[19] Si Li, Zheng-Jun Zha, Zhaoyan Ming, Meng Wang, Tat-Seng Chua, Jun Guo, and Weiran Xu. 2011. Product Comparison Using Comparative Relations. In *SIGIR (SIGIR'11)*. ACM, 1151–1152. https://doi.org/10.1145/2009916.2010094

[20] Jinfei Liu, Li Xiong, Jian Pei, Jun Luo, and Haoyu Zhang. 2015. Finding Pareto Optimal Groups: Group-Based Skyline. *Proc. VLDB Endow.* 8, 13 (Sept. 2015), 2086–2097. https://doi.org/10.14778/2831360.2831363

[21] Jinfei Liu, Li Xiong, Jian Pei, Jun Luo, Haoyu Zhang, and Si Zhang. 2019. SkyRec: Finding Pareto Optimal Groups. In *CIKM (CIKM'19)*. ACM, 2913–2916. https://doi.org/10.1145/3357384.3357838

[22] Yichao Lu, Ruihai Dong, and Barry Smyth. 2018. Why I like It: Multi-Task Learning for Recommendation and Explanation. In *RecSys (RecSys'18)*. ACM, 4–12. https://doi.org/10.1145/3240323.3240365

[23] Weizhi Ma, Min Zhang, Yue Cao, Woojeong Jin, Chenyang Wang, Yiqun Liu, Shaoping Ma, and Xiang Ren. 2019. Jointly Learning Explainable Rules for Recommendation with Knowledge Graph. In *WWW (WWW'19)*. ACM, 1210–1221. https://doi.org/10.1145/3308558.3313607

[24] Julian McAuley and Jure Leskovec. 2013. Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text. In *RecSys (RecSys'13)*. ACM, 165–172. https://doi.org/10.1145/2507157.2507163

[25] Julian McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring Networks of Substitutable and Complementary Products. In *KDD (KDD'15)*. ACM, 785–794. https://doi.org/10.1145/2783258.2783381

[26] Felipe Moraes, Jie Yang, Rongting Zhang, and Vanessa Murdock. 2020. The Role of Attributes in Product Quality Comparisons. In *CHIIR (CHIIR '20)*. ACM, 253âĂŞ262. https://doi.org/10.1145/3343413.3377956

[27] Georgina Peake and Jun Wang. 2018. Explanation Mining: Post Hoc Interpretability of Latent Factor Models for Recommendation Systems. In *KDD (KDD'18)*. ACM, 2060–2069. https://doi.org/10.1145/3219819.3220072

[28] Zhaochun Ren, Shangsong Liang, Piji Li, Shuaiqiang Wang, and Maarten de Rijke. 2017. Social Collaborative Viewpoint Regression with Explainable Recommendations. In *WSDM (WSDM'17)*. ACM, 485–494. https://doi.org/10.1145/3018661.3018686

[29] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI (UAI'09)*. AUAI Press, 452–461.

[30] Yunzhi Tan, Min Zhang, Yiqun Liu, and Shaoping Ma. 2016. Rating-Boosted Latent Topics: Understanding Users and Items with Ratings and Reviews. In *IJCAI (IJCAI'16)*. AAAI Press, 2640–2646.

[31] Yiyi Tao, Yiling Jia, Nan Wang, and Hongning Wang. 2019. The FacT: Taming Latent Factor Models for Explainability with Factorization Trees. In *SIGIR (SIGIR'19)*. ACM, 295–304. https://doi.org/10.1145/3331184.3331244

[32] Maksim Tkachenko and Hady W. Lauw. 2014. Generative Modeling of Entity Comparisons in Text. In *CIKM (CIKM'14)*. ACM, 859–868. https://doi.org/10.1145/2661829.2662016

[33] Nan Wang and Hongning Wang. 2020. Directional Multivariate Ranking. *arXiv* (2020), arXiv–2006.

[34] Nan Wang, Hongning Wang, Yiling Jia, and Yue Yin. 2018. Explainable Recommendation via Multi-Task Learning in Opinionated Text Data. In *SIGIR (SIGIR'18)*. ACM, 165–174. https://doi.org/10.1145/3209978.3210010

[35] Xiang Wang, Xiangnan He, Fuli Feng, Liqiang Nie, and Tat-Seng Chua. 2018. TEM: Tree-Enhanced Embedding Model for Explainable Recommendation. In *WWW (WWW'18)*. 1543–1552. https://doi.org/10.1145/3178876.3186066

[36] Yao Wu and Martin Ester. 2015. FLAME: A Probabilistic Model Combining Aspect Based Opinion Mining and Collaborative Filtering. In *WSDM (WSDM'15)*. ACM, 199–208. https://doi.org/10.1145/2684822.2685291

[37] Yang Yang, Jie Tang, Jacklyne Keomany, Yanting Zhao, Juanzi Li, Ying Ding, Tian Li, and Liangwei Wang. 2012. Mining competitive relationships by learning across heterogeneous networks. In *Proceedings of the 21st ACM international conference on Information and knowledge management*. 1432–1441.

[38] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit Factor Models for Explainable Recommendation Based on Phrase-Level Sentiment Analysis. In *SIGIR (SIGIR'14)*. ACM, 83–92. https://doi.org/10.1145/2600428.2609579

[39] Zhu Zhang, Chenhui Guo, and Paulo Goes. 2013. Product Comparison Networks for Competitive Analysis of Online Word-of-Mouth. *ACM Trans. Manage. Inf. Syst.* 3, 4, Article 20 (Jan. 2013), 22 pages. https://doi.org/10.1145/2407740.2407744