Towards Local Post-hoc Recommender Systems Explanations

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

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

Willème Verdeaux
Kalidea-Up, University of Tours
Gennevilliers, France
willeme.verdeaux@up.coop

ABSTRACT

Post-hoc explanation aims at defining a simple local surrogate model to shed light on a prediction produced by a complex, generally black-box, model. In the general context of classification, it has been shown that local surrogates may not be able to always capture a local explanation, i.e. for a specific instance prediction, but rather traduce more of a general behavior of the black-box. This problem is even more complex in a recommendation scenario where classes and decision boundaries are not explicitly defined and where data are very sparse by nature. We show in this paper that it is possible to tackle these problems with an efficient sampling around the recommendation instance to explain, to finally learn a proper local surrogate model. Our experiments show that our method is as accurate or better than the methods of the literature while retrieving more meaningful explainable features locally.

KEYWORDS

post-hoc explanation, recommender systems, locality

1 INTRODUCTION

Explainable AI (XAI) [26, 34] aims at understanding the rationale about the factors driving the decision process in complex machine learning models and how their prediction can be altered by changing their input [8]. In this context, post-hoc or model-agnostic explanations have gained some attention in the past years, as they are produced by explanation methods that are agnostic of the internals of the model to explain, and thus need not balance accuracy of the model to explain with the quality of the explanation [26].

The definition of surrogate models is a well-accepted approach in XAI [14], that builds upon the successful LIME algorithm. Figure 1 illustrates the main steps of LIME to define simpler, interpretable models trained to locally mimic the behavior of more complex, possibly black-box, models [13]. Defining such surrogate models involves: (1) the definition of a binary interpretable feature space in which the explainable model is defined, (2) a (ideally bijective) function to pair each instance in the original space to its binary counterpart in the interpretable space and vice-versa (Fig 1(a)(c)), (3) a binary perturbation mechanism to generate a training set for the explainable model around the binary interpretable image of the original explanation instance (Fig 1(b)), (4) the labelling of these training instances by the black-box model when projected back to the original feature space (Fig 1(d)) and finally (5) the learning of a simple model, generally a linear regression model, whose weights attached to the binary features form the expected explanation (Fig 1(e)(f)).

In this paper, we consider the specific case of recommender systems [18], that are notoriously complex prediction systems and for which computing explanation raises new challenges [43].

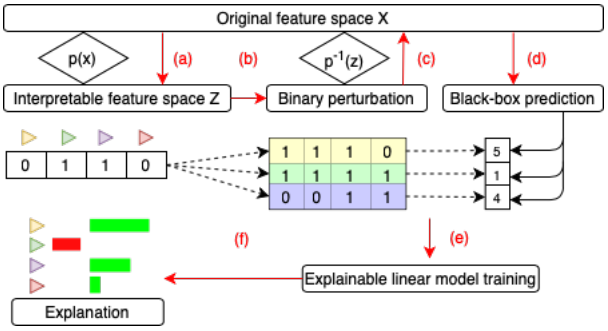

Figure 1: Main steps of LIME to build a simple surrogate model and an explanation to a black-box model prediction.

In this context, we propose to extend recommender systems explanations by reproducing and adapting LIME principles. This task is challenging for two main reasons.

First, because transposing LIME principles to the recommender systems paradigm is not straightforward, as the traditional instances / features data is replaced by a sparse and possibly very large user-item matrix of ratings. In this new context, one needs to redefine what is an instance to explain as the features and class label are not clearly identified, what are the interpretable features, what is a perturbation, and then, when perturbations are generated in the interpretable space, how to project them back to the original space to predict their ratings with the black-box so as to build a training set? Indeed, determining an out-of-sample prediction for new user-items configurations can be a complex task depending on the nature of the recommendation mechanism.

Second, LIME relies heavily on an estimation of locality around an explanation instance to ensure the quality of its prediction. However, surrogate models as proposed by LIME are not robust, in the sense that they sometimes fail to estimate correctly the explanation model around a specific instance, therefore producing a too general explanation model that accounts for a broader range of input instances [4].

We argue in this paper, that these problems are related to: (1) the binary perturbation mechanism that does not ensure that perturbed instances falls in the vicinity of the original instance when projected back to the original space, and (2) the definition of locality as a decreasing neighborhood function to balance the aforementioned effect of binary perturbation and that do not take into account decision boundary. Noticeably, in the case of recommender systems, this problem is even more critical since there is no explicit decision boundary per se.

To tackle the aforementioned challenges, we propose a new local surrogate model dedicated to recommender systems, named LIRE - Local and Interpretable Recommendations Explanations - that improves over the reference in the literature, the LIME-RS model [29] that is a direct port of LIME, in terms of quality of

the explanation by better estimating the locality of an instance to explain, and while still maintaining consistent recommendation fidelity to the original recommender system.

As such our contributions are: the definition of a new representation of a user-item instance to explain, the introduction of a real valued interpretable feature space instead of a binary interpretable space paired with an out-of-sample prediction mechanism to project perturbed instance back to the original space for the specific case of matrix factorization recommender systems. Most importantly, we propose a new definition of locality to better tackle the notion of decision boundary in recommender system decision space by coupling a new gradual perturbation mechanism, with off-the-shelves clustering algorithm and dimensionality reduction techniques such as UMAP [25] so that all users are represented in a low-dimension space where classical metric distance applies effectively. Finally, extensive comparative experiments on the MovieLens benchmark show that our new local surrogate approach is comparable in terms of prediction accuracy if not better than LIME-RS, while proposing more relevant set of interpretable features as explanations.

This paper is organized as follows: Section 2 presents the problem formulation and Section 3 details our main contributions. Section 4 presents our experiments and Section 5 presents a discussion about the problematic in the field of explainable AI before Section 6 concludes and opens future works.

2 PROBLEM FORMULATION

In what follows, we consider a (black-box) **recommendation system** as a function $f : U \times I \rightarrow \mathbb{R}_+$ where U is the set of users, I is the set of items and $\mathbb{R}_+$ is the definition domain of the ratings.

2.1 Explanation instances, interpretable features and explanations

We call an **explanation instance** the 3-tuple $\langle u, i, f(u,i)\rangle$ where $u \in U$, $i \in I$ and $f(u,i) \in \mathbb{R}_+$ denotes a prediction that we want to explain and produced by the (black-box) recommender system for the user u and item i. Importantly, our objective is not to explain the process by which the black-box recommender system works, but instead to highlight the main interpretable features that explain a specific prediction $f(u,i)$.

In [29, 33], **interpretable feature** relates to feature names that represent directly understandable and actionable pieces of domain knowledge. The representation of an (explanation) instance in the interpretable space is thus a set of interpretable feature names, technically represented by a binary feature name vector.

In our work, we denote by interpretable features the set I of n items names $I = \{I_1, \ldots, I_n\}$, each associated with a domain of value $dom(I_1), \ldots, dom(I_n) = \mathbb{R}_+^n$. We call a feature vector over I a n-tuple t of values $t = \langle c_1, \ldots, c_n\rangle$ where $t \in \mathbb{R}_+^n$. Equivalently, and whenever this is convenient, we view the tuple as a function t of signature $I \rightarrow \cup_k dom(I_k)$, denoting $t(I_k)$ the value c_k and $t_{|_{I'}}$ the restriction of t to the subset $I' \subseteq I$. Consequently, $t(I_k) = t_{|_{\{I_k\}}}$.

Following the previous notation, for an explanation instance $\langle u, i, f(u,i)\rangle$, the list of interpretable features is formalized as the tuple $t^u_{|_{\cup_{j\neq i}\{I_j\}}}$, i.e. the restriction of t to the subset of items $j \neq i \in I$ for a specific user $u \in U$. Noticeably, and contrary to previous works, this representation of an explanation instance is not binary but associates a real value to each feature name (in fact, the score of the item for this user u). This allows to express more complex perturbation mechanisms as presented in Section 3 and to better preserve locality as shown in experiments in Section 4.

Finally, in our case, and similarly to [29, 33], an **explanation** is a set of interpretable feature names, technically represented by a real-valued feature name vector, where each interpretable feature name is associated with a real weight representing the importance of the feature for the explanation instance.

2.2 Explanation model

Traditionally, explaining a recommendation boils down to determining the top-n [29] or minimal [33] subset of interpretable features that maximizes the fidelity of the surrogate model to the original model. As in [33], we restrict our work to the class of linear explanation models $g(\mathbf{z}) = \mathbf{w} \cdot \mathbf{z}$, where $\mathbf{z}$ denotes the vector of values attached to the set of interpretable features.

Consider that $\mathbf{t^u}$ is the vector of values attached to interpretable features of user u to explain instance $\langle u, i, f(u,i)\rangle$, constructed from $t^u_{|_{\cup_{j\neq i}\{I_j\}}}$. The explanation model $e_f(u,i)$ is defined as follows:

$$e_f(u,i) = \text{argmin}_{\mathbf{w}\in\mathbb{R}^n} \mathcal{L}\left(f(u,i), \mathbf{w}\cdot\mathbf{t^u}\right) + \Omega(\mathbf{w}) \qquad (1)$$

where $\mathcal{L}$ is a loss function that penalizes any difference between the original prediction $f(u,i)$ and the value predicted by the surrogate model. As in LIME [33], $\Omega(\mathbf{w})$ represents the complexity of the linear explanation model. Here, we expect our feature weighting to be parsimonious, i.e., we want that our approach discards as many as possible of the interpretable features to ease the posterior interpretation of the explanation.

As a conclusion, our problem reduces to determining the most appropriate interpretable features weights vector $w \in \mathbb{R}^n$. Our hypothesis in this paper is that it is possible to improve the quality and the relevance of w by introducing locality in the sampling process to generate the training set to learn w.

3 PROPOSED SOLUTION

As mentioned in the previous sections, our proposal is to introduce locality during the sampling of instances to train our linear surrogate model following two complementary directions: (1) by generating gradually perturbed instances around the explanation instance, and (2) by discovering natural grouping of neighbor users that share similar items scoring behaviors and for which a robust explanation system should provide close explanations.

This first two objectives raise secondary questions. Noticeably, (1) raises the problem of being able to predict a recommendation for never-seen-before users in the rating matrix, as perturbed instances may not correspond to pre-existing ones. This mechanism is called Out-Of-Sample (OOS) prediction hereafter, and we present a method for this problem in the context of the popular matrix factorization recommendation approach. (2) necessitates to be able to cluster points expressed as very large item scores vectors efficiently without falling into the curse of dimensionality problem.

3.1 Locality via the perturbed instances generation

In LIME [33] and LIME-RS [29] surrogate models are trained on instances of the original space that are antecedent of perturbations generated in the binary interpretable feature space associated to each explanation instance. Then for each perturbed instance, a prediction is performed to build a training set to learn the surrogate model. We develop hereafter how we adapt these two steps in our proposal.

3.1.1 Perturbation of user instance. Given our definition of explanation instance and its representation as a real valued vector attached to feature names (see Section 2), we define perturbations as a random modification of the values of tuple $t^u_{|_{\cup_{j\neq i} I_j}}$ based on some 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 of probability p_B the chance to modify each element of $t^u_{|_{\cup_{j\neq i} I_j}}$. For simplicity sake, we will note interchangeably $t^u_{|_{\cup_{j\neq i} I_j}}$ as t^u in the context of an explanation instance $\langle u, i, f(u,i)\rangle$ and in this case, p^u will denote a perturbation of t^u.

3.1.2 Out-Of-Sample prediction. In terms of LIME methodology, Out-Of-Sample prediction plays the role of the surjective function p^{-1} between interpretable feature space and original space as well, as the prediction $f(p^{-1}(z'))$ attached to this new instance z' (following notations in Equation 10). The role of this function is, given an interpretable feature description of a perturbed user p^u, to find a representation in the original space for this perturbation, and most importantly, as this user is totally new to the recommender system, to predict his/her rating for the item $i \in I$.

In our proposal, where we generate never-seen-before user-items signatures, determining a prediction for these new instances can be challenging. Indeed, the difficulty to implement such OOS procedure depends heavily on the recommender system: in case of k-nearest neighbor or baseline approach, it is trivial to implement a recommendation for a new user. The same holds for deep embedded recommender systems [41] which, by design, can produce prediction for any new input. In our work as in [29] however, we consider the special case of Singular Value Decomposition recommender system [5, 32], a Matrix factorization method that expresses the user-item matrix:

$$\mathbf{R} = \mathbf{W}\Sigma\mathbf{V^t} \tag{2}$$

with $\mathbf{R}$ a $m \times n$ real valued user-item matrix, and where Σ can be reduced to a diagonal matrix $\mathbf{\Sigma_{kk}}$ of size $r \times r$ ($r < m$ and $r < n$) containing only the r largest singular values of $\mathbf{R}$. In this context, the ratings for an out-of-sample user v are determined as follows:

$$\mathbf{R}(v, .) = \mathbf{r_v}\Sigma V^t \tag{3}$$

where $\mathbf{r_v}$ represents the $1 \times r$ vector representing user v in user latent space. In this context, producing an OOS prediction accounts for determining the latent representation $\mathbf{r_v}$ for OOS user v.

To this aim, we formulate the search of latent representation $\mathbf{r_v}$ as a least square optimization of the residual sum of square (RSS) defined as:

$$RSS = (p^u - \mathbf{r_v}\Sigma V^t)(p^u - \mathbf{r_v}\Sigma V^t)^t \tag{4}$$

between estimated ratings computed on $\mathbf{r_v}$ and the perturbation vector p^u that provides, in our case, all ratings but the one for item $i \in I$.

Finally, the ground truth rating for item $i \in I$ is obtained by re-injecting the optimal $\mathbf{r_v}$ into Equation 3.

3.2 Locality via instance neighborhood generation

Ideally, our locality definition should also be coherent with existing decision boundaries. In the context of recommender systems, such decision boundaries are not explicit. Our contribution concerns the determination of such explicit decision boundaries via the definition of natural neighborhood for each explanation instance. In our context, determining the neighborhood reduces to a clustering problem of the tuples $T = \{t^u\}_{u\in U}$. We mean by "natural", a grouping of tuples T such that a traditional clustering quality criterion is met, for example minimizing the intra-cluster variance as in k-means clustering [16, 24] or ensuring that there exists a transitive density relation between connected neighbors as in DBSCAN algorithm [10, 35].

However, following our previous definitions of an explanation instance, an input $t^u \in T$ representing a user in an interpretable feature space can still have several thousands of features as in our case, this relates to the set of items. This makes clustering useless because of the loss in discrimination attached to the metric used to perform the clustering, what is known as curse of dimensionality.

Several solutions exist in the literature to solve this dimensionality problem. The first solution is to perform a feature selection or weighting process on the instances in T prior to the clustering. This research domain has been extensively studied in the past years as attested by numerous publications [3, 6, 20, 22]. The difficulty lies in the definition of an objective to drive the feature selection process (as, contrary to supervised classification, there is no ground truth). An other solution would be to define clusters and their respective set of features at the same time with subspace clustering methods [2, 19, 30]. However, these approaches are generally complex and will not scale with the size of datasets in the recommender systems world.

Moreover, as our final objective is not to build a clustering per se, but to build neighborhoods as clusters from which we estimate a local explanation, we do not want to remove beforehand any information that could explain a local behavior.

For these reasons, we consider in this paper dimensionality reduction techniques such as t-SNE [40] or the more recent UMAP [25]. UMAP builds a relationship graph in high dimensionality by growing around each instance a radius that denotes the strength of the relationship with neighbors. Similar to t-SNE this high dimensional graph is then reproduced in a lower dimension space. Interestingly, UMAP provides parameters to balance the importance of respecting the local relationship versus the global structure of a data set.

In this paper, we use UMAP in conjunction with a k-means clustering. Sensitivity of our approach to these choices is not reported here for the sake of readability and, as illustrated in experiments in Section 4, will need further discussions that are left as future work.

3.3 Learning the surrogate model weights

Previous sections details how locality is introduced in the sampling of instances to build a proper training set for our surrogate

model. We now present the different variants of our approach LIRE depending on how the training set is constructed from perturbed points and cluster neighbors. Finally, the last subsection describes how the linear weights $w \in \mathbb{R}^n$ of our local surrogate model are learned.

3.3.1 Building the training set for the surrogate model. In our proposal we consider three different scenarios to define the training set of the explanation instance t^u, each one related to a locality definition.

First, in the LIRE-C approach, we consider randomly picked users from the same cluster as the user u for which the explanation is to be computed. These neighbors represents observed examples of users in the close vicinity of u. Their rating for the item $i \in I$ can be estimated by the black-box directly, that serves as ground truth. This approach is expected to be faster as there is no Out-Of-Sample prediction involved. If the cluster is smaller than the number of training instances, instances are duplicated.

Second, we consider in the LIRE-P approach only perturbations p^u as presented in the previous Section 3.1.1 following the LIME principles. This approach is supposedly the most accurate as it allows to generate numerous training examples in the close vicinity of an explanation instance and by modifying gradually the importance of each interpretable feature.

Finally, the last approach LIRE-M considers a mixed situation where half training instances originates from perturbations and the other half is generated from the cluster neighbors.

3.3.2 Explanation as a weighted regression with L1 regularization. In our context, learning the best explanation amounts to determining the weights of the most appropriate interpretable features. We formalize this problem as a simple regression problem between a training set $\mathcal{T}_{train}$ of instances expressed on interpretable features composed of either perturbations or cluster neighbors of user u and their respective predictions $\mathcal{Y}_{train}$ either obtained by direct prediction of the black-box or by our OOS prediction. We further want to achieve the simplest explanation by only retaining the most interesting features.

To do so, we consider a LASSO regression model [38] introducing a penalty term $\|\mathbf{w}\|_1$ similar to $\Omega(\mathbf{w})$ in Equation 1. In the end, our explanation can be optimized following:

$$\mathbf{diff} = \mathcal{Y}_{train} - \mathbf{w}\mathcal{T}_{train} \tag{5}$$

$$e_f(u,i) = \text{argmin}_{\mathbf{w}\in\mathbb{R}^n} \left\{ \mathbf{diff} \cdot \mathbf{diff}^t + \lambda \|\mathbf{w}\|_1 \right\} \tag{6}$$

where formally λ denotes the Lagrangian coefficient attached to the constraint that minimizes the sum of weights $\mathbf{w}$. Noticeably, in our implementation, we use a LARS [9] algorithm with no intercept to find the optimal value w.

4 EXPERIMENTS

This section describes the experiments conducted to assess the interest of our approach and the way it deals with locality to provide explanations. To do so, we have set up several experiments reported hereafter that aims more specifically at answering the following questions:

(1) **what is the impact of the internal sampling in the performance of our approach?** Should we use exclusively perturbed points around the explanation instance, neighbors from the cluster to which the explanation instance belongs or a mix of the two? As previously mentioned, to this aim, all of our experiments compare 3 settings for our LIRE algorithm: (1) LIRE-P stands for LIRE with exclusively perturbed points as sampling approach, (2) LIRE-C stands for LIRE with only neighbors from cluster and finally (3) LIRE-M represents the mixed approach with 50% perturbed points and 50% neighbors points.

(2) **how our approach compares with LIME-RS?** LIME-RS [29] is the reference method from the literature that first adapted the principle of LIME [33] to the context of recommender systems. LIME-RS considers locality only through its objective function (see Equation 9) by according more importance to training instances closer to the explanation instance. We show in our experiments that our new approach obtain comparable if not better results than LIME-RS which validates our hypothesis of integrating a more local sampling method to train the surrogate model.

(3) **how to take into account the specific nature of the recommendation task?** A recommendation is a prediction of multiple item scores that aims at ranking items to present the more meaningful to a user. In this respect, we propose 3 evaluation scenarios: (1) similarly to classification, we pick at random explanation instances (i.e. a user and an item) and evaluate the accuracy of the surrogate model to predict the black-box output, but (2) we also evaluate our approach in the context of explaining the first ranked item (called *top* hereafter) for a specific user as well as, (3) the latest ranked item (called *flop*). These two scenarios are much more aligned with a real usage of a recommender system as one may want to know why an item was recommended first and why one item was not recommended.

(4) **how meaningful and relevant is our explanation?** Similar to LIME, our approach outputs a weighted vector of interpretable features. We cannot only evaluate a surrogate explanation model based on its accuracy to the black-box. We also need to determine if the interpretable features used as an explanation are the one that were expected. To do so, we propose an experiment called *Single White Box* in which we define a linear white box model relying on 10 items (our features) that have been scored by the user to whom the explanation is proposed. This white box is supposedly the recommendation model that the surrogate tries to replicate. It is thus possible to compute the ratio of interpretable features discovered versus those expected from the white-box model. Then, as we are in a recommendation context, we also produce a ranking of the interpretable features that our approach discovers and compares it to the ranking of the white-box features based on their respective weights. We produce Normalized Discounted Cumulative Gain (NDCG) [17] measure as it is a common way to evaluate the agreement between two items rankings [28]. In our case, we use NDCG measure to evaluate how many of the most important features our approach mines among the top 3, top 5 and top 10.

(5) **how good our approach deals with locality in the recommendation process?** Our previous tests consider that the black-box model (either matrix factorization or a linear white-box) has the same general behavior among all explanation instances. We propose an experiment, called

Double White-Box, where we define 2 white-box models, each defined on 5 features that are distinct from the 5 features from the other model. The key idea is to define artificially a locality around an explanation instance as an area whose radius is computed as the distance to its k-nearest neighbor. When building the training set of our explanation approach, if training point is inside the local area, its prediction for the item will be produced by the first model, and otherwise will be produced by the second model. As such, we define a distinct behaviour depending on the locality. The objective is to verify if our local approach manage to capture this information better than our reference approach from the literature.

4.1 Datasets, black-boxes, evaluation metrics and general protocol

Datasets. Two well-known datasets from the movie recommendation service MovieLens have been used. Each describes 5-star rating and free-text tagging activity from MovieLens. We limit our main tests to the $100K$ MovieLens dataset [15] with 610 users and 9.724 items, as answering our research questions involves multiple runs with different parameters that would be too time consuming on larger datasets. Then, a first evaluation on the MovieLens 20M entries dataset is provided as a testimonial that our approach can scale to larger volume of data. This dataset contains 20.000.263 ratings generated by 138.493 users for 27.278 movies.

Black-boxes. we consider 2 different types of black-boxes algorithms. Similar to [29] we first implement a simple matrix factorization method. The interest of such approach lies in its ability to produce meaningful recommendation from a latent space of users and items even in the context of very sparse data. The difficulty for post-hoc explanation approaches such as LIME [33], that relies on perturbed points to train a surrogate model, is that it is not trivial to produce an Out-Of-Sample (OOS) prediction for these perturbed instances. In our contribution, and contrary to previous works that avoid this situation by considering only pre-existing user-items recommendations as training, we propose a proper method to perform the OOS prediction as introduced in Section 3.1.2. The second "black-box" is the linear white-box that we use to determine the quality of our explanation (see research questions (4) and (5)). To this extent, for a given explanation instance $\langle u, i, f(u, i)\rangle$, we pick at random 10 items that were evaluated by user u (excluding item i). The weights of these 10 items are randomly set between 0 and 1. The linear combination of weighted items / features produces the expected linear white-box recommendation model. In the case of research question (5) where we consider 2 white-box models, each model uses only 5 out of 10 of the previously selected items / features, so as to be clearly differentiated. In these comparisons with white-boxes, all compared methods are asked to produce an explanation over the 10 most interesting features that they identify among the set of all 9724 features.

Evaluation metrics. We consider several evaluation metrics to assess the quality of our post-hoc explanation approach:

- the **accuracy** to the black-box model is classically 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 **computation time** is estimated in seconds for one run of an approach. Here we only measure the computation time of our LIRE approach to observe the impact of OOS prediction computation versus clustering;
- the **relevance** of our interpretable model g is expressed as the ratio of features from the white-box model that are discovered by g (i.e. features whose weights exceed 0 in the model g). Let $\mathcal{F}$ be the set of features of a model, this ratio can be expressed as follows:

$$rel(f, g) = \frac{\mathcal{F}(f) \cap \mathcal{F}(g)}{\mathcal{F}(f)} \tag{7}$$

- the **feature ranking quality** is more discriminant than the previous relevance metric since it takes into account the rank of the relevant features and not only their presence / absence in the set of retrieved features. Ranks of interpretable features are provided by their weights, either set randomly in the white-box model, or learned by the surrogate model. To measure the quality of the agreement between the expected and the learned ranking, we use the traditional Normalized Discounted Cumulative Gain (NDCG) measure at rank ρ (NDCG@ρ) for values of $\rho \in \{3, 5, 10\}$.

4.2 Evaluation on matrix factorization black-box

This section reports our comparative experiments between LIME-RS and our three approaches LIRE-P (only perturbed training instance), LIRE-C (only cluster training instance), LIRE-M (mixed training instances) when explaining recommendations instances from a matrix factorization black-box model.

Protocol. For each method, we report evaluation metrics averaged over 50 explanation instances $\langle u, i, f(u, i)\rangle$. In the *random* scenario, these instances are generated by picking at random a user u and an item i from the Movie Lens 100K dataset and by predicting the rating for $\langle u, i\rangle$ based on the black-box function f. In the *top* and *flop* scenarios, only the user u is picked at random, then the black-box is used to determine items with the highest and lowest scores for user u and their respective scores as ground truth. Following the parameters of LIME-RS, the training set size of the surrogate for each explanation instance is set to 1000. In LIRE-P, 1000 perturbed points are generated with a probability $p_B = 0.1$ in the Bernouilli process with a perturbation range following the estimated overall variance set to 1.04 on the non-zero ratings. In LIRE-C, 1000 neighbors from the same cluster are considered. This is a huge constraint in our clustering model since this size of training set may not allow for a finer clustering algorithm that captures small tendencies in the dataset. As a compromise, in case the cluster is too small, we propose to replicate its data, which allows to use a k-means clustering algorithm set with 75 clusters to preserve small clusters and locality. Clustering is applied on top of a UMAP dimensionality reduction as implemented in Python `umap` package with 30 neighbors and a minimum projection distance of 0.01. The linear regression is based on LARS implementation and uses default parameters as presented in its `sklearn` version. All code is written in Python and is available as a Git project [1]. All tests on MovieLens 100K were run on a laptop with an Intel Core i7 CPU at 2.50GHz and 8 GB of RAM.

[1] https://github.com/wil0u/Lire-DOLAP2021

Method	Random	Top	Flop
Lire-P	0.47 ± 0.48	0.89 ± 0.90	1.96 ± 1.41
Lire-M	0.45 ± 0.47	1.18 ± 0.99	1.68 ± 1.21
Lire-C	1.18 ± 1.06	1.64 ± 0.90	1.93 ± 1.05
LIME-RS	0.39 ± 0.53	1.59 ± 0.76	1.55 ± 1.09

Table 1: Average and standard deviation of MAE accuracy in the Random and Top-Flop experimentation scenarios

Methods	Computation times (sec.)
LIRE-P	27.00 ± 1.98
LIRE-C	0.57 ± 0.08
LIRE-M	12.85 ± 2.08
LIME-RS*	1.60 ± 0.36

Table 2: Average computation times (in sec.) and their standard deviations for the *Random* scenario for the LIRE variants. LIME-RS computation time is only provided as a reference only as its implementation is not optimized from the original source code.

Out-Of-Sample Prediction. The OOS predictors run for 120 epochs of gradient descent using the `optim` package of `pytorch`. More specifically, within this package we use the `Adagrad` optimiser using its defaults parameters, exception made of the learning rate set to 0.1.

Comparative accuracy. Table 1 presents the results of our first comparative study based on MAE accuracy measure between the black-box and our surrogate models. In order to assess our results validity, significance t-tests have been conducted. It can be observed that on the *R*andom scenario, LIME-RS, LIRE-P and LIRE-M have comparable results (p-value of $0.41 > 0.05$ between LIME-RS and LIRE-P for example), and all 3 approaches manage to estimate correctly the prediction of the black-box. Interestingly, standard deviation values are very high which shows that, even if for most of the cases the accuracy is very good (MAE close to 0), there still exists some cases that should be investigated in the future, where the methods fail to estimate correctly. This is the case with LIRE-C that relies exclusively on cluster neighbors to train the surrogate model and that is less accurate. This is certainly due to the inadequacy of discovered clusters to represent correctly the locality either being to small and too local or being too large and thus providing a surrogate model that is too general. We leave as future work an in-depth study of the most efficient clustering for sampling. Interestingly though, mixing perturbed instances and clusters do not deteriorate the results.

In the *T*op scenario, differences are significant between LIRE-P and LIME-RS (p-value = $8 * 10^{-5}$), and between LIRE-M and LIME-RS (p-value = 0.02) which shows that our approach is more efficient when dealing with the best scored items, i.e. those that will be preferentially presented to the user. This is mainly due to the internal behavior of LIME-RS that builds the "locality" around an instance by mostly exchanging items to be scored. As a consequence, training instances will most likely consider lower scored items and in turns will not be able to capture accurately the behavior for the top instances. On the contrary, our perturbed instances paired with our OOS prediction allows for a smoother estimation of the model around the top instances. Interestingly, our approaches do not perform as well as in the Random scenario, as it is more difficult to build a representative training set: in the case of LIRE-P, perturbations are limited to the 0 to 5 range of ratings and in the case of LIRE-C, top scores may not have many neighbors to compare with, because of the generally Zipfian distribution of ratings. The same conclusion applies in the following Flop scenario.

Finally, in the *Flop* scenario, similar to what is observed for the *Random* scenario, there is no significant difference between the results (p-value = $0.10 > 0.05$ when comparing LIRE-P and LIME-RS for example). Interestingly, when mixing perturbed points and cluster neighbors it seems to improve on our sample test the performances of LIRE (not significantly though).

Computation times. During all the experiments, we have also monitored the computation times of the different variants of LIRE as reported in Table 2. Only *Random* scenario is reported as the other scenarios are exactly as intensive in terms of computation. Noticeably, LIRE-C based on cluster neighbors is the fastest approach and LIRE-P is the slowest of the batch. This was expected since LIRE-C does not require the computation of the OOS prediction in LIRE-P. The latter is very costly when considering matrix factorization black-box. Interestingly, LIRE-M provides a good speed-up over LIRE-P without degrading the accuracy in *Top* and *Flop* scenarios. Future work should investigate more this results as a LIRE-P with only 500 training instances may have the same performance and the same computation time as LIRE-M depending on the quality of the clustering used to define the neighborhood. Noticeably, our clustering may not be as efficient as expected because in many real world situations, the size of clusters follows a Zipf law with one very large cluster and many very small clusters. As a consequence, we set up our clustering parameters so as to perform a compromise between the ability to capture small trends in the data as well as more general tendencies. In the case where a cluster is too small to contain 1000 points to produce an equivalent training sample as the other approaches, we use an oversampling technique that may bias the convergence of the surrogate model, hence the performance. We leave these research questions as future work.

4.3 Single white box experiment

Table 3 contains for each method the evaluation of its ability to identify correctly the interpretable features used to generate the white-box model. Relevance does not consider the relative importance of each interpretable feature while NDCG score takes into account the ranking of the most important interpretable features of the white-box model.

This is a very difficult test as it boils down to identifying 10 features out of 9724. These features may be correlated and so in this case our surrogate has to determine which one of the correlated features to pick to explain the general behavior of the white-box. Finally, the difficulty of the task is also related to the number of items that were scored by the user for which the explanation is produced. Indeed, even when limiting the exploration of interpretable features to the set of scored items, this resolves to a possibly large search space: on average each user of the dataset scored around 614 ± 642 items. This shows that the search space may be very small or very large depending on the selected user.

First, it is interesting to notice that in this experiment, LIME-RS is not able to identify any correct interpretable feature from the set of 9724 candidates. This is due to the internal behaviour of the approach as we use in this experiment the "item mode" from the original code that is emphasized in the original paper [29]. In this mode, the sampling to train the surrogate model only varies, for a specific fixed user, the items that are considered. As a consequence, because of the nature of the white-box that is a linear combination of fixed items, the output for all training instances is the same. As a consequence, LIME-RS has no real information to decide from all features locally and tries to minimize the prediction error but on the basis of uninteresting features that are certainly correlated to some extent to the one used in the white-box (hence the good accuracy score).

Second, the baseline random approach chooses from the set of scored items for the user. This constraints greatly helps to find more easily some relevant features (but otherwise this baseline would have been pointless since it would have had 10 chances out of 9724 to find a correct feature), but does not favor the discovery of a proper ranking of the important interpretable features as illustrated by the very low NDCG scores.

Finally, LIRE in general, and more specifically LIRE-P manages to better embrace the local behavior of the white-box and identify a ratio of 0.212 features over which its model is constructed. This is clearly due to the perturbation mechanism that will slightly affect the scores of items to explore the neighborhood of an explanation instance gradually and thus can better evaluate the relative importance and correlation between features.

4.4 Double white box experiment

The double white-box experiment aims at showing to which extent each surrogate model learns locally the black-box model and to which extent it performs well doing so, based on the previous quality measure of an explanation (see Section 4.3). For each explanation instance generated randomly, we define a decision threshold that is set as the distance to the k^{th} nearest neighbors or as the distance to the farthest point in the same cluster, if there are less than k instances in the cluster.

Table 4 presents the results obtained when comparing all the methods based on an adapted relevance metric: "Relevance In" indicates the ratio of features from the first model (the one applied inside the neighborhood delimited by the decision threshold) that are identified by the surrogate model, while "Relevance Out" designates the ratio of features from the out-of-neighborhood model. A surrogate model that better approximate a local behavior is likely to have a better "Relevance In" score.

First, it can be seen in Table 4 that LIRE-P approach performs the best for the "Relevance In" score with 0.328, followed by LIRE-M and the random baseline. This shows that LIRE-P is the most effective locally to capture the important features of the in-neighborhood white-box model. This is due to its gradual perturbation mechanism that stays in the vicinity of the explanation instance, contrary to the binary perturbation of LIME-RS that does not guarantee that a perturbed instance stays in a close neighborhood. Second, concerning LIRE-M and LIRE-P, it can be observed that adding training instances from the cluster decreases the quality of the explanation. Indeed, LIRE-C has very poor results which tend to show that clusters are generally much larger than the radius defined by the decision threshold and that, in this case, training instances are often labelled by the out-of-neighborhood white-box model prediction. Finally, LIME-RS obtains very poor results with 0.036 in "Relevance In". In fact, this is due to the setting of the approach for this test, where we have changed the item mode of the previous test (that would not have performed correctly for the same reasons as previously, see Section 4.3) to the user-item mode, present in the original code from [29] but not detailed much in the paper. In this mode, training set is constructed by picking at random (user, item) pairs and their associated black-box predictions, which, as the locality is by definition smaller than the whole explanation instances space, leads to favor the out-of-neighborhood white-box to label its predictions. However, even in the case of the "Relevance Out" score, LIME-RS does not perform well because of the binary perturbation that does not ensure locality correctly. NDCG scores confirm that LIRE-P also manages to discover more of the main features coming from one or the other white-box model and better respects their ranking with a good score of 0.355 for NDCG@3.

4.5 Test on MovieLens 20M

Table 5 finally reports the comparative results of our 3 methods on the larger 20M entries dataset proposed by MovieLens. All tests were conducted on a AMD Ryzen 3700X with 32 GB of memory, and as such, computation times with previous experiments cannot be compared, but are on the same order of magnitude. First, it should be observed that our approach runs on 20M entries while the code provided for LIME-RS [29] is not able to run on the full dataset. Second, it can be seen that LIRE-M has better results compared to the test on MovieLens 100K and has similar performances in terms of MAE than LIRE-P. Indeed, differences are not significant when considering a bilateral t-test with a p-value equals to 0.18. This is interesting as it may indicate that in this very large dataset scenario, the clustering could help improving the results of the perturbation mechanism. This should be investigated more in future work. However, the cluster neighbors only training set is too dependent on the quality of the clustering algorithm to be efficient on average as shown with LIRE-C results. Finally, and similarly to our experiments on MovieLens 100K, LIRE-C is the fastest approach as it does not involve the OOS prediction mechanism.

5 RELATED WORK

Explainable recommendations refers to personalized recommendation algorithms that not only provide the user with recommendations, but also make the user aware why such items are recommended [42]. Gedikli et al. [12] evaluate different explanation types and propose a set of guidelines for designing and selecting suitable explanations for recommender systems. Indeed, state-of-the-art recommender systems [18] are notoriously complex prediction systems for which computing explanation raises new challenges [43]. *Model-intrinsic explanations* correspond to recommender systems whose decision process is simple enough to be clear for the users or that embed mechanisms to provide users with an explanation [1]. However as pointed out in [23], this kind of explanations suffer from a trade-off between transparency and accuracy of the model. Indeed, adding internal mechanisms to explain a process or a result may slow down this process or bias this result as the sole focus of the recommender system is no more the accuracy of item scores prediction but to produce a justification for these scores as well.

Methods	Relevance	NDCG@3	NDCG@5	NDCG@10
LIME-RS	0 ± 0	0 ± 0	0 ± 0	0 ± 0
Baseline Random	0.048 ± 0.083	0.025 ± 0.105	0.034 ± 0.094	0.045 ± 0.088
Lire-P	0.212 ± 0.206	0.269 ± 0.345	0.284 ± 0.282	0.255 ± 0.250
Lire-M	0.098 ± 0.160	0.093 ± 0.216	0.098 ± 0.199	0.102 ± 0.181
Lire-C	0.030 ± 0.061	0.078 ± 0.185	0.042 ± 0.100	0.048 ± 0.092

Table 3: Single white-box experiments metrics: average relevance and NDCG scores for LIME-RS, the 3 LIRE variants and a random baseline that builds a surrogate by picking 10 features and their weights at random.

Methods	Relevance In	Relevance Out	NDCG@3	NDCG@5	NDCG@10
LIME-RS	0.036 ± 0.077	0.012 ± 0.040	0.042 ± 0.138	0.343 ± 0.094	0.028 ± 0.069
Baseline Random	0.068 ± 0.111	0.080 ± 0.151	0.057 ± 0.135	0.057 ± 0.112	0.056 ± 0.103
Lire-P	0.328 ± 0.327	0.072 ± 0.119	0.355 ± 0.367	0.291 ± 0.301	0.259 ± 0.250
Lire-M	0.116 ± 0.167	0.040 ± 0.098	0.096 ± 0.227	0.099 ± 0.196	0.087 ± 0.152
Lire-C	0.020 ± 0.060	0.012 ± 0.047	0.018 ± 0.088	0.016 ± 0.060	0.014 ± 0.047

Table 4: Double white-box experiments metrics: average relevance and NDCG scores for LIME-RS, the 3 LIRE variants and a random baseline that builds a surrogate by picking 10 features and their weights at random. Relevance In (resp. Out) indicates the ratio of features from the inside-neighborhood (resp. out-of-neighborhood) model. A more local surrogate model will obtain better results in Relevance In.

Methods	MAE	Computation times (sec.)
LIRE-P	0.647 ± 0.560	5.796 ± 0.072
LIRE-M	0.516 ± 0.388	4.693 ± 0.047
LIRE-C	1.075 ± 0.914	3.109 ± 0.081

Table 5: Average MAE and computation times (in sec.) and their standard deviations on the MovieLens 20M entries dataset for LIRE-P, LIRE-M and LIRE-C.

On the contrary *post-hoc or model-agnostic explanations* do not require to access or to adapt the internals of the recommender system and thus do not decrease their accuracy [26].

Many of those post-hoc approaches have been proposed such as [31] for Matrix Factorization or similar approaches that relies on the elicitation of latent factors to perform recommendation [11, 31, 37, 44]. The inherent difficulty facing these methods is to determine an efficient way to relate the latent model to explicit interpretable features that make sense for the user. [37] integrate regression trees to guide the learning and further explain latent space while [11] introduce a framework based on deep multi-view learning to model an explanation as multi-level features template. Finally, [44] propose an Explicit Factor Model that builds an alignment between interpretable features and the latent space while [31] search for association rules expressed on features. All these explanation approaches however are tightly related to only one specific recommendation system. More recently, [39] introduces GLIDER, a system that provides an interpretation for any black-box recommender system based on features interactions rather than features significance as in the original LIME algorithm. In our work, we are interested in model agnostic local explanations as provided by LIME, in other words, models that can provide explanations as a set of feature weights, for any recommender system, given an input instance.

In this respect, the LIME-RS approach [29] provides a model agnostic explanation system, that can be applied on any black-box recommender system and that outputs a set of interpretable features and their relative importance. LIME-RS builds upon the well-known LIME approach [33] to explain recommendation by retrieving the top-n binary interpretable features as computed by LIME.

An explanation produced by LIME for an input instance $x \in \mathcal{X}$, and a prediction model f is as follows [33]

$$\xi(x) = \text{argmin}_{g \in G} \mathcal{L}\left(f, g, \pi_x\right) + \Omega(g) \tag{8}$$

where $\mathcal{L}$ is a fidelity function to the original (black-box) model f and $g \in G$ is one explanation model among all possible explainable models G. The most common explanation model is a linear prediction model and in this case an explanation corresponds to the weights of the most significant interpretable features whose combination minimize the deviation to the black-box model. Interestingly, π_x is a locality measure around instance $x \in \mathcal{X}$ and is introduced to balance the perturbations introduced in the training set. Finally, $\Omega(g)$ measures the complexity of explanation model g. LIME assumes (i) an interpretable feature space $\mathcal{Z}$ to learn a surrogate model of f and (ii) at least a surjective function from $\mathcal{X}$ to $\mathcal{Z}$.

The fidelity function $\mathcal{L}$ is expressed as a quadratic error between the predictions $f(x')$, for instances $x' \in \mathcal{X}$ and the surrogate prediction $g(z')$ for their interpretable counterparts $z' \in \mathcal{Z}$:

$$\mathcal{L}\left(f, g, \pi_x\right) = \sum_{x' \in \mathcal{X}, z' \in \mathcal{Z}} \pi_x(x') \left(f(x') - g(z')\right)^2 \tag{9}$$

where the locality measure $\pi_x = exp(-D(x, x')^2/\sigma^2)$ crucially weighs the importance of training instances x' based on their distance $D(x, x')$ with instance x. This locality importance is better illustrated when reformulating Equation 9 with the surjective function $p : \mathcal{X} \rightarrow \mathcal{Z}$, where p^{-1} is to be determined and relates the generated perturbed instances $z' \in \mathcal{Z}$ to their

antecedent $x' \in \mathcal{X}$ as follows:

$$\mathcal{L}(f, g, \pi_x, p) = \sum_{z' \in \mathcal{Z}} \pi_x(p^{-1}(z')) \left(f(p^{-1}(z')) - g(z')\right)^2 \quad (10)$$

Equation 10 clearly shows that, as p^{-1} does not guarantee that neighbors in $\mathcal{Z}$ are still neighbors in the antecedent space $\mathcal{X}$, we need a mechanism to counterweight these uninteresting training samples. Earlier works [29, 33] consider binary explanation spaces, and perturbations are uniform random changes in the binary signature of the explanations. As noted before, a binary change may have a drastic impact on potential expression of the antecedents in $\mathcal{X}$, which, again, exemplifies the role of π_x in LIME-like systems. For this reason, we discuss in this paper new ways to deal with locality around and explanation instance by introducing a more gradual interpretable space and perturbation mechanism as well as a strict locality as defined by an adapted clustering algorithm.

Further properties are discussed to create its own LIME explanation algorithm in [36]. Noticeably, [36] discusses one of the key hypothesis of LIME that consists in knowing by advance the relationship between the interpretable space and original space and indicates that, whenever possible, bijective functions should be considered to limit errors when projecting from the interpretable space to the original one. This hypothesis is very strong and explains the simplifying choices that are made by [29] not to perturb instances outside already existing instances, to avoid the definition of a proper Out-Of-Sample (OOS) process that we implement in this paper. Tightly related to this problem of OOS prediction is the ability of the explanation instance representation chosen in LIME-RS to effectively capture locality via the perturbation mechanism. Indeed, one drawback of LIME-like approaches is that they sometimes fail to estimate a proper local surrogate model [21] and rather produce a model not solely focused on the explanation instance but influenced by more general trends in the data as well.

In our proposal, we want to achieve the same flexibility as LIME-RS by extending the principle of LIME algorithm [33] and to circumvent the locality problem with the introduction of two mechanisms: (1) a more gradual perturbation mechanism and its dedicated OOS prediction, and (2) a neighborhood that can possibly better capture local decision boundaries around the explanation instance.

Our paper also raises the question of the evaluation of an explanation, as reflected by the experiments and metrics that we use. This question has been raised in [7] where they define a continuum of evaluation methods from "Function-based" that relies on benchmarks and formalized evaluation metrics with low validity and cost of explanation, to "Cognition-based" where the objective is to quantify the driving factors of features that are related to the task and finally to "Application-based" that relies on experts from the domain to evaluate in a real-use case the validity of an explanation that have high validity and cost.

In [27], the authors are interested in the type of explanations that humans are able to understand and define a set of user-studies to evaluate the cost for human to understand the rationale of an explanation based on input, output of prediction model and its explanation.

In our tests, we focus on accuracy of the surrogate model (fidelity to the black-box) and relevance of the interpretable features. Other quantitative quality measures have been proposed in the literature. For example, [29] describes a fidelity measure that does not rely on an absolute rating prediction error, but rather on differences in top-k item ratings between the black-box and its surrogate model. In this paper we also compare rankings but focus on interpretable features which is more local to an explanation instance and more related to the quality and interpretability of the explanation while [29] focus on the ability of the surrogate model to mimic the black-box behavior.

Several other quantitative metrics for explanation evaluation could be considered in the context of recommender systems. Noticeably, the robustness of an explanation following the principle of locally Lipschitz continuity in the classification context seems to be a promising idea [4] that we plan to adapt to the recommendation context in the near future.

Finally, in [26] the author describes several properties of an explanation and what would make an explanation human-friendly. These concepts and ideas should be borrowed and adapted to the context of recommender systems.

6 CONCLUSION AND FUTURE WORKS

This paper introduces new implementations of locality in post-hoc explanation approach for recommender systems. Our two main contributions are: (1) the introduction of a more gradual perturbation mechanism paired with an Out-Of-Sample prediction method dedicated to Matrix Factorization recommendation, and (2) the use an off-the-shelf k-means clustering algorithm paired with a UMAP dimensionality reduction method to determine the neighborhood of each explanation instance. On overall, our approach LIRE-P performs better than the reference LIME-RS to predict top-recommendations (LIRE-P MAE is 0.89 while LIME-RS is 1.59), and provides more relevant explanations by identifying more of the expected interpretable features (relevance LIRE-P is 0.212 and NDCG@3 is 0.269 when LIME-RS in item-mode does not find any relevant features). LIRE-P takes advantage more efficiently of simulated locality in our double white-box experiment (relevance LIRE-P is 0.328). Finally, LIRE-P scales to 20M entries when the actual internal implementation (based on hot-encoding of users and items) of LIME-RS does not allow this volume of data. Our other variant LIRE-M has in some cases comparable performances with LIRE-P but at a lower complexity. Finally, future work should improve the LIRE-C variant that is not as efficient as the other 2 approaches. We also plan in a near future to extend our test to the context of a real company use case. Future research will concern the central question of the evaluation of explanation: how to evaluate the robustness of an explanation and how to be more aligned with the recommendation problem by taking into account diversity, coverage, or multi-stakeholders context that are not studied in the general classification post-hoc explanation context.

REFERENCES

[1] Abdollahi, B., and Nasraoui, O. Using explainability for constrained matrix factorization. In *Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017, Como, Italy, August 27-31, 2017* (2017), pp. 79–83.

[2] Agrawal, R., Gehrke, J., Gunopulos, D., and Raghavan, P. Automatic subspace clustering of high dimensional data for data mining applications. *SIGMOD Rec. 27*, 2 (June 1998), 94–105.

[3] Alelyani, S., Tang, J., and Liu, H. Feature selection for clustering: A review. In *Data Clustering: Algorithms and Applications*. Chapman and Hall/CRC, 2013, pp. 29–60.

[4] Alvarez-Melis, D., and Jaakkola, T. S. On the robustness of interpretability methods. *CoRR abs/1806.08049* (2018).

[5] Bokde, D. K., Girase, S., and Mukhopadhyay, D. Role of matrix factorization model in collaborative filtering algorithm: A survey. *CoRR abs/1503.07475* (2015).

[6] Boutsidis, C., Mahoney, M. W., and Drineas, P. Unsupervised feature selection for the k-means clustering problem. In *Proc. of NIPS* (2009), pp. 153–161.

[7] Doshi-Velez, F., and Kim, B. Towards a rigorous science of interpretable machine learning. *CoRR abs/1702.08608* (2017).

[8] Doshi-Velez, F., Kortz, M., Budish, R., Bavitz, C., Gershman, S., O'Brien, D., Schieber, S., Waldo, J., Weinberger, D., and Wood, A. Accountability of AI under the law: The role of explanation. *CoRR abs/1711.01134* (2017).

[9] Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. Least angle regression. *Ann. Statist. 32*, 2 (04 2004), 407–499.

[10] Ester, M., Kriegel, H., Sander, J., and Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA* (1996), pp. 226–231.

[11] Gao, J., Wang, X., Wang, Y., and Xie, X. Explainable recommendation through attentive multi-view learning. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019* (2019), pp. 3622–3629.

[12] Gedikli, F., Jannach, D., and Ge, M. How should I explain? A comparison of different explanation types for recommender systems. *Int. J. Hum.-Comput. Stud. 72*, 4 (2014), 367–382.

[13] Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. A survey of methods for explaining black box models. *ACM Comput. Surv. 51*, 5 (2019), 93:1–93:42.

[14] Hara, S., and Hayashi, K. Making tree ensembles interpretable: A bayesian model selection approach. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain* (2018), pp. 77–85.

[15] Harper, F. M., and Konstan, J. A. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst. 5*, 4 (Dec. 2015).

[16] Jain, A. K. Data clustering: 50 years beyond k-means. *Pattern Recognit. Lett. 31*, 8 (2010), 651–666.

[17] Järvelin, K., and Kekäläinen, J. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst. 20*, 4 (2002), 422–446.

[18] Koren, Y., Bell, R. M., and Volinsky, C. Matrix factorization techniques for recommender systems. *IEEE Computer 42*, 8 (2009), 30–37.

[19] Kriegel, H.-P., and Zimek, A. Subspace clustering, ensemble clustering, alternative clustering, multiview clustering: what can we learn from each other. *In Proc. ACM SIGKDD Workshop MultiClust* (2010).

[20] Kumar, V., and Minz, S. Feature selection: A literature review. *Smart CR 4*, 3 (2014), 211–229.

[21] Laugel, T., Renard, X., Lesot, M., Marsala, C., and Detyniecki, M. Defining locality for surrogates in post-hoc interpretablity. *CoRR abs/1806.07498* (2018).

[22] Li, Y., Dong, M., and Hua, J. Localized feature selection for clustering. *Pattern Recognition Letters 29*, 1 (2008), 10–18.

[23] Lipton, Z. C. The mythos of model interpretability. *Commun. ACM 61*, 10 (2018), 36–43.

[24] MacQueen, J. B. Some methods for classification and analysis of multivariate observations. In *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability* (1967), L. M. L. Cam and J. Neyman, Eds., vol. 1, University of California Press, pp. 281–297.

[25] McInnes, L., and Healy, J. UMAP: uniform manifold approximation and projection for dimension reduction. *CoRR abs/1802.03426* (2018).

[26] Molnar, C. *Interpretable Machine Learning*. 2019. https://christophm.github.io/interpretable-ml-book/.

[27] Narayanan, M., Chen, E., He, J., Kim, B., Gershman, S., and Doshi-Velez, F. How do humans understand explanations from machine learning systems? an evaluation of the human-interpretability of explanation. *CoRR abs/1802.00682* (2018).

[28] Nguyen, P., Dines, J., and Krasnodebski, J. A multi-objective learning to re-rank approach to optimize online marketplaces for multiple stakeholders. *CoRR abs/1708.00651* (2017).

[29] Nóbrega, C., and Marinho, L. B. Towards explaining recommendations through local surrogate models. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, SAC 2019, Limassol, Cyprus, April 8-12, 2019* (2019), pp. 1671–1678.

[30] Parsons, L., Haque, E., and Liu, H. Subspace clustering for high dimensional data: A review. *SIGKDD Explor. Newsl. 6*, 1 (June 2004), 90–105.

[31] Peake, G., and Wang, J. Explanation mining: Post hoc interpretability of latent factor models for recommendation systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018* (2018), pp. 2060–2069.

[32] Rendle, S. Factorization machines. In *ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010* (2010), pp. 995–1000.

[33] Ribeiro, M. T., Singh, S., and Guestrin, C. why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2016), KDD 16, Association for Computing Machinery, p. 11351144.

[34] Samek, W., Montavon, G., Vedaldi, A., Hansen, L. K., and Müller, K.-R. *Explainable AI: interpreting, explaining and visualizing deep learning*, vol. 11700. Springer Nature, 2019.

[35] Schubert, E., Sander, J., Ester, M., Kriegel, H., and Xu, X. DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN. *ACM Trans. Database Syst. 42*, 3 (2017), 19:1–19:21.

[36] Sokol, K., Hepburn, A., Santos-Rodríguez, R., and Flach, P. A. blimey: Surrogate prediction explanations beyond LIME. *CoRR abs/1910.13016* (2019).

[37] Tao, Y., Jia, Y., Wang, N., and Wang, H. The fact: Taming latent factor models for explainability with factorization trees. In *In the 42nd Int. ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, July 21-25, 2019* (2019), pp. 295–304.

[38] Tibshirani, R. Regression shrinkage and selection via the lasso. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B 58* (1994), 267–288.

[39] Tsang, M., Cheng, D., Liu, H., Feng, X., Zhou, E., and Liu, Y. Feature interaction interpretability: A case for explaining ad-recommendation systems via neural interaction detection. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020* (2020).

[40] van der Maaten, L., and Hinton, G. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research 9(Nov)* (2008), 2579–2605.

[41] Zhang, S., Yao, L., Sun, A., and Tay, Y. Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv. 52*, 1 (2019), 5:1–5:38.

[42] Zhang, Y., and Chen, X. IEEE Now Foundations and Trends, 2020.

[43] Zhang, Y., and Chen, X. Explainable recommendation: A survey and new perspectives. *Found. Trends Inf. Retr. 14*, 1 (2020), 1–101.

[44] Zhang, Y., Lai, G., Zhang, M., Zhang, Y., Liu, Y., and Ma, S. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *The 37th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, Australia - July 06 - 11, 2014* (2014), pp. 83–92.