# Context-aware feature attribution through argumentation

Jinfeng Zhong, Elsa Negre

# Context-aware feature attribution through argumentation

JINFENG ZHONG, Paris-Dauphine University, PSL Research University, CNRS UMR 7243, LAMSADE, France
ELSA NEGRE, Paris-Dauphine University, PSL Research University, CNRS UMR 7243, LAMSADE, France

Feature attribution is a fundamental task in both machine learning and data analysis, which involves determining the contribution of individual features or variables to a model's output. This process helps identify the most important features for predicting an outcome. The history of feature attribution methods can be traced back to General Additive Models (GAMs). In recent years, gradient-based methods and surrogate models have been applied to unravel complex Artificial Intelligence (AI) systems, but these methods have limitations. To address the limitations of existing methods and advance the current state-of-the-art, we define a novel feature attribution framework called **C**ontext-**A**ware **F**eature **A**ttribution **T**hrough **A**rgumentation (CA-FATA). Our framework harnesses the power of argumentation by treating each feature as an argument that can either support, attack or neutralize a prediction. Additionally, CA-FATA formulates feature attribution as an argumentation procedure, and each computation has explicit semantics, which makes it easily understandable. CA-FATA also easily integrates side information, such as users' contexts, resulting in more accurate predictions. Our experiments on two real-world datasets demonstrate that CA-FATA, or one of its variants, outperforms existing argumentation-based methods and achieves competitive performance compared to existing context-free and context-aware methods.

CCS Concepts: • **Information systems → Recommender systems**; • **Mathematics of computing → Computing most probable explanation**.

Additional Key Words and Phrases: Feature attribution, Argumentation, Explainable recommendation

**ACM Reference Format:**
Jinfeng ZHONG and Elsa NEGRE. 2023. Context-aware feature attribution through argumentation. In . ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 INTRODUCTION

[1] Feature attribution has been a long-standing practice in the field of machine learning to determine the contribution of individual features or variables to a model's overall output. This method has also been applied in recommender models to explain their behaviors [19, 24, 26, 33, 34]. The process of feature attribution helps identify the most important features for predicting an outcome and areas for model improvement. The origin of feature attribution methods can be traced back to General Additive Models (GAMs) [13]. Although GAMs are inherently interpretable, they often suffer from limited expressivity [21]. In recent years, gradient-based methods [2] have been employed to disentangle complex Artificial Intelligence (AI) systems. These methods determine the importance of a feature $x$ in a function $f$ by calculating the derivative of $f$ with respect to the feature $x$. However, gradient-based methods may struggle with simple tasks that require understanding a moderately local region [6], and interpreting such gradients can be challenging for non-experts. To address the issue of gradient-based methods, surrogate models such as LIME [26] and SHAP [19] have emerged as two prominent post-hoc explanation methods. However, the limitations of these methods have been recognized. LIME inherently suffers from stability issues [31], as for SHAP [19] the attribution of feature importance

[1] Copyright held by the authors.


Manuscript submitted to ACM

through mathematically formalizable properties (i.e., local accuracy, missingness, and consistency) may not always align with users' expectations for explanations [17].

In recent years, argumentation-based methods have gained significant attention in the field of eXplainable Artificial Intelligence (XAI) [29]. This is due to the clear and understandable means of representing relations, such as support and attack, offered by Argumentation Frameworks (AFs), which provide explicit meanings to the computation. Under AFs, decision-making processes can be visually depicted, and optimal decisions can be explained using well-defined properties [29]. Weighted arguments are used to represent the strength of arguments and the dialectical relations between them, such as support and attack. The strength function of arguments can be carefully designed to satisfy the generalized concepts of *weak balance* [24] and *weak monotonicity* [22], which characterize how arguments influence the decision-making process (we will revisit the two notions in Section 2). These methods can be used to explain decisions made through a graphical representation of the decision-making process. Context-Aware Recommender System (CARS) [1] is an important research topic in recommender systems. CARSs can model users' preferences under different contextual situations with finer granularity and generate more personalized recommendations adapted to users' contexts. We believe that context is also crucial in argumentation frameworks, as certain arguments that are considered "good" in one context may become less accurate in another context. Therefore, it is important to leverage contexts when applying argumentation [27].

In light of the interpretability challenges associated with traditional feature attribution methods, it is reasonable to explore new avenues for improving the explainability of machine learning models. Since argumentation inherently offers interpretability, one such approach is to leverage argumentation techniques to attribute feature importance. In this paper, we introduce a novel framework for feature attribution called **C**ontext-**A**ware **F**eature **A**ttribution **T**hrough **A**rgumentation (CA-FATA). The framework employs argumentation to attribute importance to each feature, considering them as arguments that can either support, attack or neutralize a prediction. CA-FATA formulates feature attribution under argumentation frameworks, providing each computation with explicit semantics, thereby ensuring interpretability. Additionally, the framework allows for the integration of side information, such as user contexts, resulting in more accurate predictions. Our experiments on two real-world datasets demonstrate that CA-FATA outperforms existing argumentation-based methods and achieves competitive performance compared to existing context-free and context-aware methods. Therefore, CA-FATA can ensure the explainability of recommendations, while not sacrificing the accuracy of prediction.

## 2 RELATED WORK

Among the existing argumentation frameworks, three types can be identified: Abstract Argumentation Framework (AAF) [10], Bipolar Argumentation Framework (BAF) [8], Tripolar Argumentation Framework (TAF) [11]. An AAF is composed of a set of pairs $< \mathcal{A}, \mathcal{R}^- >$, where $\mathcal{R}^-$ denotes a set of attack relations between arguments such that $\forall \mathbf{a}_1, \mathbf{a}_2 \in \mathcal{A}$, $(\mathbf{a}_1, \mathbf{a}_2) \in \mathcal{R}^-$ denotes that argument $\mathbf{a}_1$ attacks argument $\mathbf{a}_2$. The relation "attacks" indicates a contradiction between two arguments. For example, considering $\mathbf{a}_1$ "This user does not like the feature of this item (one actor of a movie)" and $\mathbf{a}_2$ "This item can be recommended to this user". It is evident that $\mathbf{a}_1$ attacks $\mathbf{a}_2$. BAFs contain a set of triplets, $< \mathcal{A}, \mathcal{R}^-, \mathcal{R}^+ >$, $\mathcal{R}^-$ represents attack as in AAF. Similarly, $\mathcal{R}^+$ denotes a set of support relations between arguments such that $\forall \mathbf{a}_1, \mathbf{a}_2 \in \mathcal{A}$, $(\mathbf{a}_1, \mathbf{a}_2) \in \mathcal{R}^+$ denotes that argument $\mathbf{a}_1$ supports argument $\mathbf{a}_2$. TAFs contain a set of quadruples: $< \mathcal{A}, \mathcal{R}^-, \mathcal{R}^+, \mathcal{R}^0 >$, where $\mathcal{R}^-$ represents the attack relations, $\mathcal{R}^+$ denotes the support relations and $\mathcal{R}^0$ means neutralizing relations. In this work, we have chosen to adopt TAFs that comprise three types of relations between arguments: attack, support, and neutralizing. This is because features of items may support, attack, or neutralize the

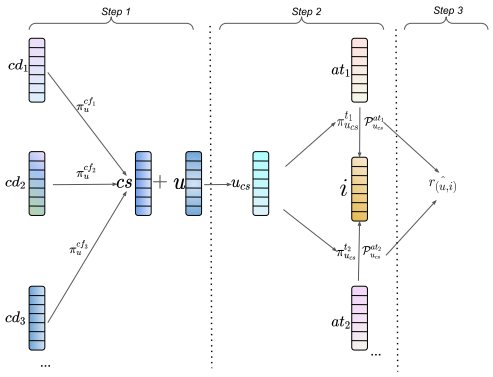



Fig. 1. Illustration of the framework of CA-FATA.

recommendation of items, indicating users' preferences towards features. As we will discuss later, the strength of the arguments in the TAFs presented in this paper is based on the users' ratings towards features.

Our research is closely related to two previous works: the *Aspect-Item framework (A-I)* introduced in [22, 24] and the *Attribute-Aware Argumentative Recommender ($A^3R$)* proposed in [33]. Both A-I and $A^3R$ use argumentation to predict users' ratings towards items, treating items and features as arguments that may attack or support each other to explain recommendations in an argumentative manner. However, these methods do not consider the influence of user contexts.

*Weak balance* and *weak monotonicity* allow for deriving intuitive explanations in an argumentative way. Essentially, the concept of *weak balance* concerns the impact of an argument on its affectees when the argument is the sole factor affecting them, while the idea of *weak monotonicity* focuses on how the potency of an argument changes when one of its affecters is silenced relative to the neutral point.

**Weak balance:** The intuition behind this notion is that if the affecter increases the strength of the affectee, then it supports the affectee. This idea has been formalized as *weak balance* in [24]. According to *weak balance*, relations under argumentation frameworks such as attacks (or supports, neutralizes) can be characterized as connections among affecters and affectees in the following way: if one affecter is isolated as the single argument that affects the affectee, then the former reduces (or increases, does not change) the latter's predicted rating with respect to the neutral point.

**Weak monotonicity :** The idea is intuitive: if the affecter supports the affectee, then muting the affecter would decrease the strength of the affectee; if the affecter attacks the affectee, then muting the affecter would increase the strength of the affectee; if the affecter neutralizes the affectee, then muting the affecter would not change the strength of the affectee. This intuition has been formalized as *weak monotonicity* in [5]. This property is formulated for two TAFs: from $< \mathcal{A}, \mathcal{R}^-, \mathcal{R}^+, \mathcal{R}^0 >$ to $< \mathcal{A}', \mathcal{R}^{-'}, \mathcal{R}^{+'}, \mathcal{R}^{0'} >$ after modifying certain arguments (e.g. muting certain features).

## 3 OUR FRAMEWORK: CA-FATA

Figure 1 depicts the structure of CA-FATA, which consists of three steps: (i) Computing the representation of target users under the target contextual situation to ensure that users' preferences are adapted to contexts and that the dialectical relations of arguments are also context-aware; (ii) Computing users' ratings towards features of items under the given contextual situation, which are then used to determine the dialectical relations; (iii) Aggregating the ratings obtained in the previous step to generate users' ratings towards items.

**Computing user representation (Step 1):** The ultimate representation of a user is determined by the user's contextual situation. In this step, our goal is to compute the representation of users that is adapted to the target

contextual situation. To achieve this, we begin by computing the importance of each contextual factor in Equation 1. The importance computed here is similar to the relevance weight in [7]. However, unlike in these two works, where the relevance weight of the context is set empirically, in our work, the importance of the context is learned in a data-driven way. Intuitively, $\pi_u^{cf}$ characterizes the extent to which user $u$ wants to take contextual factor $cf$ into account.

$$\pi_u^{cf} = \frac{exp(LeakyReLU(\langle \mathbf{u}, \mathbf{cf} \rangle))}{\sum_{cf \in C} exp(LeakyReLU(\langle \mathbf{u}, \mathbf{cf} \rangle))} \tag{1}$$

In sequence, we compute the representation of the contextual situation $cs$ by summing up all the vectors representing contextual conditions multiplied by $\pi_u^{cf}$: $\mathbf{cs} = \sum_{cd \in cs} \pi_u^{cf} \mathbf{cd}$, where $\mathbf{cs}$ is the vector that denotes contextual situation $cs$. The next step is to aggregate the representation of contextual situation $cs$ with the representation of user $u$ to obtain a specific representation of user $u$ under the contextual situation $cs$. To avoid having an excessive number of parameters[1], we sum $\mathbf{u}$ and $\mathbf{cs}$. By aggregating information from a contextual situation $cs$ and a user $u$, each user $u$ gets a specific representation $\boldsymbol{u_{cs}}$ under a contextual situation $cs$: $\boldsymbol{u_{cs}} = \mathbf{u} + \mathbf{cs}$

**Computing users' ratings towards features (Step 2):** The feature types in this paper are similar to the relations in knowledge graphs, which are directed graphs consisting of *entity-relation-entity* triplets [15]. For instance, the triplet $(HarryPotter, hasDirector, MikeNewell)$ indicates that the movie *Harry Potter* is directed by *Mike Newell*. Here, $hasDirector$ is a relation in the knowledge graph that pertains to movies, and in this paper, it corresponds to the feature type $director$. We quantify the importance of each feature type using Equation 2.

$$\pi_{u_{cs}}^{t} = \frac{exp(LeakyReLU(\langle \boldsymbol{u_{cs}}, \mathbf{at} \rangle))}{\sum_{t \in t_i} exp(LeakyReLU(\langle \boldsymbol{u_{cs}}, \mathbf{at} \rangle))} \tag{2}$$

To compute users' ratings towards features, we adopt the inner product again: $\mathcal{P}_{u_{cs}}^{at} = g(\boldsymbol{u_{cs}}, \mathbf{at})$. The representation of a user under one context differs from that under another context. As a result, the representation of user $u_{cs}$ is specific to each context, and the importance of feature type and user's rating towards features is also context-aware.

**Aggregating ratings towards features (Step 3):** After calculating the importance of each feature type and users' ratings towards each feature, $u$'s rating towards $i$ under $cs$ is:

$$\hat{r}_{(u,i)} = \sum_{t \in t_i} \pi_{u_{cs}}^{t} * \frac{\sum_{at \in at_i^t} \mathcal{P}_{u_{cs}}^{at}}{|at_i^t|} \tag{3}$$

where $t_i$ denotes all the feature types of item $i$. It should be noted that the actual value of the user $u$'s rating for item $i$ is a real number between $-1$ and 1, as defined in previous works such as [22, 24]. It is noteworthy that Equation 3 is remarkably similar to additive models, indicating that our model belongs to the family of generalized additive models. This similarity allows for easy identification of the contribution of each feature.

## 4 CONTEXT-AWARE EXPLANATIONS

Recall that the true rating $r_{(u,i)}$ is a real number between -1 and 1, then the co-domain of $\mathcal{P}_{u_{cs}}^{at}$ is also expected to be between -1 and 1. Therefore, when $\mathcal{P}_{u_{cs}}^{at} > 0$, then $\sigma(at) > 0$, indicating that $at$ is an argument that supports $rec^i$ [2]; when $\mathcal{P}_{u_{cs}}^{at} = 0$, then $\sigma(at) = 0$, indicating that $at$ is an argument that neutralizes $rec^i$; when $\mathcal{P}_{u_{cs}}^{at} < 0$, then $\sigma(at) < 0$, indicating that $at$ is an argument that attacks $rec^i$. Therefore, the TAF corresponding to a user-item interaction $(u, i)$ under $cs$ can be defined as follows:

[1]Note that other aggregation methods such as concatenation are also possible but more parameters are induced. We leave this exploration for future work.
[2]Semantically, users $u$ prefers feature $at$

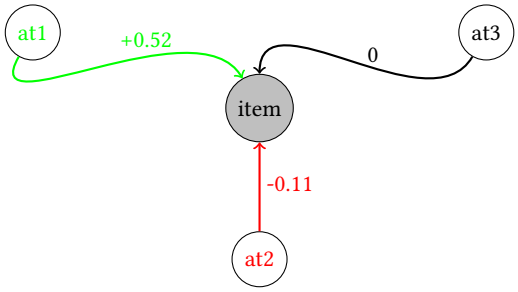



Fig. 2. A graphical representation of an argumentation procedure in a recommendation scenario. Each node represents an argument, the value on the arc denotes the strength and polarity of the argument, "+" denotes supports, "-" denotes attacks, and "0" denotes neutralizes.

Table 1. Three explanation templates for user-item interaction under contextual situation $cs = (cd_1, cd_2, cd_3, \dots)$, $SR$ denotes "strong recommendation", $WR$ denotes "weak recommendation", $NR$ "not recommended".

| Scenario | Content | Example |
|---|---|---|
| $SR$ | $at_1 = \arg\max_{at \in at_i} \mathcal{P}^{at}_{u_{cs}}$<br>$at_2 = \arg\max_{at \in at_i \setminus at_1} \mathcal{P}^{at}_{u_{cs}}$<br>$cd = \arg\max_{cd \in cs} \pi^{cf}_{u}$ | When $cd$, we recommend you this item because you like $at_1$ and $at_2$. |
| $WR$ | $at_1 = \arg\max_{at \in at_i} \mathcal{P}^{at}_{u_{cs}}$<br>$at_2 = \arg\min_{at \in at_i \setminus at_1} \mathcal{P}^{at}_{u_{cs}}$<br>$cd = \arg\max_{cd \in cs} \pi^{cf}_{u}$ | When $cd$, we recommend you this item because you like $at_1$ although you dislike $at_2$. |
| $NR$ | $at_1 = \arg\min_{at \in at_i} \mathcal{P}^{at}_{u_{cs}}$<br>$at_2 = \arg\min_{at \in at_i \setminus at_1} \mathcal{P}^{at}_{u_{cs}}$<br>$cd = \arg\max_{cd \in cs} \pi^{cf}_{u}$ | When $cd$, we do not recommend you this item because you dislike $at_1$ and $at_2$. |

*Definition 4.1.* The TAF corresponding to $(u, i)$ under $cs$ is a 4-tuple: $< \mathcal{A}, \mathcal{R}^-, \mathcal{R}^+, \mathcal{R}^0 >$ such that: $\mathcal{R}^- = \{(at, rec^i) | \mathcal{P}^{at}_{u_{cs}} < 0\}$; $\mathcal{R}^+ = \{(at, rec^i) | \mathcal{P}^{at}_{u_{cs}} > 0\}$; $\mathcal{R}^0 = \{(at, rec^i) | \mathcal{P}^{at}_{u_{cs}} = 0\}$.

According to the definition, $\mathcal{P}^{at}_{u_{cs}}$ determines the polarity of arguments: if $\mathcal{P}^{at}_{u_{cs}}$ is positive then the argument (feature) supports the recommendation of item $i$ to user $u$; if $\mathcal{P}^{at}_{u_{cs}}$ is negative then the argument (feature) attacks the recommendation of item $i$ to user $u$; if $\mathcal{P}^{at}_{u_{cs}}$ is 0 then the argument neutralizes the recommendation. By setting $\sigma(at) = \mathcal{P}^{at}_{u_{cs}}$ and $\sigma(rec^i) = \hat{r}(u, i)$, TAF corresponding to $(u, i)$ under $cs$ satisfies *weak balance* and *weak monotonicity* (Proof omitted due to lack of space). As a running example, Figure 2 presents the TAF for a user-item interaction under the contextual situation $cs$. In this TAF, each feature of the item represents an argument. The user's rating towards each feature determines the strength and polarity of the argument, thereby reflecting the user's preference. The strength of argument 1 is +0.52, indicating that the user likes feature 1 (e.g., a movie director or actor), and this feature supports the recommendation of the item to the user. The strength of argument 2 is −0.11, indicating that the user does not like feature 2 and that this feature attacks the recommendation of the item to the user. Finally, the strength of argument 3 is 0, indicating that this feature does not influence the user's rating. Note that, according to the three steps in Section 3, the prediction score could differ under different contexts, even for the same user and item. Therefore, the corresponding TAFs would also differ.

After conducting the above analyses, we propose three explanation templates in Table 1, similar to the three explanation types in [22], but with the inclusion of users' contexts. In each scenario, we select the most influential

contextual condition (as determined by Equation 1). For "strong recommendation", we propose selecting the two strongest arguments (aka. features) that support the recommendation of the item. In "weak recommendation", we propose selecting the strongest argument that supports the recommendation of the item and the strongest one that attacks the recommendation of the item. In "not recommended", we propose selecting the two strongest arguments (aka. features) that attack the recommendation of the item. Each template includes contextual information along with the corresponding arguments that either support or attack the recommendation of the item. In summary, CA-FATA is a versatile model that can be used to explain both the reasons for recommended items as well as the reasons why some items should not be recommended. Additionally, users have the flexibility to define their own templates according to their specific needs.

## 5 GENERATING RECOMMENDATIONS USING CA-FATA

Since the problem of explainable recommendations also involves generating high-quality predictions. In this section, we will conduct experiments on four real-world datasets to address the following research questions: **RQ1**, can CA-FATA achieve competitive performance compared to baseline methods? What are the advantages of CA-FATA compared to baseline methods? **RQ2**, how does context influence the performance of CA-FATA? **RQ3**, how does the importance of feature type affect the performance of CA-FATA?

### 5.1 Datasets and experiment setting

We have conducted experiments on the following real-world datasets: **Frappé:** This dataset is collected by [3]. This dataset originated from Frappé, a context-aware app recommender. There are 96 303 logs of usage from 957 users under different contextual situations, 4 082 apps are included in the dataset. Following [28], we apply log transformation to the number of interactions. As a result, the number of interactions is scaled to $0 - 4.46$. Each contextual situation is composed of 7 contextual conditions and five types of features. **Yelp:** This dataset contains users' reviews on bars and restaurants in metropolitan areas in the USA and Canada. Consistent with previous studies by [12, 35], we use the records between January 1, 2019 to December 31, 2019, which contains 904 648 observations. There are 8 contextual factors and three feature types. For the two datasets, we have adopted the 10-core setting, following [32], to ensure data quality. This means that only users with at least 10 interactions are kept.

We compare the following baselines: (i) **MF** [16]: This classic collaborative filtering method only considers user-item pairs and computes the inner product of the vectors representing users and items to make predictions. (ii) **CAMF-C** [4]: An extension of **MF** that incorporates the global influence of contexts on ratings. (iii) **FM** [25]: A strong baseline that models the second-order interactions between all information related to user-item interactions. (iv) **NeuMF** [14]: A method that combines matrix factorization and MLP (Multi-Layer Perceptron) to model the latent features of users and items. (v) **ECAM-NeuMF** [28]: An extension of **NeuMF** that integrates contextual information. Note that the authors in [28] do not release the implementation detail, for the NeuMF part, we empirically set the MLP factor size as 8, the sizes of the hidden layer as $(16, 8, 4)$, the GMF (Generalized Matrix Factorization) factor size as 16. This setting also applies to pure NeuMF. (vii) **A-I** [22–24]: An argumentation-based framework that computes users' ratings towards features, which are then aggregated to obtain the ratings towards items. Following [24][3], we set the "collaborative factor" as 0.8, 20 most similar users are selected, and all the feature importance is set as 0.1.

In these two datasets, users give explicit ratings towards items, therefore the squared loss is utilized to optimize parameters of CA-FATA: $L = \sum_{(u,i,cs)\in\mathcal{T}} (\hat{r}_{(u,i,cs)} - r_{(u,i,cs)})^2 + \lambda\|\Theta\|_2^2$, where $\mathcal{T}$ is the training set, $\hat{r}_{(u,i,cs)}$ is the

[3]For detail please refer to https://github.com/CLArg-group/KR2020-Aspect-Item-Recommender-System.

Table 2. Comparison between CA-FATA and baselines on RMSE and MAE, the second best are underlined. FATA is basically a variant of CA-FATA, the difference between CA-FATA and FATA is that FATA does not consider users' contexts and is actually the $A^3R$ [33] model. The version with "AVG" means that the importance of each feature type is set the same for all users.

| Model | | Yelp | | Frappé | |
|---|---|---|---|---|---|
| | | RMSE | MAE | RMSE | MAE |
| Context-free | MF | 1.1809 | 0.9446 | 0.8761 | 0.6470 |
| | NeuMF | 1.1710 | 0.8815 | 0.6841 | 0.5207 |
| Context-aware | FM | 1.1703 | 0.9412 | 0.7067 | 0.5796 |
| | CAMF-C | 1.1693 | 0.9241 | 0.7283 | 0.5727 |
| | LCM | 1.1687 | 0.9294 | 0.6952 | 0.5396 |
| | ECAM-NeuMF | 1.1098 | <u>0.8636</u> | 0.5599 | 0.4273 |
| Argumentation-based | A-I | 1.3978 | 1.1205 | 1.1711 | 0.9848 |
| Our propositions | FATA | 1.1434 | 0.9059 | 0.6950 | 0.5439 |
| | AVG-FATA | 1.1611 | 0.9314 | 0.6970 | 0.5461 |
| | **CA-FATA** | **1.1033** | **0.8519** | **0.5154** | **0.3910** |
| | AVG-CA-FATA | <u>1.1035</u> | 0.8637 | <u>0.5254</u> | <u>0.4025</u> |

predicted rating and $r_{(u,i,cs)}$ denotes the actual rating, $\lambda$ denotes the regularization parameter to reduce over-fitting, $\Theta$ denotes the parameters of CA-FATA. We implement CA-FATA using Pytorch[4] and we optimize the parameters using *mini-batch Adam*. The testing platform is Tesla P100-PCIE, 16GB memory in CPU. The hyper-parameters are tuned through a grid search: the learning rate is tuned on $[10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}]$; the batch size is tuned on $[128, 256, 512, 1024, 2048, 4096]$; regularization parameter is tuned on range $[5 * 10^{-5}, 10^{-4}, 5 * 10^{-3}, 10^{-3}, 10^{-2}]$. The embedding size is tuned on $[8, 16, 32, 64, 128, 256]$. Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) are selected as the primary evaluation metrics. We follow the convention established in [9, 18, 28] by splitting the datasets into a training set, a test set, and a validation set, with a ratio of $8:1:1$.

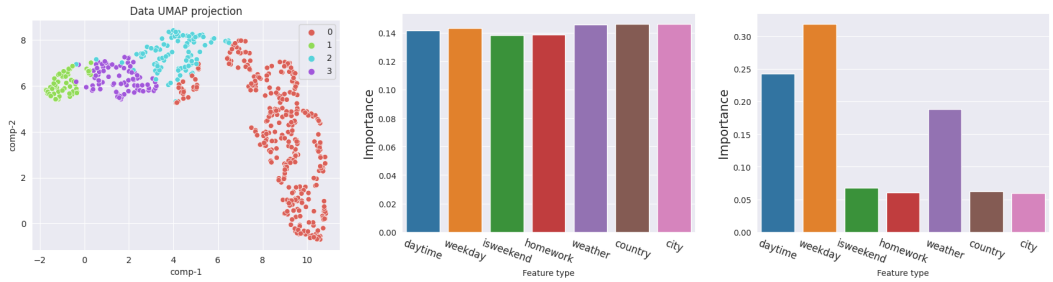
### 5.2 Rating prediction (RQ1)

Table 2 presents the results of the rating prediction experiments. We observe that CA-FATA performs better than all baselines on both the Yelp dataset and the Frappé dataset, indicating its superiority in handling complex contextual information. The following are some specific observations: (i) CA-FATA performs well on the two datasets, outperforming all baselines, demonstrating its ability to model users' preferences under different contexts. Another advantage of CA-FATA is the ability to provide argumentative explanations, which is not possible for these baselines. (ii) Compared to A-I, on Yelp, CA-FATA achieves a significant reduction in RMSE and MAE. (iii) A horizontal comparison of Frappé and Yelp datasets shows that CA-FATA performs better on Frappé than on Yelp. We attribute this difference to the sparsity of the dataset, as Yelp is still highly sparse even after applying the 10-core setting, with a sparsity of 99.84%, while Frappé has a sparsity of 94.47%. To summarize, the advantages of CA-FATA are as follows: (i) it achieves competitive performance compared to both context-free and context-aware baselines. These baselines use factorization-based methods such as MF, FM, and CAMF-C, and some combine neural networks like NeuMF and ECAM-NeuMF, which makes them difficult to interpret. On the other hand, CA-FATA provides explicit semantics for each computation and generates argumentative explanations (see Table 1 for some examples); (ii) compared to the argumentation-based method A-I, CA-FATA significantly improves prediction accuracy and generates context-aware explanations.

### 5.3 Abalation study (RQ2 and RQ3)

In order to investigate the impact of contextual factors on the performance of CA-FATA, we propose an alternative approach called FATA, which neglects user contexts, identical to the $A^3R$ model proposed in [33]. Results presented in Table 2 (**refer to rows 11 and 13**) demonstrate that CA-FATA outperforms FATA, indicating that incorporating user

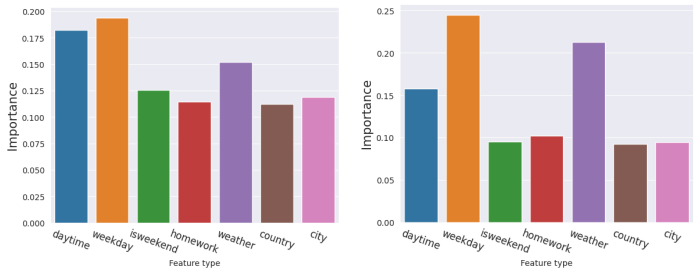
[4]Access to source code is provided in https://github.com/JinfengZh/ca-fata/tree/master




(a) Cluster of users according to importance of each contextual factor

(b) Average importance of each contextual factor of users from cluster 0

(c) Average importance of each contextual factor of users from cluster 1




(d) Average importance of each contextual factor of users from cluster 2

(e) Average importance of each contextual factor of users from cluster 3

Fig. 3. A case study on Frappé that shows the clustering of users according to the contextual factor importance learned by $CA-FATA$. The histogram shows the average importance of each contextual factor in the cluster.

contexts enables more nuanced modeling of user preferences and improves prediction accuracy. This conclusion is reinforced by the superior performance of CAMF-C over MF and ECAM-NeuMF over NeuMF. To investigate the influence of feature type importance on our proposed model's performance, we introduce AVG-CA-FATA and AVG-FATA for CA-FATA and FATA, respectively (**refer to rows 12 and 14 in Table 2**). In these models, the importance of each feature type is uniformly set for all users. For instance, in Frappé, where there are five feature types, the importance is set to 0.2 for all users, while in Yelp, where there are three feature types, the importance is set to 0.33. Results demonstrate that AVG-CA-FATA performs worse than CA-FATA, as does AVG-FATA when compared to FATA. Furthermore, comparisons between FATA, AVG-FATA, CA-FATA, and AVG-CA-FATA confirm the advantages of incorporating user contexts and modeling feature type importance across users.

To further visualize the impact of context, we represent each user by their contextual factor importance, computed using Equation 1. We use the Frappé dataset as an example, where a vector of seven dimensions represents each user: $(\pi_u^{daytime}, \pi_u^{weekday}, \pi_u^{isweekend}, \pi_u^{homework}, \pi_u^{weather}, \pi_u^{country}, \pi_u^{city})$. We apply K-means clustering for its simplicity and effectiveness [30], and find that four clusters best fit the dataset, as illustrated in Figure 3(b). We then use UMAP [20] to visualize the clustering results. Note that other dimension reduction techniques could also be used, but we choose UMAP because it can preserve the underlying information and general structure of the data. The average importance of each contextual factor in the seven clusters is shown in Figures 3(b), 3(c), 3(d), 3(e), revealing that users pay different levels of attention to contextual factors in the different clusters. Note that the same visualization applies to the Yelp data, due to limited space, we have omitted the visualization on the Yelp dataset.

## 6 CONCLUSIONS AND PERSPECTIVES

In light of the interpretability challenges associated with existing feature attribution methods, we present a novel feature attribution framework called **C**ontext-**A**ware **F**eature **A**ttribution **T**hrough **A**rgumentation (CA-FATA). CA-FATA is a feature attribution framework that treats features as arguments that can either support, attack, or neutralize a prediction using argumentation procedures. This approach provides explicit semantics to each step and allows for easy incorporation of user context to generate context-aware recommendations and explanations. The argumentation scaffolding in CA-FATA is designed to satisfy two important properties: *weak balance* and *weak monotonicity*, which highlights how features influence a prediction. These properties help identify important features and study how they influence the prediction task. We also introduce three explanation scenarios - strong recommendation, weak recommendation, and not recommended, which can be used to explain why items have been recommended or not recommended. Further investigation shows that CA-FATA can be integrated with interactive RSs, which take into account immediate user feedback to improve and adapt recommendations on the go, please refer to here for more details. Our experimental results show that CA-FATA outperforms several strong baselines regarding RMSE, MAE, highlighting its ability to provide both accuracy, too. In the future, we plan to explore the applicability of CA-FATA in other domains (e.g. under the ranking prediction scenario.) to verify its generalizability. To compute the score of contextual factor and feature type, we adopted the inner product (in Equation 2). We plan to explore other functions for this purpose. Additionally, we intend to conduct user studies to evaluate and compare the qualities of explanations generated by other explanation methods.

## REFERENCES


[1] Gediminas Adomavicius and Alexander Tuzhilin. 2011. Context-aware recommender systems. In *Recommender systems handbook*. Springer, 217–253.

[2] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. 2019. Gradient-based attribution methods. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning* (2019), 169–191.

[3] Linas Baltrunas, Karen Church, Alexandros Karatzoglou, and Nuria Oliver. 2015. Frappe: Understanding the usage and perception of mobile app recommendations in-the-wild. *preprint arXiv:1505.03014* (2015).

[4] Linas Baltrunas, Bernd Ludwig, and Francesco Ricci. 2011. Matrix factorization techniques for context aware recommendation. In *Proceedings of the fifth ACM RecSys*. 301–304.

[5] Pietro Baroni, Antonio Rago, and Francesca Toni. 2019. From fine-grained properties to broad principles for gradual argumentation: A principled spectrum. *IJAR* 105 (2019), 252–286.

[6] Blair Bilodeau, Natasha Jaques, Pang Wei Koh, and Been Kim. 2022. Impossibility Theorems for Feature Attribution. *arXiv preprint arXiv:2212.11870* (2022).

[7] Maximiliano CD Budán, Maria Laura Cobo, Diego C Martinez, and Guillermo R Simari. 2020. Proximity semantics for topic-based abstract argumentation. *Information Sciences* 508 (2020), 135–153.

[8] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. 2005. On the acceptability of arguments in bipolar argumentation frameworks. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*. Springer, 378–389.

[9] Huiyuan Chen, Xiaoting Li, Kaixiong Zhou, Xia Hu, Chin-Chia Michael Yeh, Yan Zheng, and Hao Yang. 2022. TinyKG: Memory-Efficient Training Framework for Knowledge Graph Neural Recommender Systems. In *Proceedings of the 16th ACM Conference on Recommender Systems*. 257–267.

[10] Phan Minh Dung. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence* 77, 2 (1995).

[11] Dov M Gabbay. 2016. Logical foundations for bipolar and tripolar argumentation networks: preliminary results. *Journal of Logic and Computation* 26, 1 (2016), 247–292.

[12] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems*. 299–315.

[13] Trevor J Hastie. 2017. Generalized additive models. In *Statistical models in S*. Routledge, 249–307.

[14] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[15] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. 2021. Knowledge graphs. *Synthesis Lectures on Data, Semantics, and Knowledge* 12, 2 (2021), 1–257.

[16] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[17] I Elizabeth Kumar, Suresh Venkatasubramanian, Carlos Scheidegger, and Sorelle Friedler. 2020. Problems with Shapley-value-based explanations as feature importance measures. In *International Conference on Machine Learning*. PMLR, 5491–5500.

[18] Zeyu Li, Wei Cheng, Yang Chen, Haifeng Chen, and Wei Wang. 2020. Interpretable click-through rate prediction through hierarchical attention. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 313–321.

[19] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *NeurIPS* 30 (2017).

[20] Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018).

[21] Christoph Molnar. 2020. *Interpretable machine learning*. Lulu. com.

[22] Antonio Rago, Oana Cocarascu, Christos Bechlivanidis, David Lagnado, and Francesca Toni. 2021. Argumentative explanations for interactive recommendations. *Artificial Intelligence* 296 (2021), 103506.

[23] Antonio Rago, Oana Cocarascu, Christos Bechlivanidis, and Francesca Toni. 2020. Argumentation as a framework for interactive explanations for recommendations. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, Vol. 17. 805–815.

[24] Antonio Rago, Oana Cocarascu, and Francesca Toni. 2018. Argumentation-based recommendations: Fantastic explanations and how to find them. In *Proceedings of IJCAI*. 1949–1955.

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

[26] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " 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*. 1135–1144.

[27] Juan Carlos Lionel Teze, Lluis Godo, and Guillermo Ricardo Simari. 2018. An argumentative recommendation approach based on contextual aspects. In *SUM*. Springer, 405–412.

[28] Moshe Unger, Alexander Tuzhilin, and Amit Livne. 2020. Context-aware recommendations based on deep learning frameworks. *ACM Transactions on Management Information Systems* 11, 2 (2020), 1–15.

[29] Alexandros Vassiliades, Nick Bassiliades, and Theodore Patkos. 2021. Argumentation and explainable artificial intelligence: a survey. *The Knowledge Engineering Review* 36 (2021).

[30] T Velmurugan and T Santhanam. 2010. Computational complexity between K-means and K-medoids clustering algorithms for normal and uniform distributions of data points. *Journal of computer science* 6, 3 (2010), 363.

[31] Giorgio Visani, Enrico Bagli, Federico Chesani, Alessandro Poluzzi, and Davide Capuzzo. 2022. Statistical stability indices for LIME: Obtaining reliable explanations for machine learning models. *Journal of the Operational Research Society* 73, 1 (2022), 91–101.

[32] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR*. 165–174.

[33] Jinfeng Zhong and Elsa Negre. 2022. A 3 R: Argumentative explanations for recommendations. In *2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 1–9.

[34] Jinfeng Zhong and Elsa Negre. 2022. Shap-enhanced counterfactual explanations for recommendations. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*. 1365–1372.

[35] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 1893–1902.