Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Exploiting both Vertical and Horizontal Dimensions of Feature Hierarchy for Effective Recommendation

Zhu Sun,* **Jie Yang,†** **Jie Zhang,*** **Alessandro Bozzon†**
*Nanyang Technological University, Singapore, †Delft University of Technology, The Netherlands
*{sunzhu, zhangj}@ntu.edu.sg, †{j.yang-3, a.bozzon}@tudelft.nl

Abstract

Feature hierarchy (FH) has proven to be effective to improve recommendation accuracy. Prior work mainly focuses on the influence of *vertically* affiliated features (i.e. child-parent) on user-item interactions. The relationships of *horizontally* organized features (i.e. siblings and cousins) in the hierarchy, however, has only been little investigated. We show in real-world datasets that feature relationships in horizontal dimension can help explain and further model user-item interactions. To fully exploit FH, we propose a unified recommendation framework that seamlessly incorporates both vertical and horizontal dimensions for effective recommendation. Our model further considers two types of semantically rich feature relationships in horizontal dimension, i.e. complementary and alternative relationships. Extensive validation on four real-world datasets demonstrates the superiority of our approach against the state of the art. An additional benefit of our model is to provide better interpretations of the generated recommendations.

Introduction

Feature-based recommendation has been widely studied to resolve *data sparsity* and *cold start* problems in recommender systems. Generally, features of users and items can be organized in different structures, e.g. flat or hierarchical. Feature hierarchy (FH) (He et al. 2016) – as a natural yet powerful structure to describe human knowledge – has proven to be effective to boost recommendation accuracy.

Early work incorporates FH for better recommendation by converting it to a flat structure (Ziegler, Lausen, and Schmidt-Thieme 2004; Koenigstein, Dror, and Koren 2011; Kanagal et al. 2012; Mnih 2012). The reduction to a simpler knowledge structure, while simplifying the formalization of the recommendation problem, brings severe information loss. Recently a few studies consider the structured nature of FH, assuming that items are characterized by the affiliated features in the hierarchy. Such characterization can be modeled as the *influence* of features on user-item interactions, which is either manually defined (He et al. 2016), or automatically learnt from data (Yang et al. 2016).

All the methods above consider the influence of features in vertical dimension of the hierarchy, i.e. only features with child-parent affiliation have influence on user-item interactions. They ignore, however, another important dimension of FH, i.e. the horizontal dimension. Sibling and cousin features, i.e. positioned in the same layer of the hierarchy, might capture latent relationships that could be used to better characterize user-item interactions, and, consequently, to enhance recommendation accuracy. In the following we use a running example to illustrate how the horizontal dimension of FH can help characterize user-item interactions.

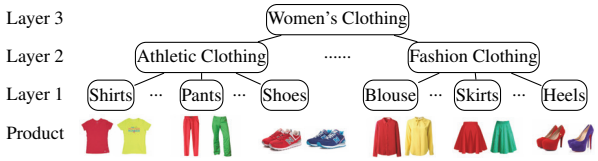

Figure 1: Running example of feature hierarchy.

Running Example Consider a Web product recommender system, where the goal is to recommend products to users. Figure 1 depicts a 3-layer category hierarchy of Women's Clothing. Categories in this hierarchy are organized in vertical dimension (e.g. Shirts and Athletic Clothing) or horizontal dimension (e.g. Shirts and Pants). Suppose a customer who prefers athletic style to fashion style. She may buy more items of Athletic Clothing, such as athletic shoes and pants to match each other, instead of items of Fashion Clothing, e.g. heels or skirts. In this case, the two sibling categories Athletic Clothing and Fashion Clothing at Layer 2 are characterized by an *alternative* relationship, as they are purchased by the user in a mutually exclusive fashion. The sibling categories at Layer 1, Athletic Shoes and Pants, are characterized by a *complementary* relationship, as they are jointly purchased by the user. Whereas the cousin categories at this layer, e.g. Athletic Shoes and Heels are *alternative* as determined by the relationship of their parent categories.

The example highlights that feature relationships in horizontal dimension can provide additional characterization of user-item interactions. It is, however, nontrivial to exploit such kind of relationships, as the vertical affiliation of features in different layers should also be preserved. As illustrated in the above example, users' preferences on items (e.g. Athletic Shoes and Heels) could also be affected by the rela-

tionships of their vertically affiliated features across different layers (e.g. Shoes - Athletic Clothing, Heels - Fashion Clothing). In other words, it is often impossible to disentangle the horizontal dimension from the vertical one.

Hence, this paper contributes a unified recommendation framework HieVH that seamlessly exploits both dimensions of FH, to boost recommendation accuracy. To model the vertial dimension, HieVH adapts latent factors of items by adding weighted aggregation of their affiliated features' latent factors, to better model item latent factors. The weights are automatically learnt from data. Horizontally, feature relationships are incorporated as regularizers at each layer of the hierarchy, to better model feature latent factors. In doing so, through the adaption of item latent factors with feature latent vectors in vertical dimension, feature relationships in horizontal dimension can be inherited by items. The result is a method that can seamlessly fuse vertical and horizontal dimensions of FH. While existing methods (e.g. ReMF (Yang et al. 2016)) consider vertical dimension, we stress it is nontrivial to extend them to integrate horizontal dimension, due to the lack of a matching mechanism in vertical dimension such as the use of feature latent factors.

Extensive experiments on four real-world datasets show that our approach achieves superior performance over state-of-the-art counterparts, with an average improvements of 5.23% on AUC. Besides, by uncovering the semantically rich feature relationships (*alternative* and *complementary*) between the recommended and rated items, HieVH provides better interpretations of the generated recommendations.

Related Work

Mapping Feature Hierarchy into Flat Generic feature-based recommendation methods, including collective matrix factorization (CMF) (Singh and Gordon 2008; Lippert et al. 2008), factorization machine (FM) (Rendle 2010; Rendle et al. 2011), and SVDFeature (Chen et al. 2012), are originally designed for incorporating features organized in a flat structure. Early methods incorporating FH (Ziegler, Lausen, and Schmidt-Thieme 2004; Weng et al. 2008) model a user's taxonomy preferences as a flat feature vector. Later, some latent factor model (LFM) based methods (Shi, Larson, and Hanjalic 2014) have been designed. For example, (Koenigstein, Dror, and Koren 2011; Mnih 2012; Lu et al. 2012; Kanagal et al. 2012) propose adding feature latent vectors into user or item latent factors. Despite this, blending FH into all the above models requires converting the hierarchy into a flat structure, thus losing the structural information encoded in the hierarchy.

Modeling the Vertical Dimension of Feature Hierarchy Menon et al. (2011) propose an ad-click prediction method that considers FH of ads. However, it assumes that an ad is conditionally independent from all higher layer features. He et al. (2016) devise a visually-aware recommendation model by manually defining the feature influence in vertical dimension of the hierarchy. Recently, Yang et al. (2016) design a recommendation method that automatically learns such influence on user/item latent factors by a parameterized regularization traversing from root to leaf features.

These studies, however, are limited to feature influence of vertical dimension, ignoring feature relationships of horizontal dimension. Besides, they are nontrivial to be extended to seamlessly integrate horizontal feature relationships, due to the lack of a matching mechanism in vertical dimension (e.g. feature latent factors). In our unified approach, we seamlessly model both dimensions of FH, and further consider semantically rich feature relationships, i.e. *alternative* and *complementary* (Mas-Colell et al. 1995; McAuley, Pandey, and Leskovec 2015).

Modeling Implicit User and Item Hierarchy Recently, Zhang et al. (2014) and Wang et al. (2015) propose to model implicit hierarchical structure within users and/or items, based on historical user-item interactions. Our work differs from these two models, in that we consider leveraging *explicit* FH to guide the learning of latent factors.

Measuring Feature Influence and Relationships

This section first introduces our metrics for measuring feature influence in vertical dimension, and feature relationships in horizontal dimension of FH. To demonstrate the need for richer feature hierarchy characterization of user-item interactions for better recommendation, we then apply the proposed metrics to analyze Amazon Web store data.

Let $\mathcal{U}, \mathcal{I}$ denote the set of users and items, and $\mathcal{F}$ denote the set of features organized in a hierarchy. r_{ui} is the rating given by user $u \in \mathcal{U}$ to item $i \in \mathcal{I}$. Each item $i \in \mathcal{I}$ is affiliated with a subset of features $\mathcal{F}(i) = \{f_i^1, f_i^2, \ldots, f_i^L\}$, organized as a path from leaf feature f_i^1 to root feature f_i^L. Let $P(e_i)$ denote the probability of the event that an item i is rated by a user, defined as,

$$P(e_i) = \frac{|\{u|u \in \mathcal{U}, r_{ui} \neq 0\}|}{|\mathcal{U}|}$$

Based on this definition, we use *item co-occurrence* IC to measure the closeness of two items.

Definition 1 (Item Co-occurrence).

$$\text{IC}(i,j) = \frac{P(e_i \cap e_j)}{P(e_i) \times P(e_j)}$$

where $P(e_i \cap e_j)$ is the joint probability of the event that both items i and j are rated by a user.

The IC measure can be used to define both feature influence in vertical dimension, and feature relationships in horizontal dimension of the hierarchy, as illustrated below.

Definition 2 (Feature Influence of Vertical Dimension). *Given the items $i_1, i_2, \ldots$ characterized by a same subset of feature path $\mathcal{F}(i_1) = \mathcal{F}(i_2) = \ldots = \{f^1, \ldots, f^L\}$, the influence of an arbitrary feature f^l ($1 \leq l \leq L$) in the path on these items is defined as the following vector,*

$$\overrightarrow{\text{FI}}(f^l) = \frac{1}{|f^l| - 1}\Big[\sum_{j \in f^l, i_1 \neq j} \text{IC}(i_1, j), \sum_{j \in f^l, i_2 \neq j} \text{IC}(i_2, j), \ldots\Big]$$

where each element in the vector is the average IC between the target item and all the other items affiliated to the feature. This definition allows us to test the difference among the influence of features in the same path.

We then define feature relationships in horizontal dimension, based on item relationships formalized as follows.

Definition 3 (Item Relationships). *Items i, j are alternative if $P(e_i|e_j) < P(e_i)$ and $P(e_j|e_i) < P(e_j)$; they are complementary if $P(e_i|e_j) > P(e_i)$ and $P(e_j|e_i) > P(e_j)$.*

Two items i and j are therefore *alternative*, if the probability of i being rated given j is rated (e.g. $P(e_i|e_j)$), is lower than that without knowing whether j is rated or not (e.g. $P(e_i)$). Contrarily, they are complementary if the former is larger.

We now turn to the quantification of item relationships, which will be used later for measuring feature relationships. It turns out that, IC can be a proper metric for measuring item relationships, according to the following theorem.

Theorem 1 (Item Relationships Measured by IC).

$$\begin{array}{lcl} \text{Items } i \text{ and } j \text{ are alternative} & \Longleftrightarrow & \text{IC} < 1 \\ \text{Items } i \text{ and } j \text{ are independent} & \Longleftrightarrow & \text{IC} = 1 \\ \text{Items } i \text{ and } j \text{ are complementary} & \Longleftrightarrow & \text{IC} > 1 \end{array}$$

A Smaller value of IC (< 1) *indicates a stronger alternative relationship between items i and j; vice versa, a larger value of* IC (> 1) *indicates a stronger complementary relationship between items i and j.*

Proof. Using the relationship between joint and conditional probability, $P(e_i \cap e_j) = P(e_j|e_i) \times P(e_i)$, we have

$$\text{IC}(i,j) = \frac{P(e_i \cap e_j)}{P(e_i) \times P(e_j)} = \frac{P(e_j|e_i) \times P(e_i)}{P(e_i) \times P(e_j)} = \frac{P(e_j|e_i)}{P(e_j)}$$

Similarly, with $P(e_i \cap e_j) = P(e_i|e_j) \times P(e_j)$, we have $\text{IC}(i,j) = \frac{P(e_i|e_j)}{P(e_i)}$. Thus, we can see that if $\text{IC}(i,j) < 1$, then $P(e_j|e_i) < P(e_j)$ and $P(e_i|e_j) < P(e_i)$, vise versa, suggesting an *alternative* relationship between items i and j is equivalent to $\text{IC}(i,j) < 1$. A smaller value of IC would indicate a larger gap between $P(e_j|e_i)$ and $P(e_j)$, $P(e_i|e_j)$ and $P(e_i)$, i.e. a stronger *alternative* relationship; the opposite also holds, i.e. a stronger *alternative* relationship indicates a smaller value of IC. If $\text{IC}(i,j) > 1$, then $P(e_j|e_i) > P(e_j)$ and $P(e_i|e_j) > P(e_i)$, vice versa, suggesting a *complementary* relationship between items i and j is equivalent to $\text{IC}(i,j) > 1$. A larger IC indicates a stronger *complementary* relationship; the opposite also holds. Similarly, if $\text{IC}(i,j) = 1$, then $P(e_j|e_i) = P(e_j)$ and $P(e_i|e_j) = P(e_i)$, vice versa, hence items i, j are *independent* and $\text{IC}(i,j) = 1$ are equivalent. □

The independence between two items provides no additional characterization of user preferences, thus it is neither beneficial for recommendation. We do not consider it particularly useful before; it will be, however, properly handled by our proposed method. With the above metric for measuring item relationships, we now propose the metric for feature relationships in horizontal dimension.

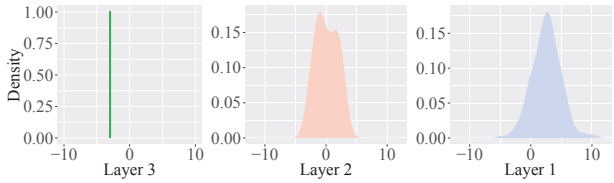

Figure 2: Distribution of feature relationships (log-scaled, i.e. $x = log(\text{FR})$) at 3 layers of the hierarchy.

Definition 4 (Feature Relationships in Horizontal Dimension). *The relationship of two features f and g is given by*

$$\text{FR}(f,g) = \frac{1}{|f| \times |g|} \sum\nolimits_{i \in f} \sum\nolimits_{j \in g} \text{IC}(i,j)$$

FR is defined as the average IC between all pairs of items, where the two items in each pair are characterized respectively by the two features. Similar to IC, $\text{FR}(f,g) < 1, > 1, = 1$ indicate that features f and g are *alternative*, *complementary* and independent, respectively.

Feature Influence and Relationships in Real-world Data We now show the presence of feature influence and relationships in the Amazon Web store data – Clothing, Shoes & Jewelry. The details of the dataset are deferred to the Experimental Results section. For demonstration purpose, we only show the results of the top-3 layers of the hierarchy.

The hierarchy contains 116 categories at Layer 1, thus 116 paths in vertical dimension. Comparing the influence of features in the same path, we find that 74.33% of feature influence at Layer 1 is significantly larger than that at Layer 2, and that 72.57% of the influence at Layer 2 is significantly larger than that at Layer 3 (Paired t-test, p-value < 0.01).

The root layer (Layer 3) contains two features, Women's Clothing and Men's Clothing. It can be observed from Figure 2 that the two features have an *alternative* relationship, indicating that women and men clothing are generally not purchased together. For Layer 2, we observe that the feature relationships are evenly distributed on the side $log(\text{FR}) < 0$ and $log(\text{FR}) > 0$, indicating that both *alternative* and *complementary* feature relationships exist at this layer. An example of *complementary* features is Woman Watches, Man Watches, suggesting that women and men watches are usually purchased together, e.g. for couples, despite of the fact that women's clothing and men's clothing are *alternative*. When looking at feature relationships at Layer 1, we can see that the relationships among most features are *complementary*, e.g. women active clothing and women athletic shoes. Overall, as a general trend, more *complementary* relationships can be observed in lower layers than upper layers, suggesting that customers tend to buy items characterized differently by fine-grained features to match each other.

The HieVH Framework

This section describes the HieVH framework – that seamlessly exploits both vertical and horizontal dimensions of FH to boost recommendation performance.

The Basic Recommendation Model Our method is built on the latent factor model (LFM), where each user and item in the high dimensional user-item interaction space are mapped into a low dimensional space. We generalize the basic LFM to seamlessly integrate both vertical and horizontal dimensions of FH by minimizing the following equation:

$$\mathcal{J} = \overbrace{\sum_{o_{ui}\neq 0} C\big(r_{ui}, <\theta_u, \overline{\theta}_i>\big)}^{\text{cost function}} + \overbrace{\alpha \sum_{f,g\in\mathcal{F}} \Psi(\theta_f, \theta_g) + \Omega(\Theta)}^{\text{regularizers}}$$

where $o_{ui} = 1$ if user u rates item i, otherwise 0; $\theta_u, \theta_i, \theta_f \in \mathbb{R}^d$ are the latent factors of user u, item i and feature f, respectively; d is the dimension of latent factors; $C(\cdot)$ is a convex cost function (e.g. quadratic function) measuring the difference between the real rating r_{ui} and the predicted rating, i.e., the inner product of θ_u and $\overline{\theta}_i$; and $\overline{\theta}_i = \Phi(\theta_i, \theta_f)$ is the adaptive item latent factor considering the influence of features in vertical dimension on item latent factors through function Φ; Ψ is the regularization function to constrain the difference between θ_f and θ_g based on the relationships among features in horizontal dimension; α controls the importance of Ψ; $\Omega(\Theta)$ with $\Theta = \{\lambda, \theta_u, \theta_i, \theta_f\}$ are regularizers to avoid over-fitting; λ is the regularization hyperparameter. The main challenge is how to effectively formulate the functions Φ, Ψ by integrating the influence and relationships of features in the two dimensions of FH.

Modeling Vertical Dimension Features are vertically affiliated in the hierarchy. Based on the results shown in the previous section, we observe that an item i is characterized by all the affiliated features $\mathcal{F}(i) = \{f_i^1, f_i^2, \ldots, f_i^L\}$, organized as a path in the hierarchy with different degrees. Hence, we formulate the function $\Phi(\theta_i, \theta_f)$ to adapt the latent factor of item i, i.e., θ_i, by adding to it the latent factors of its affiliated features, i.e., $\mathcal{F}(i)$ in the hierarchy, given by:

$$\begin{aligned}\overline{\theta}_i &= \Phi(\theta_i, \theta_f, \vartheta_f) \\ &= \theta_i + \big[\vartheta_{f^1}, \vartheta_{f^2}, \cdots, \vartheta_{f^L}\big] \begin{bmatrix} -\theta_{f^1}- \\ -\theta_{f^2}- \\ \cdots \\ -\theta_{f^L}- \end{bmatrix}_{L\times d}\end{aligned}$$

where $\vartheta_{\mathcal{F}(i)} = [\vartheta_{f^1}, \vartheta_{f^2}, \ldots, \vartheta_{f^L}]$ is the parameter vector, indicating the different influence of features in $\mathcal{F}(i)$ on item i. It can be automatically learnt by our model. θ_{f^l} $(1 \le l \le L)$ is the latent vector of feature $f^l \in \mathcal{F}(i)$.

In this equation, any items, e.g. i and j, that belong to the same feature set, i.e., $\mathcal{F}(i) = \mathcal{F}(j)$, share the same parameter vector, i.e., $\vartheta_{\mathcal{F}(i)} = \vartheta_{\mathcal{F}(j)} = [\vartheta_{f^1}, \vartheta_{f^2}, \ldots, \vartheta_{f^L}]$. That is to say, the features organized in a same path influence all items belonging to the leaf feature in that path. In this way, we reduce the number of parameters and avoid over-fitting. The number of parameter vectors is the total number of the unduplicated feature paths in FH, which is equal to the size of leaf feature set. Note that, in the adaptive function Φ, a good estimation of feature latent factors is essential to accurately adapt item latent factors, which can be facilitated by considering horizontal dimension of FH, as given below.

Modeling Horizontal Dimension From the perspective of horizontal dimension, features are organized as siblings or cousins at the same layer of the hierarchy. Data analysis on real-world data shows the presence of two types of feature relationships, i.e., *alternative* and *complementary*, which are highly useful to better model feature latent factors.

Hence, we incorporate such kind of feature relationships by assuming that in each Layer l $(1 \le l \le L)$ of the hierarchy: if two features are *alternative*, then the distance of their latent factors should be large; if *complementary*, the distance of their latent factors should be small. Based on the above assumption, we devise the following regularizer to better model feature latent factors,

$$\Psi(\theta_f, \theta_g) = \sum\nolimits_{l=1}^{L} \sum\nolimits_{f,g\in\mathcal{F}^l, f<g} \sigma_{fg} \|\theta_f - \theta_g\|_F^2$$

where $\mathcal{F}^l$ is the feature set at Layer l of the hierarchy; $\sigma_{fg} = log(\text{FR}(f,g))$ with $\sigma_{fg} < 0, > 0, = 0$ indicating f, g are *alternative*, *complementary* and *independent*, respectively. Through adaptive function Φ adding latent vectors of affiliated features to their items' latent factors, feature relationships in horizontal dimension are inherited by items. Consequently, the better estimated feature latent factors can more accurately adapt item latent factors.

Similarly, we also incorporate item relationships to help better model item latent factors by assuming that if two items are *alternative*, the distance of their latent factors should be large; if *complementary*, it should be small. Based on this, we design the following regularizer,

$$\Psi(\theta_f, \theta_g; \theta_i, \theta_j) = \Psi(\theta_f, \theta_g) + \sum\nolimits_{i,j\in\mathcal{I}, i<j} \sigma_{ij} \|\theta_i - \theta_j\|_F^2$$

where $\sigma_{ij} = log(\text{IC}(i,j))$ with $\sigma_{ij} < 0, > 0, = 0$ indicating items i, j are alternative, complementary and independent.

Note that σ_{fg}, σ_{ij} seamlessly accommodate our assumptions illustrated below: if two features f, g are alternative, then we have FR < 1, thus $\sigma_{fg} < 0$. In this case, minimizing Ψ leads to large distance between θ_f and θ_g; if f, g are complementary, then we have FR > 1, thus $\sigma_{fg} > 0$. In this case, minimizing Ψ leads to small distance between θ_f and θ_g; if f, g are independent, then FR $= 1$, thus $\sigma_{fg} = 0$. In this case, the independent feature relationships are not considered in Ψ. σ_{ij} holds similar properties as σ_{fg}.

Once the feature and item relationships are incorporated into the objective function $\mathcal{J}$, we can more accurately model feature and item latent factors in function Φ, thus can ultimately better model user-item interactions.

Remark HieVH seamlessly integrates the modeling of both vertical and horizontal dimensions of FH. Though in this paper we focus on item FH, HieVH can as well accommodate user FH. It is noteworthy to remark how HieVH is able to handle arbitrarily imbalanced FH, thus making its application suitable to a wide variety of application scenarios. Specifically, it first determines the depth of a feature as the number of layers from the root feature to this feature in a top-down fashion. Then, the features that have the same depth are on the same layer.

Algorithm 1: HieVH Optimization Process

Input: rating matrix $\mathbf{R}$, feature hierarchy $\mathcal{F}$, $d, \alpha, \lambda, \gamma, Iter$

1 Initialize θ, ϑ with small values;
2 $L \leftarrow$ the highest layer of $\mathcal{F}$;
// Parameter update for $\mathcal{J}$
3 **for** $t = 1; t \leq Iter; t++$ **do**
4 **foreach** $u \in \mathcal{U}, i \in \mathcal{I}$ **do**
5 $\theta_u^{(t)} \leftarrow \theta_u^{(t-1)} - \gamma \nabla \mathcal{J}(\theta_u)$;
6 $\theta_i^{(t)} \leftarrow \theta_i^{(t-1)} - \gamma \nabla \mathcal{J}(\theta_i)$;
7 **for** $l = 1; l \leq L; l++$ **do**
8 **foreach** $f \in \{\mathcal{F}^l \cap \mathcal{F}(i)\}$ **do**
9 $\theta_f^{(t)} \leftarrow \theta_f^{(t-1)} - \gamma \nabla \mathcal{J}(\theta_f)$;
10 $\vartheta_f^{(t)} \leftarrow \vartheta_f^{(t-1)} - \gamma \nabla \mathcal{J}(\vartheta_f)$;
11 **if** $\mathcal{J}$ *has converged* **then**
12 break;

Optimization We adopt the widely used stochastic gradient descent method to optimize HieVH. The update rules of all the variables are given by the following equations. The optimization process is shown in Algorithm 1, which is mainly composed of parameter update (line 3-12).

$$\nabla \mathcal{J}(\theta_u) = \sum\nolimits_{i \in \mathcal{I}} o_{ui} \left(<\theta_u, \overline{\theta}_i> - r_{ui} \right) \overline{\theta}_i + \lambda \theta_u$$

$$\nabla \mathcal{J}(\theta_i) = \sum_{u \in \mathcal{U}} o_{ui} \left(<\theta_u, \overline{\theta}_i> - r_{ui} \right) \theta_u + \lambda \theta_i + \alpha \sum_{i,j \in \mathcal{I}, i<j} \sigma_{ij} (\theta_i - \theta_j)$$

$$\forall f \in \mathcal{F}^l, l = \{1, 2, \cdots, L\},$$

$$\nabla \mathcal{J}(\theta_f) = \sum\nolimits_{u \in \mathcal{U}} \sum\nolimits_{i \in \mathcal{I}, f \in \mathcal{F}(i)} \vartheta_f o_{ui} \left(<\theta_u, \overline{\theta}_i> - r_{ui} \right) \theta_u + \lambda \theta_f + \alpha \sum\nolimits_{f,g \in \mathcal{F}^l, f<g} \sigma_{fg} (\theta_f - \theta_g)$$

$$\nabla \mathcal{J}(\vartheta_f) = \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}, f \in \mathcal{F}(i)} o_{ui} \left(<\theta_u, \overline{\theta}_i> - r_{ui} \right) <\theta_u, \theta_f> + \lambda \vartheta_f$$

Complexity Analysis The computational time is mainly taken by evaluating the objective function $\mathcal{J}$ and updating the relevant variables. The time to compute $\mathcal{J}$ is $\mathcal{O}(d|\mathbf{R}| + dn^2)$, where $|\mathbf{R}|$ is the number of non-zero observations in the rating matrix $\mathbf{R}$, and n is the number of items. For the gradients $\nabla \mathcal{J}(\theta_u), \nabla \mathcal{J}(\theta_i), \nabla \mathcal{J}(\theta_f), \nabla \mathcal{J}(\vartheta_f)$, the computational time are $\mathcal{O}(d|\mathbf{R}|)$, $\mathcal{O}\left(d|\mathbf{R}| + d\frac{n(n-1)}{2}\right)$, $\mathcal{O}\left(d|\mathcal{F}||\overline{\mathbf{R}}| + d\sum_{l=1}^{L} \frac{|\overline{\mathcal{F}}|(|\overline{\mathcal{F}}|-1)}{2}\right)$, $\mathcal{O}(L|\mathcal{F}^1||\overline{\mathbf{R}}|)$, respectively. Wherein $|\mathcal{F}|$ is the total number of features in the hierarchy; $|\overline{\mathbf{R}}|$ is the average number of ratings under each feature; $|\overline{\mathcal{F}}|$ is the average number of features at each layer of the hierarchy. Generally due to $L < |\overline{\mathcal{F}}| < |\mathcal{F}^1| < |\mathcal{F}| \ll n$ and $|\overline{\mathbf{R}}| < |\mathbf{R}|$, the overall computational complexity of Algorithm 1 is $(Iter \times \mathcal{O}(d|\mathbf{R}| + dn^2))$. In summary, our proposed framework is scalable to large datasets.

Experimental Results

Datasets To validate HieVH, we use the Amazon Web store dataset (McAuley et al. 2015). This dataset has recently been applied for evaluating recommendation methods incorporating FH (He et al. 2016; Yang et al. 2016). Similar to these work, we consider the Clothing Shoes & Jewelry dataset; to evaluate the generalizability of HieVH, we further consider three other datasets in different domains, including Electronics, CDs & Vinyl, and Home & Kitchen. The FHs of all the datasets are imbalanced. We uniformly sample the datasets to balance their sizes for cross-dataset comparison. Table 1 reports the statistics of the datasets.

Figure 3: The impact of parameter α.

Table 1: Descriptive statistics of the datasets.

Data	#users	#items	#ratings	#features	#layers
Cloth.	36,000	42,201	60,141	2,764	7
Elect.	43,234	38,766	77,962	1,292	6
C. & V.	33,868	36,320	71,872	1,293	6
H. & K.	44,519	37,445	73,820	2,002	5

Comparison Methods We compare with six state-of-the-art algorithms, 1) MF (Salakhutdinov and Mnih 2007): matrix factorization model; 2) CMF (Singh and Gordon 2008): collective MF; 3) FM (Rendle 2010): factorization machine; 4) TaxMF (Chen et al. 2012; Koenigstein, Dror, and Koren 2011): taxonomy based MF ; 5) Sherlock (He et al. 2016): visually-aware model ; 6) ReMF (Yang et al. 2016): recursive regularization based MF. Methods 2-3 only utilize features in FH without considering the structure. Methods 4-6 are all based on FH. Besides, three variants of our proposed framework are compared. A) HieV: only considers vertical feature influence; B) HieVC: exploits vertical feature influence and horizontal *complementary* feature relationship; C) HieVH: fuses both vertical feature influence and horizontal *complementary* & *alternative* feature relationships.

Evaluation Standard 5-fold cross validation is adopted to evaluate all the methods. The Area Under the ROC Curve (AUC) is used as the evaluation metric. Larger AUC indicates better recommendation performance.

Parameter Settings Optimal parameter settings have been empirically estimated. We set $d = 10$ and apply a grid search in $\{0.001, 0.01, 0.1\}$ for γ, λ and 1/2-way regulariza-

tion of FM; $\alpha = 0.5, 0.01$ for CMF and ReMF, respectively; for Sherlock, we use the same settings as (He et al. 2016).

Impact of α In HieVH, α controls the importance of feature relationships in the horizontal dimension of FH. We apply grid search in $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$ to investigate the impact of α on recommendation performance. Results are shown in Figure 3. As α varies from small to large, the performance first increases then decreases, with the maximum reached at the range $[10^{-3}, 10^{-2}]$. The performance variations across datasets suggest the need for dataset-specific settings; the similarity in performance variation across α values demonstrates the robustness of HieVH.

Comparative Results Table 2 summarizes the performance of all comparison methods across all datasets, where two views are created for each dataset: 'All Users' indicates all users are considered in the test data; while 'Cold Start' indicates only users with ≤ 5 ratings are involved in the test data. Several interesting observations can be noted.

Compared with all other methods incorporating FH, MF considering no auxiliary information performs the worst, indicating the effectiveness of feature based recommendation. The methods originally designed for the flat feature structure, including CMF and FM, generally perform worse than the FH based methods (TaxMF, Sherlock and ReMF). Since FH needs to be converted into a flat structure when applied into CMF and FM, the result demonstrates that useful information is lost in the conversion. FM outperforms CMF and even some FH based methods. This could be explained by FM further considering user-feature interactions, in addition to the user-item and item-feature interactions, as in CMF.

Among the three state-of-the-art FH based methods, Sherlock performs better than TaxMF, but worse than ReMF. The reason behind is that TaxMF views the influence of features in different layers of FH identically, whereas Sherlock weights the influence of features in different layers differently. However, the weights are defined manually. In contrast, ReMF automatically learns such influence by a parameterized regularization traversing from root to leaf features.

We now compare the three variants of our framework – HieV, HieVC and HieVH, with the recently proposed ReMF. By considering vertical feature influence only, HieV performs slightly better than ReMF. The possible explanation is that in HieV, item latent factors are directly adapted by the affiliated feature latent factors; whereas in ReMF, latent factors of items are regularized by those of items that share common ancestor features, which means items are indirectly influenced by their affiliated features. In other words, the adaption of item latent factors in HieV is more straightforward than that in ReMF, thus more effective. HieVC upgrades HieV by adding *complementary* feature relationships in the horizontal dimension; HieVC is then promoted to HieVH by further incorporating *alternative* feature relationships. In results, HieVC performs better than HieV, but worse than HieVH, implying that both *complementary* and *alternative* feature relationships among horizontally organized features help improve recommendation accuracy.

Overall, when compared with all the other comparison methods across all the datasets, HieVH achieves the best performance. The improvements w.r.t. all users and cold start are 5.23%, 5.11% on average, respectively, which are statistically significant (Paired t-test, p-value < 0.001). This implies that the recommendation performance can be further enhanced by appropriately considering both vertical and horizontal dimensions of feature hierarchy.

Figure 4: The example of recommendations.

Interpretations by HieVH We now analyze how the incorporation of feature relationships can better explain user-item interactions. To this end, we first derive for each user the feature relationships between the rated items (i.e. training data), and the correctly recommended items (i.e. intersection between recommended items and test data). We calculate the percentage of complementarity Cp and alternativity Ap among these relationships for each user. Good recommendations would result in a high percentage of complementary items and low percentage of alternative items.

Table 3 shows the average Cp and Ap for all users in the test data at the top-3 layers of the Clothing hierarchy (Layer 3 excluded since only an alternative relationship exists). We could see that from ReMF, HieV to HieVC, HieVH, with *complementary* and *alternative* feature relationships considered, Cp increases, and Ap decreases in both layers. Among all the methods, HieVH achieves the highest Cp, and lowest Ap, with significant improvements over HieVC (Paired t-test, p-value < 0.01). This clearly indicates that by incorporating the two types of feature relationships, the recommendations better approximate real user preferences. Example users to whom the recommendations benefit from feature relationships generated by HieVH are shown in Figure 4. The recommendations to users u_1 and u_2 are better because of the *complementarity* among fashion clothing that u_1 is more fond of, and among athletic clothing that u_2 instead is more fond of, and the *alternativity* between fashion and athletic clothing. Similarly, the recommendations for u_3 are provided because of her interests in fashion clothing and lingerie. For user u_4, it is discovered that he likes clothing collocation, i.e. the *complementarity* of the items he purchased.

Conclusions

Feature hierarchy is known to enhance recommendation performance. Existing methods only consider feature influence in vertical dimension, ignoring feature relationships in horizontal dimension. In this paper, we first show the presence of feature influence and relationships in real-world datasets based on our proposed metrics. Then we design the HieVH to seamlessly exploit both the vertical and horizontal dimensions of feature hierarchy for better recommendation. Exper-

Table 2: Performance (AUC) of comparison methods. The best performance is highlighted in bold; the second best performance of other methods is marked by '*'; 'Improve' indicates the relative improvements that HieVH achieves w.r.t. the '*' results.

Datasets	Cases	MF	CMF	FM	TaxMF	Sherlock	ReMF	HieV	HieVC	HieVH	Improve
Clothing.	All Users	0.5455	0.5646	0.6826	0.6509	0.6747	0.7015*	0.7160	0.7291	**0.7375**	5.13%
	Cold Start	0.5426	0.5667	0.6629	0.6493	0.6702	0.7032*	0.7124	0.7284	**0.7352**	4.55%
Electronic.	All Users	0.5555	0.5762	0.6839	0.6569	0.6915	0.7337*	0.7512	0.7672	**0.7748**	5.60%
	Cold Start	0.5526	0.5735	0.6831	0.6475	0.6982	0.7305*	0.7474	0.7658	**0.7741**	5.97%
C.& V.	All Users	0.5478	0.5622	0.6356	0.6905	0.7082	0.7249*	0.7328	0.7516	**0.7600**	4.84%
	Cold Start	0.5433	0.5609	0.6231	0.6881	0.7076	0.7243*	0.7315	0.7514	**0.7588**	4.76%
H. & K.	All Users	0.5420	0.5545	0.6938	0.6469	0.6938	0.7279*	0.7456	0.7574	**0.7667**	5.33%
	Cold Start	0.5395	0.5562	0.6915	0.6511	0.6973	0.7275*	0.7412	0.7554	**0.7650**	5.15%

Table 3: Cp and Ap of the Clothing hierarchy.

Approaches	Layer 1		Layer 2	
	Cp	Ap	Cp	Ap
ReMF	87.62%	12.38%	75.23%	24.76%
HieV	88.89%	11.11%	76.19%	23.89%
HieVC	92.62%	7.38%	84.62%	15.38%
HieVH	95.65%	4.35%	89.35%	10.65%

imental results on four real-world datasets show that HieVH consistently outperforms state-of-the-art methods. Besides, HieVH provides better interpretations of the generated recommendations. Our future work will focus on the integration of knowledge bases, which contain feature hierarchy as meta data for describing items, as well as extra information of item relationships, to further improve recommendation.

Acknowledgments

This work is supported by the SIMTech-NTU Joint Laboratory on Complex Systems. This work is partially funded by the Social Urban Data Lab (SUDL) of the Amsterdam Institute for Advanced Metropolitan Solutions (AMS).

References

Chen, T.; Zhang, W.; Lu, Q.; Chen, K.; Zheng, Z.; and Yu, Y. 2012. Svdfeature: a toolkit for feature-based collaborative filtering. *JMLR* 13(12):3619–3622.

He, R.; Lin, C.; Wang, J.; and McAuley, J. 2016. Sherlock: sparse hierarchical embeddings for visually-aware one-class collaborative filtering. In *IJCAI*.

Kanagal, B.; Ahmed, A.; Pandey, S.; Josifovski, V.; Yuan, J.; and Garcia-Pueyo, L. 2012. Supercharging recommender systems using taxonomies for learning user purchase behavior. *VLDB Endowment* 5(10):956–967.

Koenigstein, N.; Dror, G.; and Koren, Y. 2011. Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy. In *RecSys*.

Lippert, C.; Weber, S. H.; Huang, Y.; Tresp, V.; Schubert, M.; and Kriegel, H.-P. 2008. Relation prediction in multi-relational domains using matrix factorization. In *NIPS*.

Lu, K.; Zhang, G.; Li, R.; Zhang, S.; and Wang, B. 2012. Exploiting and exploring hierarchical structure in music recommendation. In *AIRS*.

Mas-Colell, A.; Whinston, M. D.; Green, J. R.; et al. 1995. *Microeconomic theory*, volume 1.

McAuley, J.; Targett, C.; Shi, Q.; and van den Hengel, A. 2015. Image-based recommendations on styles and substitutes. In *SIGIR*.

McAuley, J.; Pandey, R.; and Leskovec, J. 2015. Inferring networks of substitutable and complementary products. In *KDD*.

Menon, A. K.; Chitrapura, K.-P.; Garg, S.; Agarwal, D.; and Kota, N. 2011. Response prediction using collaborative filtering with hierarchies and side-information. In *KDD*.

Mnih, A. 2012. Taxonomy-informed latent factor models for implicit feedback. In *KDD Cup*.

Rendle, S.; Gantner, Z.; Freudenthaler, C.; and Schmidt-Thieme, L. 2011. Fast context-aware recommendations with factorization machines. In *SIGIR*.

Rendle, S. 2010. Factorization machines. In *ICDM*.

Salakhutdinov, R., and Mnih, A. 2007. Probabilistic matrix factorization. In *NIPS*.

Shi, Y.; Larson, M.; and Hanjalic, A. 2014. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys* 47(1):3.

Singh, A. P., and Gordon, G. J. 2008. Relational learning via collective matrix factorization. In *KDD*.

Wang, S.; Tang, J.; Wang, Y.; and Liu, H. 2015. Exploring implicit hierarchical structures for recommender systems. In *IJCAI*.

Weng, L.-T.; Xu, Y.; Li, Y.; and Nayak, R. 2008. Exploiting item taxonomy for solving cold-start problem in recommendation making. In *ICTAI*.

Yang, J.; Sun, Z.; Bozzon, A.; and Zhang, J. 2016. Learning hierarchical feature influence for recommendation by recursive regularization. In *RecSys*.

Zhang, Y.; Ahmed, A.; Josifovski, V.; and Smola, A. 2014. Taxonomy discovery for personalized recommendation. In *WSDM*.

Ziegler, C.-N.; Lausen, G.; and Schmidt-Thieme, L. 2004. Taxonomy-driven computation of product recommendations. In *CIKM*.