# Hierarchical Visual-aware Minimax Ranking Based on Co-purchase Data for Personalized Recommendation

Xiaoya Chong
xychong2-c@my.cityu.edu.hk
City University of Hong Kong
Hong Kong, China

Qing Li
csqli@comp.polyu.edu.hk
The Hong Kong Polytechnic University
Hong Kong, China

Howard Leung
howard@cityu.edu.hk
City University of Hong Kong
Hong Kong, China

Qianhui Men
qianhumen2-c@my.cityu.edu.hk
City University of Hong Kong
Hong Kong, China

Xianjin Chao
xjchao2-c@my.cityu.edu.hk
City University of Hong Kong
Hong Kong, China

## ABSTRACT

Personalized recommendation aims at ranking a set of items according to the learnt preferences of the user. Existing methods optimize the ranking function by considering an item that the user has not bought yet as a negative item and assuming that the user prefers the positive item that he has bought to the negative item. The strategy is to exclude irrelevant items from the dataset to narrow down the set of potential positive items to improve ranking accuracy. It conflicts with the goal of recommendation from the seller's point of view, which aims to enlarge that set for each user. In this paper, we diminish this limitation by proposing a novel learning method called Hierarchical Visual-aware Minimax Ranking (H-VMMR), in which a new concept of predictive sampling is proposed to sample items in a close relationship with the positive items (e.g., substitutes, compliments). We set up the problem by maximizing the preference discrepancy between positive and negative items, as well as minimizing the gap between positive and predictive items based on visual features. We also build a hierarchical learning model based on co-purchase data to solve the data sparsity problem. Our method is able to enlarge the set of potential positive items as well as true negative items during ranking. The experimental results show that our H-VMMR outperforms the state-of-the-art learning methods.

## CCS CONCEPTS

• **Information systems → Learning to rank**; *Collaborative filtering*; • **Theory of computation → Bayesian analysis**.

## KEYWORDS

Recommender Systems, Personalized Ranking, Visual Features

**ACM Reference Format:**
Xiaoya Chong, Qing Li, Howard Leung, Qianhui Men, and Xianjin Chao. 2020. Hierarchical Visual-aware Minimax Ranking Based on Co-purchase Data for Personalized Recommendation . In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3366423.3380007

## 1 INTRODUCTION

Personalized recommendation is an important task in recommender systems, which aims to rank a set of items and return the top ones to the user. It benefits both sellers and buyers. Sellers can gain revenue increase with precision marketing, while users spend less time finding their interested ones from a tremendous amount of items.

The key of personalized ranking is to distinguish potential positive items (items the user may like but has not bought yet) from true negative items (items the user dislikes) for each user with observed feedbacks (e.g., purchases, clicks). The best known methods for personalized ranking include point-wise learning such as Collaborative Filtering (CF) and Factorization Machines (FM) [24], as well as pair-wise learning such as Bayesian Personalized Ranking (BPR) [25]. Matrix Factorization (MF) [20] computes the preference score between a user and an item, which is often adopted by both learning methods. Point-wise learning commonly makes use of item similarity computation and regression. Pair-wise learning proposes the concept of negative sampling and aims at maximizing the probability that the user prefers a positive item to a negative one. It usually outperforms point-wise learning since it is directly optimized for ranking. However, in this way, the system can only distinguish the items with properties that the user dislikes. The ranking strategy of BPR is to enlarge the set of negative items instead of positive items, which is in contradiction with the commercial goal of recommender systems.

In this paper, we propose a new concept of predictive sampling to complement the negative sampling method. For predictive sampling, we sample an item from the set of substitutes and compliments of the positive item. For negative sampling, we first remove the items closely related to the positive item, then we select a negative item from the remaining items and calculate the item quality by computing its visual distance from the positive item. Predictive item is to explore potential properties that the user may like while negative item is used to rule out properties that the user is not interested in. We adopt a minimax strategy to make a balance between these two processes. In recommender systems, users only give feedbacks to

*WWW '20, April 20–24, 2020, Taipei, Taiwan*

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380007

a limited number of items. To solve the data sparsity problem, we build a hierarchical learning model based on co-purchase data to augment the training samples. Our ranking strategy takes both the seller's and buyer's demands into considerations.

The contributions of our work are:

1. We propose the concept of predictive sampling and present a novel minimax ranking method to maximize individual preference difference between positive and negative items, as well as minimize such difference between positive and predictive items.

2. We further extend our method by building a hierarchical learning architecture based on co-purchase data to augment the training samples. We apply our method to three state-of-the-art pair-wise learning methods: BPR [25], VBPR [13] and DVBPR [19].

3. We conduct the experiments on Amazon dataset of Clothes, Shoes and Jewelry introduced in [22]. We use Area Under the ROC Curve (AUC), Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) to evaluate our model. The experimental results show that our model outperforms other models under different settings.

## 2 RELATED WORKS

Point-wise learning methods such as Collaborative Filtering (CF) have achieved great success for recommendations in the past. [24] optimizes CF by introducing factorized parameters which are able to model various interactions between variables even with sparse data. Point-wise learning is a regression method and is not directly optimized for ranking.

Under pair-wise learning, BPR proposed in [25] is the state-of-the-art model for personalized ranking problem. It samples a negative item for each observed user-item pair in the dataset and assumes that the user's preference degree for a positive item is higher than a negative item. Many works have been done for optimizing BPR. Some people focus on reforming negative sampling to decrease the pair-wise loss. [27, 28] use a generator ($G$) to sample informative negative items for the discriminator ($D$) to learn and adopt Reinforcement Learning to train $G$ based on the reward given by $D$. Others including [1, 7, 11, 30] improve it by using heterogeneous information such as content (e.g., tags, topics) and context (e.g., time, place). Lately, [2] combines Generative Adversarial Networks (GAN) [8] with CF to solve the data sparsity problem by rating augmentation. [14] adopts GAN to generate adversarial perturbations for user and item embeddings to improve the robustness of BPR. Pair-wise learning only considers negative items and has difficulties in finding potential positive items.

Deep learning has also proved to be effective in recommender systems especially with massive data. In [4], the authors propose to train the linear model together with Deep Neural Network (DNN) to capture complex interactions between variables. [9, 29] further improve it by replacing the linear model with FM or a cross network to remove the need of manual feature engineering. [15] uses DNN to replace the inner product of CF, and [26] proposes a user-item/item-item distance-based DNN with attention layer to capture non-linear relationships of users and items. As a black box model, DNN lacks interpretability.

A recent research trend in recommender systems is to make it visual-aware. Some focus on item classification and recommend items by retrieving similar items without considering the user's purchase history. [5, 22] predict which item goes well with another by learning visual distances between items. [18] optimizes clothes segmentation method for real-world images with the help of pose estimation. Only considering visual features is not enough for making recommendations when the users pay more attention to other aspects of the item such as functionality and price. In [13], the authors propose visual-aware BPR (VBPR), which uses both user's feedbacks and visual features. They extract visual features from item images at pixel level by CaffeNet [17], and add a visual interaction term to extend the MF function. [6, 10, 12, 21, 23, 31] further improve visual-aware model by utilizing other information (e.g., social-temporal dynamics, users' reviews), while [19] combines CNN with VBPR to remove the need of pre-extracted visual features. Their methods rely on additional information or networks to supervise the model for performance improvement, which are different from our work of adopting a novel minimax ranking strategy to optimize the learning method.

## 3 PERSONALIZED RANKING

### 3.1 Problem Formulation

Let $U = \{u_1,u_2,...,u_{|U|}\}$ be the set of users and $I = \{i_1,i_2,...,i_{|I|}\}$ be the set of items. The user-item interaction $(u,i) \in S$ ($S \subseteq U \times I$) denotes the purchase/click behavior, which is also known as implicit feedback. The items a user has bought/clicked in the past are positive items. The items a user has not bought/clicked may be true negative items the user does not like or potential positive items the user may like.

We also define $I_u^+$ to denote the set of items to which user $u$ has expressed positive feedback and $U_i^+$ to denote the set of users who give positive feedback to item $i$.

### 3.2 Preliminaries

A common practice to do personalized ranking is first defining a preference score $\hat{x}_{u,i}(\Theta)$, which denotes user $u$'s preference degree for item $i$. For brevity, we use $\hat{x}_{u,i}$ for $\hat{x}_{u,i}(\Theta)$. A common Matrix Factorization (MF) method to define $\hat{x}_{u,i}$ is:

$$\hat{x}_{u,i} = \alpha_i + < \beta_u, \beta_i >$$

where $\alpha_i$ is the item bias term, and $\beta_u$ and $\beta_i$ are user $u$'s and item $i$'s latent factors. We regard $\beta_i$ as the properties of item $i$, while $\beta_u$ as user $u$'s preferences towards $\beta_i$. '<, >' denotes the inner product of two vectors.

Visual-aware models add a new visual interaction pair to MF:

$$\hat{x}_{u,i} = \alpha_i + < \beta_u, \beta_i > + < v_u, v_i >$$

where $v_i$ represents the visual properties of item $i$, and $v_u$ represents user $u$'s preferences towards $v_i$.

Point-wise learning usually uses regression models to learn $\hat{x}_{u,i}$ directly. Pair-wise learning requires sampling a negative item $j$ for each observed $(u, i)$ interaction. For user $u$, it aims to maximize the probability that $u$ prefers $i$ to $j$ :

$$P(i > j|u, \Theta_{u,i,j}) = \sigma(\hat{\delta}_{uij}) = \sigma(\hat{x}_{u,i} - \hat{x}_{u,j}) \quad (1)$$

where $\sigma(\cdot)$ function can be logistic sigmod function, heaviside function, etc.

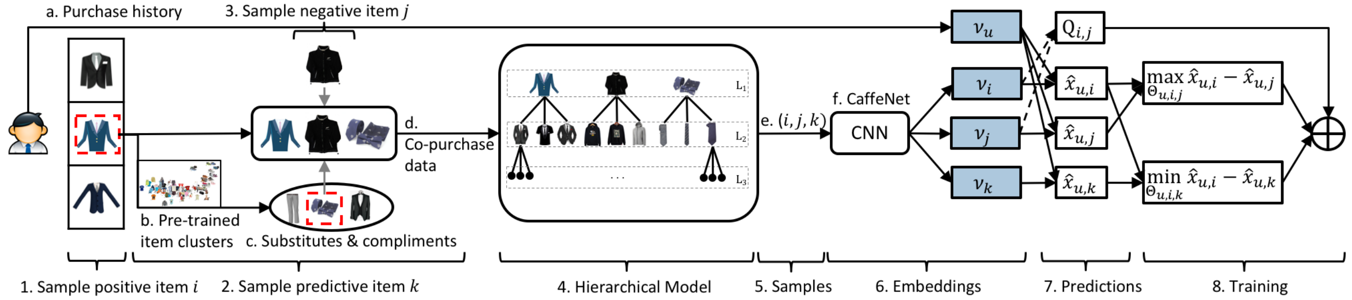

a. Purchase history
3. Sample negative item $j$
$v_u$
$Q_{i,j}$
$v_i$
$\hat{x}_{u,i}$
$\max_{\Theta_{u,i,j}} \hat{x}_{u,i} - \hat{x}_{u,j}$
d. Co-purchase data
$L_1$
$L_2$
$L_3$
f. CaffeNet
e. $(i,j,k)$
CNN
$v_j$
$\hat{x}_{u,j}$
$v_k$
$\hat{x}_{u,k}$
$\min_{\Theta_{u,i,k}} \hat{x}_{u,i} - \hat{x}_{u,k}$
b. Pre-trained item clusters
c. Substitutes & compliments
1. Sample positive item $i$
2. Sample predictive item $k$
4. Hierarchical Model
5. Samples
6. Embeddings
7. Predictions
8. Training

**Figure 1: Overview of our H-VMMR. Number 1-8 denote the process order of our model, and letter a-f denote different data or networks utilized by our model. In the hierarchical model, we have three tree structures for positive item $i$, negative item $j$ and predictive item $k$ respectively. Each tree has $L$ layers, and the nodes in the first layer $L_1$ are root nodes.**

## 4 METHODOLOGY

We create a training data $S_{train} : U \times I \times I \times I$:

$$S_{train} = \left\{(u, i, j, k) | i \in I_u^+ \wedge j \in I \setminus I_u^+ \wedge k \in I \setminus I_u^+ \wedge j \neq k\right\}$$

The semantic of a 4-tuple $(u, i, j, k)$ is that user $u$ is assumed to prefer positive item $i$ over negative item $j$ and like predictive item $k$ as much as $i$. The three sets of $i$, $j$ and $k$ are pairwise disjoint for each $u$. The interaction between $i$ and $j$ will indicate the latent properties that $u$ may dislike, while the interaction between $i$ and $k$ will help the model discover the positive properties that in $u$'s favor. In this section, we explain the details of our sampling method to get $j$ and $k$, then we introduce our ranking algorithm. Figure 1 shows the overview of our model.

### 4.1 Sampling

**Predictive Sampling.** Human naturally create associations between various items. These associations lead to different item collocations, which can be found in items from the same category (known as substitutes), as well as from totally different ones (known as compliments).

If this information is available in the dataset, we can make use of it to sample $k$. If not, the method proposed in [22] can be used for finding related items. The distance of two items $i$ and $k$ is computed using $l^2$-norm squared:

$$d_{i,k} = \|v_i - v_k\|_2^2 \tag{2}$$

where $v$ represents the visual features of a certain item.

The probability term $P(r_{ik} \in R)$ is used to define whether $i$ and $k$ have a relationship:

$$P(r_{ik} \in R) = \sigma_c(-d_{i,k}) = \frac{1}{1 + e^{d_{i,k} - c}} \tag{3}$$

where $\sigma_c(\cdot)$ is the sigmoid function with an offset $c$.

**Negative Sampling.** We first exclude all the items which have relationships with the positive item $i$ from the dataset $I$, and then sample a negative item $j$ from the remaining items.

We assume that there exists an ideal visual distance $\eta$ for $(i, j)$ pair. If $d_{i,j}$ is smaller or larger than $\eta$, $(i, j)$ will provide too little or too much information for our model to learn. More specifically, when $d_{i,j}$ is large, $i$ and $j$ may have no common properties. Though we know $u$ prefers $i$ to $j$, we do not know which item property leads to $u$'s dislike for $j$. Hence $j$ provides too much information for our model, and vice versa. We propose to use weighted negative sampling based on the visual distance between items to reduce the risk of negative sampling. We define the quality $Q$ of each $(i, j)$ pair:

$$Q_{i,j} = g(d_{i,j}) = e^{-\frac{(d_{i,j} - \eta)^2}{2\sigma_1^2}} \tag{4}$$

where $\eta$ and $\sigma_1$ are hyper parameters to tune during training.

### 4.2 Minimax Optimization Criterion

Let $P = \{p_1, p_2, ..., p_l\}$ denotes the $l$ properties in each item. Intuitively, each property $p \in P$ can be regarded as category (e.g., clothes, shoes), color (e.g., black, white), texture (e.g., coarse, velvety), etc. Positive property subset $P^+ \subseteq P$ denotes properties which are preferred by the user, and the negative property subset $P^- \subseteq P$ denotes properties which the user is not interested in. Our goal is to find $P^+$ and $P^-$ for each user. Once we find that, we can give priority to the items which contain $p \in P^+$ and at the same time avoid items with $p \in P^-$.

To find $P^- \subseteq P$, Eq. 1 is used. By maximizing Eq. 1, we decrease user u's preferences for each property $p \in P_j - P_i$, which is regarded as a possible negative property $\in P^-$ for $u$. Once we get $P^-$, we push items with $p \in P^-$ back to the bottom of the ranking order of $u$, which corresponds to the exclusion strategy.

$P^+ \subseteq P$ can be obtained in a similar way. Since $k$ has a close relationship with $i$, we assume that user $u$ has a great chance of buying $k$ in the future. Hence $u$ should have a high preference score for $k$ as well. We define the difference of user $u$'s preference between $i$ and $k$ as $\hat{\delta}_{uik}$, and $\hat{\delta}_{uik}$ should be close to zero according to the above assumption.

We use $P(i > k|u, \Theta_{u,i,k})$ to measure the probability that $u$ prefers $i$ to $k$. And $\Theta_{u,i,k}$ can be optimized by:

$$\Theta^*_{u,i,k} = arg \min_{\Theta_{u,i,k}} \sum_{(u,i,k) \in S_{train}} P(i > k|u, \Theta_{u,i,k}) \tag{5}$$

where $\Theta_{u,i,k} \subseteq \Theta$ denotes parameters related to $u$, $i$, $k$. Note that $\Theta_{u,i,j} \vee \Theta_{u,i,k} = \Theta$.

For ease of calculation, we define Eq. 5 as:

$$\Theta^*_{u,i,k} = arg \max_{\Theta_{u,i,k}} \sum_{(u,i,k) \in S_{train}} g(\hat{\delta}_{uik}) \tag{6}$$

**Table 1: Dataset statistics (after processing)**

| Dataset | Setting | Category | #users | #items | #feedbacks |
|---|---|---|---|---|---|
| Men | all | multi | 16,243 | 73,784 | 120,023 |
| | | shoes | 2,017 | 8,381 | 13,253 |
| | cold | multi | 6,689 | 33,493 | 38,206 |
| | | shoes | 714 | 4,405 | 4,829 |
| Women | all | multi | 59,312 | 272,590 | 503,771 |
| | | shoes | 10,229 | 49,359 | 79,419 |
| | cold | multi | 7,141 | 39,066 | 40,213 |
| | | shoes | 6,282 | 29,192 | 37,290 |

where $\mu$ in $g(\cdot)$ is set to zero, so that when $\hat{\delta}_{uik} \rightarrow 0$, $g(\hat{\delta}_{uik})$ can get its maximal value.

By Eq. 5, our model learns that each property $p \in P_k - P_i$ does not affect the user u's preference for an item, which indicates that $u$ also likes $p$ and $p$ becomes an element of set $P^+$ for $u$. By adopting this strategy, we can acquire the positive property set $P^+$ for every user. We then pop up items with $p \in P^+$ to the top of the ranking order. So far one round of the selection procedure is completed.

To balance the exclusion and selection methods, we define our objective function as a minimax equilibrium:

$$\underset{\Theta_{u,i,k}}{min}\ \underset{\Theta_{u,i,j}}{max} \sum_{(u,i,j,k)\in S_{train}} p(i > j|u, \Theta_{u,i,j}) + p(i > k|u, \Theta_{u,i,k}) \quad (7)$$

Using maximum posterior estimator (MAP) for Eq. 7, we can derive the following optimization criterion:

$$\Theta^* = arg\,\underset{\Theta}{max} \sum_{(u,i,j,k)\in S_{train}} \omega_- log(\sigma(\hat{\delta}_{uij})) + \omega_+ log(g(\hat{\delta}_{uik})) \quad (8)$$
$$-\lambda_\Theta \left\|\Theta\right\|^2$$

where $\omega_-$ and $\omega_+$ are added to weight the two terms. $\lambda_\Theta$ is used for regularization.

We adopt stochastic gradient descent (SGD) to learn our model since it is a good choice for deriving the maximum value of a function. Given a random sampled 4-tuple $(u, i, j, k)$, an update is performed:

$$\Theta \leftarrow \Theta + \alpha * (\omega_- * \frac{1}{1 + e^{\hat{\delta}_{uij}}} * \frac{\partial \hat{\delta}_{uij}}{\partial \Theta} - \omega_+ * \frac{1}{\sigma_2^2} * \hat{\delta}_{uik} * \frac{\partial \hat{\delta}_{uik}}{\partial \Theta} - \lambda_\Theta * \Theta) \quad (9)$$

where $\alpha$ is the learning rate. We assign $Q_{i,j}$ to $\omega_-$ to ensure that the influence of the first term depends on the quality of $(i, j)$ pair. And $\omega_+$ is chosen during hyper parameter tuning.

## 4.3 Hierarchical Learning Model

We can further extend our model by using hierarchical learning. We use $N(i)$ to denote the items co-purchased with $i$. The idea is that if we know that $u$ prefers $i$ to $j$, then we infer that $u$ prefers $i \in N(i)$ to $j \in N(j)$. In layer $L$, there are $(|N(i)|^{L-1})^3$ 4-tuple $(u, i, j, k)$. We feed these training samples into our model layer by layer. As the layers become deeper, we have less confidence in our inferences, hence the learning rate will decay to ensure a smaller scale of weight adjustments when involving deeper layers. The learning rate for layer $L$ is $\alpha\gamma^{L-1}$, with the attenuation coefficient $\gamma \in (0, 1)$. In the following, we use width $W$ to denote the number of child nodes of each item and depth $D$ to denote the number of layers.

**Table 2: AUC on test set. The higher the AUC, the better the model. Under 'Setting', 'all' evaluates the overall accuracy while 'cold' evaluates the model performance in the item cold-start setting. Under 'Category', 'multi' denotes the Clothing, Shoes and Jewelry dataset, while 'shoes' only contains a single category of Shoes items.**

| Dataset | Men | | | | Women | | | |
|---|---|---|---|---|---|---|---|---|
| Setting | all | | cold | | all | | cold | |
| Category | multi | shoes | multi | shoes | multi | shoes | multi | shoes |
| a. WR-MF | 0.4453 | 0.4583 | 0.3230 | 0.3600 | 0.4555 | 0.4014 | 0.2980 | 0.3919 |
| b. BPR | 0.5125 | 0.5183 | 0.4881 | 0.4983 | 0.5640 | 0.4961 | 0.3920 | 0.4824 |
| c. H-BPR | 0.6236 | 0.6044 | 0.4474 | 0.4181 | 0.6650 | 0.6203 | 0.3920 | 0.4491 |
| d. MMR | 0.5131 | 0.5199 | 0.4980 | **0.5098** | 0.5648 | 0.4962 | 0.4830 | 0.4824 |
| e. H-MMR | **0.6393** | **0.6258** | **0.5114** | 0.4875 | **0.6822** | **0.6395** | 0.4960 | **0.5499** |
| f. IRGAN | 0.5001 | 0.5096 | 0.4975 | 0.5070 | 0.5525 | 0.4999 | **0.4998** | 0.4935 |
| g. APR | 0.5027 | 0.5090 | 0.5017 | 0.4987 | 0.5620 | 0.5015 | 0.4983 | 0.4977 |
| Impv e vs. b | 24.74% | 20.74% | 4.77% | -2.16% | 20.95% | 28.90% | 26.53% | 13.99% |
| h. VBPR | 0.7242 | 0.6989 | 0.7065 | 0.6603 | 0.7389 | 0.7032 | 0.7157 | 0.6610 |
| i. H-VBPR | 0.7388 | 0.7283 | 0.7310 | 0.6861 | 0.7583 | 0.7284 | 0.7220 | 0.6924 |
| j. VMMR | 0.7418 | 0.7330 | 0.7450 | 0.7158 | 0.7493 | 0.7453 | 0.7639 | 0.7255 |
| k. H-VMMR | **0.7582** | **0.7516** | **0.7622** | **0.7508** | **0.7613** | **0.7635** | **0.7670** | **0.7435** |
| Impv j vs. g | 4.69% | 7.54% | 7.88% | 13.70% | 3.03% | 8.57% | 7.16% | 12.48% |
| l. DVBPR | 0.7207 | 0.7198 | **0.7128** | 0.6984 | 0.7482 | **0.7919** | 0.7525 | 0.7588 |
| m. DVMMR | **0.7234** | **0.7234** | 0.7061 | **0.7179** | **0.7578** | 0.7911 | **0.7556** | **0.7619** |
| Impv m vs. l | 0.37% | 0.50% | -0.93% | 2.79% | 1.28% | -0.10% | 0.41% | 0.40% |
| Impv k vs. l | 5.20 % | 4.41% | 6.93% | 7.50% | 1.75% | -3.58% | 1.50% | -2.41% |

# 5 EXPERIMENTS

We conduct experiments on Amazon dataset to evaluate our model. Firstly, we introduce our dataset and evaluation metrics. Then we introduce some baseline models to be compared with our model. Finally, we show the experimental results quantitatively.

## 5.1 Dataset

We use the dataset from Amazon.com introduced by [22]. We select two datasets, Men and Women datasets from the Clothing, Shoes and Jewelry category for experiments. We remove cold users with $|I_u^+| < 5$. We also remove users with $|I_u^+| > 100$ and items with $|U_i^+|$>100 to increase sparsity. For Men and Women datasets, we use all items and cold items sub-datasets separately. In cold items setting, we only keep items with $|U_i^+| <= 5$. Meanwhile, we do experiments for single category (Shoes) and multi-category (Clothing, Shoes and Jewelry) items in both all and cold items sub-datasets. Statistics of our selected datasets are shown in Table 1.

## 5.2 Visual Features

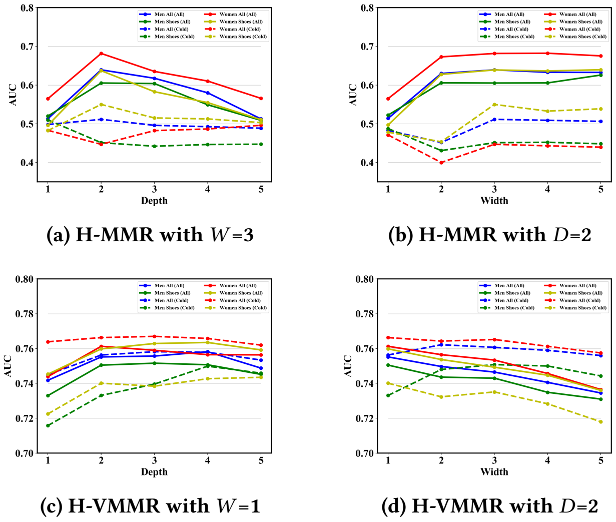
The visual features are retrieved from CaffeNet by [22]. The model they use is composed of 5 convolutional layers and 3 fully-connected layers, which has been pre-trained on 1.2 million ImageNet (ILSVRC2010) images. They use the output of the second fully-connected layer (FC7). For each image, $F$=4096 dimensional features are extracted at pixel-level.

For deep learning models, image photos instead of visual features are used as model inputs.

## 5.3 Evaluation Metrics

Leave-one-out evaluation is used in our experiments. For each user, we randomly select one user-item interaction for test and validation, and others are used for training.

Our model is learned from $S_{train}$. We use $S_{validation}$ to detect parameter overfitting. Early stop is adopted once we detect overfitting. The results are evaluated on $S_{test}$ by Area Under the



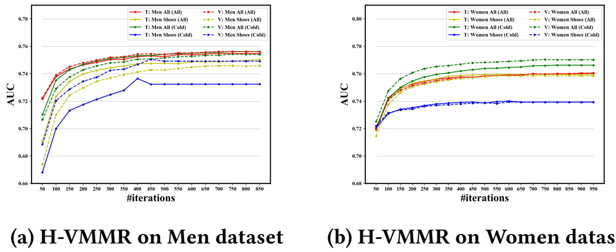
(a) H-MMR with $W$=3

(b) H-MMR with $D$=2

(c) H-VMMR with $W$=1

(d) H-VMMR with $D$=2

**Figure 2: AUC of H-MMR and H-VMMR with different depth ($D$) and width ($W$) on 8 datasets.**



(a) H-VMMR on Men dataset

(b) H-VMMR on Women dataset

**Figure 3: AUC of H-VMMR with different training iterations on Men and Women datasets. 'T' evaluates AUC on test set while 'V' evaluates AUC on validation set.**

ROC Curve (AUC), Hit Ratio (HR@N) and Normalized Discounted Cumulative Gain (NDCG@N).

## 5.4 Comparison Models

Six state-of-the-art MF-based models from existing work are used for comparison:

**WR-MF** [16]: It is a point-wise learning model. It uses the inner product of a user-factor and an item-factor to measure the user's preference score for each item. Least-square optimization with regularization is used for learning the model.

**BPR** [25]: It is a pair-wise learning model. It samples negative items and assumes that the user prefers a positive item over a negative item. It uses MF to compute the user's preference for an item and MAP to derive the objective function.

**H-BPR**: Extends BPR by our hierarchical learning.

**MMR**: Our minimax ranking without visual features or $Q_{i,j}$.

**H-MMR**: Extends MMR by our hierarchical learning.

**IRGAN** [27] : It trains a generative retrieval model $G$ and a discriminative retrieval model $D$ iteratively to optimize each other. $G$ is used to retrieve difficult examples for $D$ to learn as well as learn model parameters, while $D$ is used to classify whether an item is positive or negative for a user.

**APR** [14]: It generates adversarial perturbations for user and item embeddings to maximize the objective function of BPR. While BPR tries to defend the adversarial model and aims at minimizing the objective function. By adversarial training, the robustness of BPR can be improved.

**VBPR** [13]: It is a visual-aware pair-wise learning model. It optimizes the MF function of BPR by adding another visual interaction term.

**H-VBPR**: Extends VBPR by our hierarchical learning.

**VMMR**: Our minimax ranking with visual features.

**H-VMMR**: Extends VMMR by our hierarchical learning.

**DVBPR** [19]: This model combines deep learning with BPR framework. It uses CNN-F [3] to extract visual features from items and trains CNN-F and BPR jointly.

**DVMMR**: Combines CNN-F with our minimax ranking.

## 5.5 Discussion

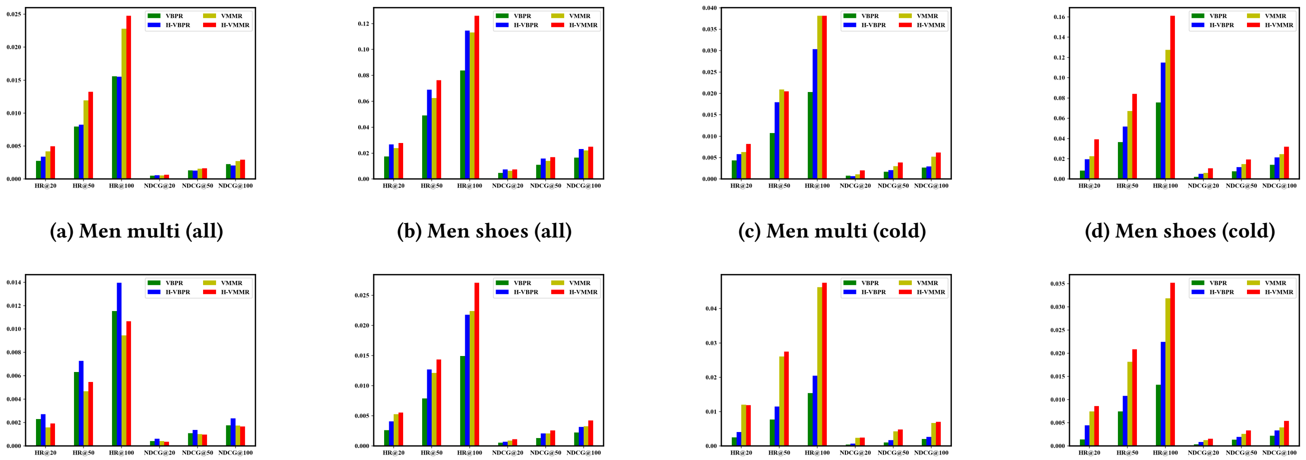
We analyze the performance of our model under different settings.

**AUC.** We combine our model with BPR, VBPR and DVBPR and show the model performances in terms of AUC in Table 2. It can be concluded that non visual-aware models (row *a-g*) are not comparable to visual-aware models (row *h-m*) in both all and cold items settings on Clothing, Shoes and Jewelry dataset. Deep learning models (row *l-m*) do not outperform traditional machine learning methods (row *h-k*) in most cases. One explanation is that DVBPR trains CNN and BPR at the same time, which means both the visual representations of the item and the learnt preferences of the user are adjusted according to the objective function. VBPR uses pre-extracted visual features and it only needs to adjust the learnt preferences of the user during learning. Once the objective function becomes more complex, it becomes difficult for DVBPR to learn because it has much more variables compared with VBPR. To conclude, our H-MMR outperforms BPR by 17.30%, H-VMMR outperforms VBPR by 8.13% and DVMMR outperforms DVBPR by 0.59% across all datasets.

**Hierarchical learning.** Figure 2 shows that H-MMR has the highest AUC with $D$=2 and $W$=3. For H-VMMR, a depth of 3 layers is slightly better than 2 layers but the difference is little. For ease of computation, we use $D$=2 to do experiments and find that the optimal $W$ is 1 for most of the datasets. H-VMMR requires a smaller number of $W$ and $D$ compared with H-MMR. It is because co-purchased items usually look similar or have visual relationships with each other. There are some information overlaps between the hierarchical model and visual features.

**Iterations.** Figure 3 shows that it usually takes 500-1000 iterations for our model to converge, which depends on the size of the training dataset. Usually AUC on validation set is higher than test set.

**HR and NDCG.** Since traditional machine learning methods (row *h-k*) achieve the highest AUC in most cases, we compute HR@N and NDCG@N of these models in Figure 4. Due to the extreme sparsity of our data, $N \in \{20, 50, 100\}$ is used. It can be concluded that our model (red) outperforms the comparison models in terms of HR@N and NDCG@N for most of the datasets, which proves our



**(a) Men multi (all)**

**(b) Men shoes (all)**

**(c) Men multi (cold)**

**(d) Men shoes (cold)**

**(e) Women multi (all)**

**(f) Women shoes (all)**

**(g) Women multi (cold)**

**(h) Women shoes (cold)**

**Figure 4: HR@N and NDCG@N of VBPR, H-VBPR, VMMR and H-VMMR on 8 datasets (higher is better).**

model is better at predicting top-ranked items since it optimizes the ranking from two directions: enlarging the set of potential positive items as well as true negative items.

## 5.6 Implementation Details

We do hyper parameter tuning carefully for all of the models. For our model, the learning rate $\alpha$ is 0.005. We use the same value for $\eta$ and the standard deviation $\sigma_1$. We also bound $Q_{i,j}$ in a range of [0.75,1]. The weight parameter $\omega_+$ is 1. The variance $\sigma_2^2$ is 50. The embedding size of all latent vectors is 48. The attenuation coefficient $\gamma \in (0.9,1)$. $\lambda_\theta$ is 0.0001 for all visual parameters.

## 6 CONCLUSIONS AND FUTURE WORK

Existing personalized ranking methods make use of exclusion strategy to reduce the number of items the user may like to improve ranking accuracy, which cannot best meet the seller's demands. In this paper, we address this problem by proposing a novel hierarchical minimax ranking method with predictive sampling and a modified version of negative sampling based on visual features. The statistical results prove that our model outperforms others.

In the future, we will consider an alternative method for extracting visual features from items, which will further improve the ranking accuracy. Besides, we want to further increase the diversity of the top-$N$ results and at the same time improve AUC. It is challenging because often diversity and AUC go in opposite directions. Other information of users and items can be incorporated into our model to solve this problem.

## ACKNOWLEDGMENTS

The work described in this paper was fully supported by grants from City University of Hong Kong (Project No. 7004916 and 9678139).

## REFERENCES

[1] Chenwei Cai, Ruining He, and Julian McAuley. 2017. SPMC: socially-aware personalized markov chains for sparse sequential recommendation. *arXiv preprint arXiv:1708.04497* (2017).

[2] Dong-Kyu Chae, Jin-Soo Kang, Sang-Wook Kim, and Jaeho Choi. 2019. Rating Augmentation with Generative Adversarial Networks towards Accurate Collaborative Filtering. In *The World Wide Web Conference*. ACM, 2616–2622.

[3] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531* (2014).

[4] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. ACM, 7–10.

[5] Guillem Cucurull, Perouz Taslakian, and David Vazquez. 2019. Context-Aware Visual Compatibility Prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 12617–12626.

[6] Yashar Deldjoo, Mehdi Elahi, Massimo Quadrana, and Paolo Cremonesi. 2018. Using visual features based on MPEG-7 and deep learning for movie recommendation. *International journal of multimedia information retrieval* 7, 4 (2018), 207–219.

[7] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph Neural Networks for Social Recommendation. In *The World Wide Web Conference*. ACM, 417–426.

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[9] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

[10] Ruining He, Chen Fang, Zhaowen Wang, and Julian McAuley. 2016. Vista: A visually, socially, and temporally-aware model for artistic recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 309–316.

[11] Ruining He, Wang-Cheng Kang, and Julian McAuley. 2017. Translation-based recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 161–169.

[12] Ruining He, Chunbin Lin, Jianguo Wang, and Julian McAuley. 2016. Sherlock: sparse hierarchical embeddings for visually-aware one-class collaborative filtering. *arXiv preprint arXiv:1604.05813* (2016).

[13] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 507–517.

[14] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. 2018. Adversarial personalized ranking for recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 355–364.

[15] 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*. International World Wide Web Conferences Steering Committee, 173–182.

[16] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *ICDM*, Vol. 8. Citeseer, 263–272.

[17] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 675–678.

[18] Yannis Kalantidis, Lyndon Kennedy, and Li-Jia Li. 2013. Getting the look: clothing recognition and segmentation for automatic product suggestions in everyday photos. In *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval*. ACM, 105–112.

[19] Wang-Cheng Kang, Chen Fang, Zhaowen Wang, and Julian McAuley. 2017. Visually-aware fashion recommendation and design with generative image models. In *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE, 207–216.

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

[21] Qiang Liu, Shu Wu, and Liang Wang. 2017. DeepStyle: Learning user preferences for visual recommendation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 841–844.

[22] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 43–52.

[23] Pablo Messina, Vicente Dominguez, Denis Parra, Christoph Trattner, and Alvaro Soto. 2019. Content-based artwork recommendation: integrating painting metadata with neural and manually-engineered visual features. *User Modeling and User-Adapted Interaction* 29, 2 (2019), 251–290.

[24] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International Conference on Data Mining*. IEEE, 995–1000.

[25] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 452–461.

[26] Thanh Tran, Xinyue Liu, Kyumin Lee, and Xiangnan Kong. 2019. Signed Distance-based Deep Memory Recommender. In *The World Wide Web Conference*. ACM, 1841–1852.

[27] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 515–524.

[28] Qinyong Wang, Hongzhi Yin, Zhiting Hu, Defu Lian, Hao Wang, and Zi Huang. 2018. Neural memory streaming recommender networks with adversarial training. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2467–2475.

[29] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*. ACM, 12.

[30] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. 2014. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, 283–292.

[31] Yongfeng Zhang, Qingyao Ai, Xu Chen, and W Bruce Croft. 2017. Joint representation learning for top-n recommendation with heterogeneous information sources. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 1449–1458.