Towards Hands-Free Visual Dialog Interactive Recommendation

Tong Yu, Yilin Shen, Hongxia Jin
Samsung Research America
Mountain View, CA, USA
{tong.yu, yilin.shen, hongxia.jin}@samsung.com

Abstract

With the recent advances of multimodal interactive recommendations, the users are able to express their preference by natural language feedback to the item images, to find the desired items. However, the existing systems either retrieve only one item or require the user to specify (*e.g.*, by click or touch) the commented items from a list of recommendations in each user interaction. As a result, the users are not hands-free and the recommendations may be impractical.

We propose a hands-free visual dialog recommender system to interactively recommend a list of items. At each time, the system shows a list of items with visual appearance. The user can comment on the list in natural language, to describe the desired features they further want. With these multimodal data, the system chooses another list of items to recommend. To understand the user preference from these multimodal data, we develop neural network models which identify the described items among the list and further predict the desired attributes. To achieve efficient interactive recommendations, we leverage the inferred user preference and further develop a novel bandit algorithm. Specifically, to avoid the system exploring more than needed, the desired attributes are utilized to reduce the exploration space. More importantly, to achieve sample efficient learning in this hands-free setting, we derive additional samples from the user's relative preference expressed in natural language and design a pairwise logistic loss in bandit learning. Our bandit model is jointly updated by the pairwise logistic loss on the additional samples derived from natural language feedback and the traditional logistic loss. The empirical results show that the probability of finding the desired items by our system is about 3 times as high as that by the traditional interactive recommenders, after a few user interactions.

Introduction

In traditional interactive recommender systems, the user feedback (*i.e.*, rating or click) is continuously collected to improve the recommendations. The recent advances of recommenders enable the users to comment on the item images via natural language, to express their preference (Guo et al. 2018; 2019; Vo et al. 2019; Yu, Shen, and Jin 2019; Yu et al. 2019). However, the existing systems either retrieve only one item or require the user to specify (*e.g.*, by click or touch) the commented items in each user interaction. As a result, the recommendations are less efficient and the users are not hands-free. In certain user scenarios of personal assistants, such as *Amazon echo show*[1] and *Google home hub*[2], the hands-free feature is very desirable. For instance, when the users prepare foods or take care of babies, it is unrealistic to expect the users to easily touch or click the screen.

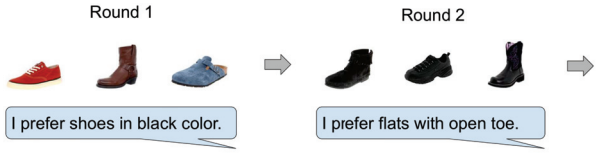


Figure 1: An example in our hands-free recommender system. In each round, the user provides *one comment to the recommended list*, and the system is not aware of which item(s) among the list are commented. The system predicts which parts are commented by the user, understands the user preference, and provides another list of recommendations.

In this paper, (i) we design a hands-free interactive recommender system, based on which (ii) we further develop a bandit algorithm to efficiently find the desired items. In our hands-free system, without touching or clicking the screens, the users find the items by providing natural language feedback to the item images. At each time, the user is provided a list of recommended items. Then, the user provides one comment to the list of items, to approach the desired item. Figure 1 shows an example. The recommendations are provided to a user with multiple rounds. Assume the user wants black flats with open toe and floral pattern. In the first round, the user is recommended with a list of shoes. The user views the list and provides a comment '*I prefer shoes in black color*'. Then, the system updates and give recommendations in the second round. The user gets more recommendations of black shoes, and makes another comment '*I prefer flats with*

[1] https://www.amazon.com/All-new-Echo-Show-2nd-Gen/dp/B077SXWSRP

[2] https://store.google.com/product/google_home_hub

open toe'. With this comment, the system updates again. To build a fully hands-free system, the inputs should be raw speech signals. In this work, we simplify this step and assume we already have the spoken language output from an automatic speech recognition component.

To understand the user preference from these multimodal data, we develop neural network models which identify the described items among the list and further predict the desired visual attributes. To achieve efficient interactive recommendations, we leverage the inferred user preference and further develop a bandit algorithm, sleeping pairwise ranking bandit (`SPR bandit`). Specifically, to avoid the system exploring more than needed, the desired attributes are utilized to reduce the exploration space. More importantly, to achieve sample efficient learning in this hands-free setting, we derive additional samples from the user's relative preference expressed in natural language and design a novel pairwise logistic loss in bandit learning. Our bandit model is jointly updated by the pairwise logistic loss on the additional samples derived from natural language feedback and the traditional logistic loss.

In summary, we make three major contributions.

- We propose a novel hands-free multimodal interactive recommender which enable the users to find the desired items by natural language, without requiring the users to touch the screens.
- We develop multimodal neural networks to effectively understand the user natural language feedback on the visual appearance of a list of items.
- With limited positive data samples in this hands-free setting, we propose `SPR bandit` to achieve more sample efficient learning, compared to traditional bandits.

Related Work

Interactive Recommenders with Multimodal Data

Multimodal data, such as natural language and image, have been leveraged in recommender systems. With the recent advances of deep learning and reinforcement learning, conversational interactive recommendations are becoming increasingly popular (Christakopoulou, Radlinski, and Hofmann 2016; Greco et al. 2017; Sun and Zhang 2018; Li et al. 2018; Zhang et al. 2018). A recent work (Guo et al. 2018) enables user natural language feedback to candidate items' visual appearance for interactive item retrieval. (Yu, Shen, and Jin 2019) further extends this approach to interactively recommend a list of items. More advanced ways of combining the text and image inputs for image retrieval are proposed and compared in (Vo et al. 2019; Guo et al. 2019). However, these multimodal systems either retrieve only one item or require the user to specify (*e.g.*, by click or touch) the commented items from a list of recommendations. As a result, the users are not hands-free. In certain user scenarios of personal assistants (*e.g.*, Amazon echo show and Google home hub), the hands-free feature is very desirable. For instance, when the users prepare foods or take care of babies, it is unrealistic to expect the users to touch or click the screen.

Interactive Recommenders by Bandits

Multi-armed bandits balance exploration and exploitation in traditional interactive recommender systems (Chapelle and Li 2011). There are different online algorithms maximizing the rewards online, such as Upper Confidence Bound (UCB) and Thompson Sampling (TS) (Auer, Cesa-Bianchi, and Fischer 2002; Russo et al. 2018). Cascading bandits are proposed to interactively recommend a list of items (Kveton et al. 2015; Zong et al. 2016). Sleeping bandits are studied in the setting where the set of available arms varies arbitrarily with time (Kleinberg, Niculescu-Mizil, and Sharma 2010; Chatterjee et al. 2017). Traditional bandit algorithms update the models mainly based on simple user feedback, such as ratings or clicks. In the early time steps, these algorithms usually explore more than needed (Liu et al. 2018b), due to the random initialization of the confidence interval in UCB or prior distribution in TS. However, it is very important to recommend suitable items in the early time steps. In this paper, we improve the interactive recommender's performance in the early time steps by leveraging the multimodal data and proposing a sample efficient bandit algorithm. Augmenting bandits with deep learning models has recently been studied in (Riquelme, Tucker, and Snoek 2018; Weber et al. 2018; Liu et al. 2018a).

A Hands-free Multimodal Recommender

In this section, we introduce our system. Our system and the data inputs are shown in Figure 2. The data inputs include the recommended lists and the user feedback. There are three components in our system: *item identifier*, *visual dialog encoder*, and `SPR bandit`. At each time, the user is provided with a list of recommendations. Then, the user gives a comment on the visual appearance of the items in the list. The system receives the item images, and user comments in natural language. When a user finds a desired item, the user describes a sentence to notify the system. The item identifier predicts which items are commented. Based on the commented items' visual appearance and the comment in natural language, the visual dialog encoder infers the user preference. Based on the inferred preference, `SPR bandit` provides another list of recommendations.

Item Identifier to Predict Described Items

In this section, we introduce the item identifier component, which predicts the commented items within the list of recommendations. In our system, in each round the user provides a comment to a list of items. The system is not aware which parts of items are commented by the user. We need to know which items the user is talking about in the list, to further infer the user preference.

It is very challenging to predict the exact commented items, because the same comment can apply to multiple items. For example, assume the recommended list includes blue clogs and yellow sandals. If a user describes '*I prefer red shoes*', the user may want either red clogs or red sandals. Alternatively, we identify and retain the items not containing the visual feature described by the user. As an example, assume the recommended list includes blue clogs,

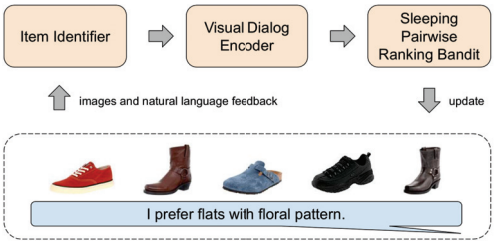


Figure 2: The components and data inputs of our system.

yellow sandals and red boots. If a user describes '*I prefer red shoes*', the user typically want red clogs or red sandals rather than red boots, since red boots already contain the described feature (*i.e.*, red). If a user desires red boots, the user will either like the recommended item or describe some visual attributes other than color. We remove the red boots before we use visual dialog encoder to infer the user preference.

The item identifier consists of an image encoder, a text encoder and an identification classifier, as shown in the green block of Figure 3. We encode each item by an image encoder. Specifically, following (Guo et al. 2018), we encode the image of item e_{t-1} by ResNet101 and then a linear mapping, where the output is denoted as $\texttt{ImgEnc}(e_{t-1})$. The ResNet101 is pre-trained on ImageNet. Given each pair of candidate item e_{t-1} and desired item e^*, the user describe their difference o_{t-1} in natural language. We encode each comment by a text encoder. Specifically, we encode this feedback o_{t-1} by one-hot encoding, a linear mapping and then a CNN (Kim 2014), where the output is denoted as $\texttt{TxtEnc}(o_{t-1})$. Then, $\texttt{ImgEnc}(e_{t-1})$ and $\texttt{TxtEnc}(o_{t-1})$ are concatenated and embedded through a linear transformation to be further used as the input of the identification classifier, which is a 3-layer fully connected neural network. Item e_t is derived by considering the features in both e_{t-1} and o_{t-1}. Therefore, e_{t-1} does not contain the feature described by o_{t-1}, while e_t contains the feature described by o_{t-1}. Given inputs of e_{t-1} and o_{t-1}, the neural network outputs $\texttt{Iden}(e_{t-1}, o_{t-1})$, which should be 1. Given inputs of e_t and o_{t-1}, the neural network outputs $\texttt{Iden}(e_t, o_{t-1})$, which should be 0. We train the item identifier by minimizing a cross-entropy loss function.

Visual Dialog Encoder to Infer Desired Attributes

The visual dialog encoder component understands the desired attributes. Specifically, this component outputs the item which looks similar to the desired items, given the inputs of one candidate item and its received comment.

To develop the visual dialog encoder, there are various operators to fuse the image and text inputs with various loss functions (Vo et al. 2019; Guo et al. 2019; 2018). We follow the approach in (Guo et al. 2018). The visual dialog encoder has an image encoder and text encoder. The image encoder and text encoder in this component shares the same architecture and parameters with the image encoder and text encoder in the item identifier. As shown in Figure 3, $\texttt{ImgEnc}(e_{t-1})$ and $\texttt{TxtEnc}(o_{t-1})$ are concatenated and embedded through a linear transformation to be further used as the input of a state tracker, which is a GRU followed by a linear mapping. The output of the state tracker is denoted as $\texttt{VisDiaEnc}(e_{t-1}, o_{t-1})$, the final encoding of the item image e_{t-1} and its comment o_{t-1}. The visual dialog encoder is trained by optimizing the triplet loss and cross entropy loss (Guo et al. 2018). The distance between $\texttt{VisDiaEnc}(e_{t-1}, o_{t-1})$ and $\texttt{ImgEnc}(e^*)$ is minimized during the training, where e^* is a desired item. As a result, we can rely on the desired attributes $\texttt{VisDiaEnc}(e_{t-1}, o_{t-1})$ to find items close to the desired item e^*.

In our system, the components of item identifier and visual dialog encoder are pre-trained on a training dataset in the offline setting with the cross-entropy loss function for the identification classifier and the loss functions in (Guo et al. 2018). To provide personalized recommendations based on the output of item identifier and visual dialog encoder, SPR bandit learns in an online setting for new users (not necessarily existing in the training data).

SPR bandit to Make Interactive Recommendations

SPR bandit makes interactive recommendations based on the multimodal feedback. At each time, SPR bandit selects a list of items to recommend. Then, the user provides multimodal feedback to the list of items. With this feedback, the bandit model updates and recommend another list for next time. SPR bandit is based on Thompson sampling (TS), considering TS generally outperforms Upper Confidence Bound (UCB) (Chapelle and Li 2011). SPR bandit is different from the traditional bandit algorithms in that (i) it uses a constrained exploration strategy with the constraints learned from the visual dialog encoder, and (ii) it is jointly optimized by a traditional loss and a pairwise logistic loss, to achieve sample efficient learning in this hands-free setting.

Model and Online Leaning Setting Following (Chapelle and Li 2011), we assuming there exists $\theta^* \in \mathbb{R}^{d\times 1}$ such that, for any item e, the probability of it being desired is $\sigma(x_e^\top \theta^*)$. Here $x_e = \texttt{ImgEnc}(e) \in \mathbb{R}^{d\times 1}$ is the encoding of item e and $\sigma(.)$ is the sigmoid function. The agent is to learn θ_t performing similarly to θ^* online. Assume the size of the recommended list is K, in total there are L items, $[L]$ is a ground set of the L items, and $\Pi_K([L])$ is the set of all K-permutations of set $[L]$. Following (Zong et al. 2016), we define the reward function $f(A_t, \theta_t)$: if at time t the recommended list A_t contains at least one desired item, $f(A_t, \theta_t) = 1$. Otherwise, $f(A_t, \theta_t) = 0$. We minimize the expected cumulative regret

$$R(n) = \mathbb{E}[\sum_{t=1}^{n} (f(A^*, \theta_t) - f(A_t, \theta_t))],$$

where $A^* = \arg\max_{A\in\Pi_K([L])} f(A, \theta^*)$ is the optimal list. To learn the optimal list over time, we develop an online algorithm with two special designs, as follows.

Constrained and Efficient Exploration Similar to (Yu, Shen, and Jin 2019), we develop a bandit algorithm with constrained explorations to find the desired items more efficiently. The traditional bandits usually explores more than

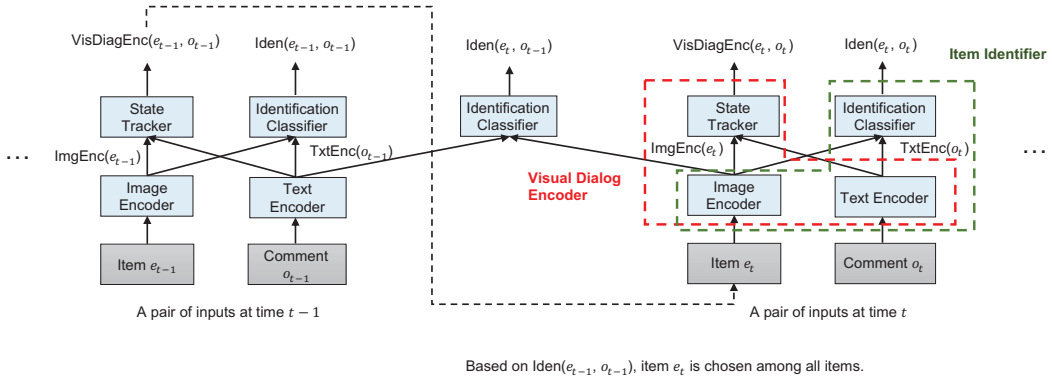


Figure 3: The models of the item identifier and visual dialog encoder at time $t-1$ and time t. In this figure, we omit some linear transformation layers for a clear overview of the model.

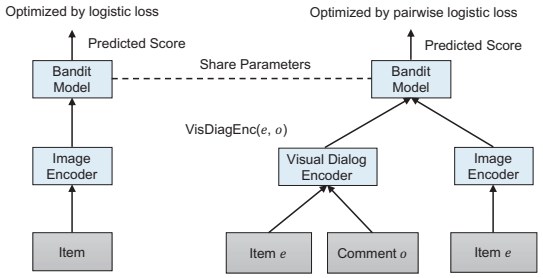


Figure 4: The `SPR bandit` model. The left part shows the model optimized by traditional loss, and the right part shows the model optimized by the pairwise logistic loss. All items are within the constrained exploration space. The detail of the visual dialog encoder is shown in Figure 3.

needed (Liu et al. 2018b). To control the exploration, our algorithm learns constraints from user feedback via the visual dialog encoder. The constraints guide exploration to consider only a subset of the items at each time. Thus, the user can find more desired items with fewer interactions.

Specifically, we *reduce the exploration space* by leveraging the output from the visual dialog encoder. Assume item e receives comment o. With $\texttt{VisDiaEnc}(e, o)$, we can roughly estimate the region where user desired items are in the space of arms. In our algorithm, only the arms within this region are available to recommend and the rest arms are not available, that is, sleeping (Kleinberg, Niculescu-Mizil, and Sharma 2010; Chatterjee et al. 2017). In practice, we maintain a vector x_{center} to estimate the centroid of the region. Only the set $\mathcal{B}$ containing the top K' arms closest to x_{center} is available to recommend and the rest arms are not available. We keep updating x_{center}, so that it converges to the true centroid of the region.

Pairwise Logistic Loss for Sample Efficient Learning Compared to the setting that a list may receive multiple comments (Yu, Shen, and Jin 2019), in our hands-free setting a list only receives one comment. (Yu, Shen, and Jin 2019) shows that the system performance degrades with fewer comments. Besides, there are not enough samples (especially *positive samples*, *i.e.*, items being desired) to learn an accurate model in early steps of bandit learning. Therefore, it is desirable to develop a sample efficient learning algorithm to effectively leverage the limited number of comments. To alleviate this issue, we develop a pairwise logistic loss function to effectively derive additional data samples to augment our bandit model and achieve high accuracy, especially in the early steps of learning. The pairwise logistic loss function characterizes how the bandit model fits the pairwise ranking of items based on the user natural language feedback. Our exploration model is jointly updated by the traditional logistic loss function and our pairwise logistic loss function. For simplicity of exposition, we assume only one item is commented at each time step in this section, while multiple items could be commented at each time. Algorithm 1 presents our algorithm in a more general case.

Given an item e and its comment o, we can obtain $\texttt{VisDiaEnc}(e, o)$. Inspired by the rank logistic regression (Lo et al. 2009; Wu et al. 2012), we develop a pairwise logistic loss function for bandit learning. As in (Lo et al. 2009; Herbrich, Graepel, and Obermayer 1999), optimizing the pairwise logistic loss function given x_e and $\texttt{VisDiaEnc}(e, o)$ is mathematically equivalent to optimizing the logistic loss function with feature $\texttt{VisDiaEnc}(e, o) - x_e$ and label 1 when θ is a linear model. That is, we minimize the following pairwise loss

$$\min_{\theta} \lambda\theta^\top\theta - \sum_{t=1}^{n-1} \log(\sigma((\texttt{VisDiaEnc}(e_t, o_t) - x_{e_t})\theta)), \tag{1}$$

where n is the total number of user interaction rounds, λ is the weight of the regularization term, and e_t is the commented item with comment o_t at time t. This design leads to more sample efficient learning by collecting more positive samples with reward 1 in the early steps, when most recommended items only receive reward 0. Consider an example where the user has 10 rounds of interactions with the system. Usually the recommendations in the first 10 rounds are not the user's desired items. So we assume that only 2 items are desired by the user. In the rest 8 rounds, we only receive the

Algorithm 1 `SPR bandit`

Input: λ, L, K, K', d
1 $\tau = 1, \tau' = 1, \bar{\theta}_0 = \mathbf{0} \in \mathbb{R}^{d\times 1}, S_0 = \lambda^{-1} I_d \in \mathbb{R}^{d\times d}$, $x_{\text{center}} = \mathbf{0} \in \mathbb{R}^{d\times 1}, \mathcal{B} = [L]$
2 **forall** $t = 1, \cdots, n$ **do**
3 Sample the model parameters $\theta_t \sim \mathcal{N}(\bar{\theta}_{t-1}, S_{t-1})$
4 **forall** $k = 1, \cdots, K$ **do**
5 $a_k^t \leftarrow \arg\max_{e \in \mathcal{B} - \{a_1^t, \cdots, a_{k-1}^t\}} x_e^\top \theta_t$
6 **end**
7 Recommend items $\mathcal{A}_t \leftarrow (a_1^t, \cdots, a_K^t)$
8 **if** *Any items in* $\mathcal{A}_t$ *are desired* **then**
9 Observe the desired items $\mathcal{C}_t \subset [K]$
10 **else**
11 Observe the comment o_t and set $\mathcal{O}_t = \varnothing$
12 **forall** $k = 1, \cdots, K$ **do**
13 **if** $\texttt{Iden}(a_k^t, o_t) = 1$ **then** $\mathcal{O}_t = \mathcal{O}_t \cup \{k\}$;
14 **end**
15 **end**
16 **forall** $k = 1, \cdots, K$ **do**
17 $e \leftarrow a_k^t$, $z_\tau \leftarrow x_e$, $y_\tau \leftarrow \mathbb{1}\{k \in \mathcal{C}_t\}$, $\tau \leftarrow \tau + 1$
18 **if** $k \in \mathcal{O}_t$ **then**
19 $z_\tau \leftarrow \texttt{VisDiaEnc}(e, o_t) - x_e$ *// pairwise loss*
20 $y_\tau \leftarrow 1$
21 $x_{\text{center}} \leftarrow \frac{x_{\text{center}} \times (\tau' - 1) + \texttt{VisDiaEnc}(e, o_t)}{\tau'}$
22 $\tau \leftarrow \tau + 1, \tau' \leftarrow \tau' + 1$
23 **end**
24 **end**
25 **forall** $k' = 1, \cdots, K'$ **do**
26 $b_{k'} \leftarrow \arg\min_{e \in [L] - \{b_1, \cdots, b_{k'-1}\}} ||x_e - x_{\text{center}}||$
27 **end**
28 $\mathcal{B} \leftarrow (b_1, \cdots, b_{K'})$
29 Update S_t and $\bar{\theta}_t$
30 **end**

user comments to improve the undesired items. With the traditional loss, only 2 positive samples are available in the first 10 rounds. By optimizing the pairwise logistic loss, 8 extra samples can be derived. Therefore, by jointly optimizing the traditional loss and the pairwise logistic loss, $2 + 8 = 10$ positive samples are available in the first 10 rounds, which leads to more sample efficient learning. In the online setting, the bandit model parameters are *jointly updated* by the pairwise logistic loss function and the traditional logistic loss, as shown in the right part of Figure 4.

Algorithm By considering the above two designs, we summarize `SPR bandit` as shown in Algorithm 1. The inputs are the hyper-parameter λ of the Gaussian distribution, the total number of items L, the size of list K, a hyper-parameter K' and the dimensionality of the image feature vector d. From line 3 to line 7, we sample the model parameters, select the top K items from $\mathcal{B}$ and recommend them to the user. From line 8 to line 24, we collect the user feedback. The set $\mathcal{O}_t \subset [K]$ records the indices of the commented items within the recommended list. The set $\mathcal{C}_t \subset [K]$ records the indices of the satisfying items within the recommended list. In line 17, the samples are collected for bandit learning with the traditional logistic loss. In lines 19 and 20, based on the user relative preference expressed in natural language, additional samples are derived for bandit learning with the pairwise logistic loss. In traditional bandits, the samples are the bandit arms and their reward. We derive additional samples, which are the difference between a pair of arms and the two arms' relative relationship expressed in natural language. In line 21, we update x_{center}. From line 25 to line 28, we update $\mathcal{B}$. Only the set $\mathcal{B}$ containing the top K' arms closest to x_{center} is available to recommend, to achieve constrained exploration. In line 29, we update distribution of the model parameter. Following (Chapelle and Li 2011; Bishop 2006; MacKay 1992), we approximate the posterior distribution of the model parameter by a Gaussian distribution. Specifically, the posterior at time t can be approximated by $\theta_t \sim \mathcal{N}(\bar{\theta}_{t-1}, S_{t-1})$. The covariance matrix is $S_t = \left(\sum_{i=1}^{\tau} \sigma(z_i^\top \theta_t)(1 - \sigma(z_i^\top \theta_t)) z_i z_i^\top + \lambda I_d\right)^{-1}$. To estimate $\bar{\theta}_t$, we minimize the following loss

$$\min_{\theta} \lambda \theta^\top \theta - \sum_{i=1}^{\tau} \Big(y_i \log(\sigma(z_i \theta)) + (1 - y_i) \log(1 - \sigma(z_i \theta)) \Big). \quad (2)$$

By only considering the user comments on the undesired items, the loss in (2) is reduced to the loss in (1).

Experiments

Dataset and Online Evaluation

We evaluate different approaches on the footwear dataset (Berg, Berg, and Shih 2010; Guo et al. 2018). The online evaluation of our system is non-trivial, since it is unrealistic to collect the user comments on all possible list of items. For feasible and comparable evaluations, we follow the evaluation protocol used in (Guo et al. 2018; Yu, Shen, and Jin 2019). By generating user comments in natural language[3], a *relative captioner* is used to act as a surrogate for real human users. The inputs of the relative captioner is a pair of (i) *candidate image* and (ii) *desired image*. The output of the relative captioner describes the visual differences between any pair of candidate image and desired image. Following (Yu, Shen, and Jin 2019), in each user session we assume the user finds a target category of shoes (*e.g.*, 'wedding shoes', 'athletic shoes' and 'boots'). We further assume that the users randomly comment on one candidate item from the recommended list. With the above settings, the user comments on all possible list of items are available.

Similar to (Guo et al. 2018), we train the item identifier and visual dialog encoder on $10,000$ images, and evaluate our recommender in the online setting on another dataset with $4{,}658$ images. Following (Chapelle and Li 2011; Kveton et al. 2015; Zong et al. 2016; Yu, Shen, and Jin 2019), we evaluate different approaches by the average cumulative reward, which is defined as $R'(n) = \frac{1}{n} \sum_{t=1}^{n} r_t$. At time step t, if at least one item belongs to the target category, $r_t = 1$. Otherwise, $r_t = 0$. For each experiment, we run 20 times, and report the average results and the standard errors. We show the results up to $n = 100$ steps. Except the user case and user study sections, the size of the list is $K = 10$.

[3]The authors of (Guo et al. 2018) release the captioner codes in Github: https://github.com/XiaoxiaoGuo/fashion-retrieval.

Improvements by Multimodal Data

To validate the advantages of utilizing multimodal data, we compare our system to iterative recommenders only relying on user click feedback to recommend a list of items. Specifically, we compare SPR bandit to CascadeUCB1 (Kveton et al. 2015) and a parametric extension CascadeLinTS (Zong et al. 2016). CascadeUCB1 extends UCB1 (Auer, Cesa-Bianchi, and Fischer 2002) in solving traditional bandit problems. In each step of CascadeUCB1, the upper confidence bounds (UCBs) of the first clicked item and all items up to the first clicked item in the list are updated. The items with the top K highest UCBs form the recommended list. Instead of treating each item independently, CascadeLinTS extends CascadeUCB1 by learning a predictor of the attraction probabilities of items from their features. Following the cascade setting (Kveton et al. 2015; Zong et al. 2016), we only consider the first desired item in each list, such that the only difference between SPR bandit and (Kveton et al. 2015; Zong et al. 2016) is that SPR bandit has the extra input of the items' visual appearances and user feedback in natural language. Another potential baseline is the contextual bandit, where the output of the visual dialog encoder is used as extra input of the traditional bandit model. By incorporating context, we develop C-CascadeLinTS. Note that our system is not directly comparable to (Yu, Shen, and Jin 2019), since the data inputs are different. (Yu, Shen, and Jin 2019) requires the users to touch the screen and point out (*i.e.*, click) the commented items, while our system is hands-free and no click data is needed. See more discussions in the following pairwise logistic loss experiment.

We report the results in Figure 5a. With multimodal data, SPR bandit performs much better than CascadeUCB1 and CascadeLinTS. With parameterization, CascadeLinTS outperforms CascadeUCB1. When $n = 10$, SPR bandit achieves 3 times reward, compared to CascadeUCB1 and CascadeLinTS. This demonstrates a significant advantage of our system when n is small. C-CascadeLinTS fails to outperform CascadeLinTS. Our problem is essentially different from the contextual bandit problem. In each user session, the user finds desired items with similar characteristics. As a result, the context embedding is always very similar and does not provide extra information in the model learning. Instead, with context the model has more parameters and converges more slowly.

In the hands-free setting, the user comment is not restricted on a specified item in the list. Therefore, in real world, user comments can be misleading or even erroneous. We further evaluate the robustness of our system by randomly introducing noisy comments. With some probability, we randomly sample a user comment as the response to list of items. By adjusting the probability, we evaluate the system robustness in tasks with different difficulty levels in Figure 5b. With moderate amount of noise (25% or 50%), our approach still performs well. Note in line 21 of Algorithm 1, the cumulative moving average of the visual dialog encoder outputs can effectively smooth out some noisy comments. We also evaluate our system's robustness with items randomly identified from the list (*i.e.,* without using the identification classifier). Similarly, due to the cumulative moving average, only minor performance decrease is observed in the simulated evaluation with moderate noise. We further observe that in certain cases the user comments are direct reference of the desired images such that the information from the candidate images become less important, which is similar to the observation in (Guo et al. 2019).

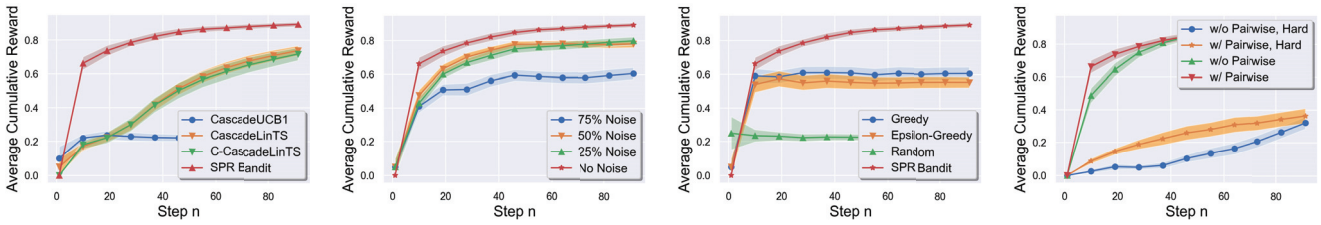
Improvements by SPR bandit

Constrained and Efficient Exploration To validate the exploration strategy of our algorithm, given the same multimodal inputs we compare SPR bandit to other common search strategies, including random, greedy and ϵ-greedy (Sutton and Barto 2018). We design a simple greedy approach by extending (Guo et al. 2018) to recommend multiple items. At each step, we identify the commented item from the list. Based on the item and its feedback, the visual dialog encoder generates an item looking similar to the desired item. With KNNs, the top K candidate items closest to the generated item are recommended in the next step. Based on the greedy approach, we further develop a ϵ-greedy baseline. With probability ϵ, the K recommendations are randomly sampled. Otherwise, the above greedy approach is adopted. We report the results by ϵ-greedy when $\epsilon = 0.1$.

The results are reported in Figure 5c. With the benefit of the multimodal data, greedy, ϵ-greedy, and SPR bandit all perform similarly well before $n = 10$. After $n = 10$, greedy and ϵ-greedy cannot efficiently learn, while SPR bandit learns steadily and achieve much better performance than greedy and ϵ-greedy. With the simple exploration, ϵ-greedy fails to outperform greedy because of the large number of arms (*i.e.*, actions). On the contrary, with our careful design, SPR bandit effectively handles exploration and exploitation and performs much better than the baselines.

Pairwise Logistic Loss for Sample Efficient Learning We validate the sample efficient learning by the pairwise logistic loss. We compare SPR bandit with its variant without using pairwise loss. In addition to the above evaluation setting, we further evaluate them on a harder task when the positive samples are very limited. In this task, the user aims to find rare items which only belong to a small portion of the items in each shoes category. Specifically, in this hard task, only 10 items (sampled from each target category of shoes) are desired by the user in each session.

The results are reported in Figure 5d. There is a clear gap between the performance of SPR bandit with pairwise logistic loss and SPR bandit without pairwise logistic loss. In the early steps of algorithms or in the hard task, the positive samples (*i.e.*, desired items) are especially limited. In the both two cases, we observe the improvements by pairwise logistic loss become more significant. The results validate the sample efficient learning by the pairwise logistic loss. The algorithm in (Yu, Shen, and Jin 2019) can not be directly compared to SPR bandit. However, for (Yu, Shen, and Jin 2019), we can predict the commented items in each list without the desired items. With this extra information, the algorithm (Yu, Shen, and Jin 2019) can be reduced to SPR bandit without pairwise logistic loss. Therefore, SPR bandit with pairwise logistic loss can achieve more



(a) (b) (c) (d)

Figure 5: The experimental results: (a) the improvements by multimodal data, (b) the system robustness in the hands-free setting, (c) the results by different search strategies, and (d) more sample efficient learning by the pairwise logistic loss.

sample efficient learning than (Yu, Shen, and Jin 2019).

A Use Case from Logged Experimental Results

Based on our logged results, we show an example of how the user interacts with our recommender system to achieve the desired items in Figure 6. In this example, the desired items belong to a specific type of sport shoes. We show the logged recommended items and the user feedback in rounds (*i.e.*, steps of algorithms) 1, 3, and 5. Due to the limited space, we show the results when $K = 4$. With more rounds of recommendations, we observe that more recommendations are visually similar to the desired items (marked by the green check). In round 1, the user comments '*I prefer white sneakers with low top sneakers*'. In rounds 3 and 5, there are more sneakers which are white or have while components. We do observe that some shoes in rounds 3 and 5 do not perfectly meet the user's previous comment(s). This is caused by the exploration strategy of the bandits in early steps of learning.

A User Study of Interactive Recommender Systems

In the above experiments, we follow the evaluation protocol in (Guo et al. 2018) and evaluate all approaches with a relative captioner. Although the quality of the adopted captioner has been comprehensively evaluated by real humans in (Guo et al. 2018), it is still very interesting to conduct user study to compare our system to the interactive recommender system only relying on traditional user feedback (*i.e.*, clicks). Specifically, we compare our system to the system based on `CascadeLinTS`, considering that `CascadeLinTS` outperforms `CascadeUCB1` in the above experiments. For each system, we conducted 400 runs of user study. In each run of user study, the user aims to find a particular type of shoes and has at most 30 rounds of interactions with the system. We adopt the setting where $K = 4$. With $K = 4$, we can easily display the images of the recommended items in the screen of mobile devices, and still enjoy the advantages of recommending multiple items in each user interaction.

We report the results under two metrics. The first metric is the total number of desired items found within the 30 rounds of interactions. The second metric is the number of rounds used to find the desired items for the first time. With our system, in average the users find 11.8 items and they find the first desired item after 6 rounds. On the contrary, with the traditional recommender, in average the users find 4.7 items and they find the first desired item after 13 rounds. These

Figure 6: A use case example from the logged results. The green check marks the desired items found by our system.

results further validate the advantages of our recommender compared to the traditional interactive recommender.

Conclusion

With the recent advances of multimodal recommenders, the users are able to express their preference by natural language feedback to the item images, to find the desired items. However, the existing systems either retrieve only one item or require the user to specify (*e.g.*, by click or touch) the commented items from a list of recommendations in each user interaction (Guo et al. 2018; Yu, Shen, and Jin 2019). As a result, the users are not hands-free. However, in certain user scenarios of personal assistants, the hands-free feature is very desirable. For instance, when the users prepare foods or take care of babies, it is unrealistic to expect the users to touch or click the screen. In this hands-free setting, we further propose a sample efficient bandit algorithm to interactively recommend a list of items. The empirical results validate that the probability of finding the desired items by our system is about 3 times as high as that by the traditional interactive recommenders, after a few user interactions.

References

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine*

learning 47(2-3):235–256.

Berg, T. L.; Berg, A. C.; and Shih, J. 2010. Automatic attribute discovery and characterization from noisy web data. In *ECCV*, 663–676. Springer.

Bishop, C. M. 2006. Pattern recognition and machine learning (information science and statistics).

Chapelle, O., and Li, L. 2011. An empirical evaluation of thompson sampling. In *NIPS*, 2249–2257.

Chatterjee, A.; Ghalme, G.; Jain, S.; Vaish, R.; and Narahari, Y. 2017. Analysis of thompson sampling for stochastic sleeping bandits. In *UAI*.

Christakopoulou, K.; Radlinski, F.; and Hofmann, K. 2016. Towards conversational recommender systems. In *KDD*, 815–824. ACM.

Greco, C.; Suglia, A.; Basile, P.; and Semeraro, G. 2017. Converse-et-impera: Exploiting deep learning and hierarchical reinforcement learning for conversational recommender systems. In *Conference of the Italian Association for Artificial Intelligence*, 372–386. Springer.

Guo, X.; Wu, H.; Cheng, Y.; Rennie, S.; Tesauro, G.; and Feris, R. 2018. Dialog-based interactive image retrieval. In *NIPS*. 676–686.

Guo, X.; Wu, H.; Gao, Y.; Rennie, S.; and Feris, R. 2019. The fashion iq dataset: Retrieving images by combining side information and relative natural language feedback. *arXiv preprint arXiv:1905.12794*.

Herbrich, R.; Graepel, T.; and Obermayer, K. 1999. Support vector learning for ordinal regression. In *1999 Ninth International Conference on Artificial Neural Networks ICANN 99.(Conf. Publ. No. 470)*, 97–102.

Kim, Y. 2014. Convolutional neural networks for sentence classification. In *EMNLP*, 1746–1751.

Kleinberg, R.; Niculescu-Mizil, A.; and Sharma, Y. 2010. Regret bounds for sleeping experts and bandits. *Machine learning* 80(2-3):245–272.

Kveton, B.; Szepesvari, C.; Wen, Z.; and Ashkan, A. 2015. Cascading bandits: Learning to rank in the cascade model. In *ICML*, 767–776.

Li, R.; Kahou, S. E.; Schulz, H.; Michalski, V.; Charlin, L.; and Pal, C. 2018. Towards deep conversational recommendations. In *Advances in Neural Information Processing Systems*, 9725–9735.

Liu, B.; Yu, T.; Lane, I.; and Mengshoel, O. 2018a. Customized nonlinear bandits for online response selection in neural conversation models. In *AAAI*.

Liu, B.; Wei, Y.; Zhang, Y.; Yan, Z.; and Yang, Q. 2018b. Transferable contextual bandit for cross-domain recommendation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Lo, H.-Y.; Chang, K.-W.; Chen, S.-T.; Chiang, T.-H.; Ferng, C.-S.; Hsieh, C.-J.; Ko, Y.-K.; Kuo, T.-T.; Lai, H.-C.; Lin, K.-Y.; et al. 2009. An ensemble of three classifiers for kdd cup 2009: Expanded linear model, heterogeneous boosting, and selective naive bayes. In *KDDCup*, 57–64.

MacKay, D. J. 1992. The evidence framework applied to classification networks. *Neural computation* 4(5):720–736.

Riquelme, C.; Tucker, G.; and Snoek, J. 2018. Deep bayesian bandits showdown. In *ICLR*.

Russo, D. J.; Van Roy, B.; Kazerouni, A.; Osband, I.; Wen, Z.; et al. 2018. A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning* 11(1):1–96.

Sun, Y., and Zhang, Y. 2018. Conversational recommender system. In *SIGIR*, 235–244.

Sutton, R. S., and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.

Vo, N.; Jiang, L.; Sun, C.; Murphy, K.; Li, L.-J.; Fei-Fei, L.; and Hays, J. 2019. Composing text and image for image retrieval-an empirical odyssey. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6439–6448.

Weber, N.; Starc, J.; Mittal, A.; Blanco, R.; and Màrquez, L. 2018. Optimizing over a bayesian last layer. In *NeurIPS workshop on Bayesian Deep Learning*.

Wu, K.-W.; Ferng, C.-S.; Ho, C.-H.; Liang, A.-C.; Huang, C.-H.; Shen, W.-Y.; Jiang, J.-Y.; Yang, M.-H.; Lin, T.-W.; Lee, C.-P.; et al. 2012. A two-stage ensemble of diverse models for advertisement ranking in kdd cup 2012. In *KDDCup*.

Yu, T.; Shen, Y.; Zhang, R.; Zeng, X.; and Jin, H. 2019. Vision-language recommendation via attribute augmented multimodal reinforcement learning. In *Proceedings of the 27th ACM International Conference on Multimedia*, 39–47. ACM.

Yu, T.; Shen, Y.; and Jin, H. 2019. An visual dialog augmented interactive recommender system. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 157–165. ACM.

Zhang, Y.; Chen, X.; Ai, Q.; Yang, L.; and Croft, W. B. 2018. Towards conversational search and recommendation: System ask, user respond. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 177–186. ACM.

Zong, S.; Ni, H.; Sung, K.; Ke, N. R.; Wen, Z.; and Kveton, B. 2016. Cascading bandits for large-scale recommendation problems. In *UAI*, 835–844.