Linked Variational AutoEncoders for Inferring Substitutable and Supplementary Items

Vineeth Rakesh
Technicolor
vineeth.mohan@technicolor.com

Suhang Wang
Pennsylvania State University
suhang.wang@psu.edu

Kai Shu
Huan Liu
Arizona State University
{skai2,huan.liu}@asu.edu

ABSTRACT

Recommendation in the modern world is not only about capturing the interaction between users and items, but also about understanding the relationship between items. Besides improving the quality of recommendation, it enables the generation of candidate items that can serve as substitutes and supplements of another item. For example, when recommending Xbox, PS4 could be a logical substitute and the supplements could be items such as game controllers, surround system, and travel case. Therefore, given a network of items, our objective is to learn their content features such that they explain the relationship between items in terms of substitutes and supplements. To achieve this, we propose a generative deep learning model that links two variational autoencoders using a connector neural network to create Linked Variational Autoencoder (LVA). LVA learns the latent features of items by conditioning on the observed relationship between items. Using a rigorous series of experiments, we show that LVA significantly outperforms other representative and state-of-the-art baseline methods in terms of prediction accuracy. We then extend LVA by incorporating collaborative filtering (CF) to create CLVA that captures the implicit relationship between users and items. By comparing CLVA with LVA we show that inducing CF-based features greatly improve the recommendation quality of substitutable and supplementary items on a user level.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Information systems** → **Collaborative filtering**.

KEYWORDS

Variational Autoencoder; Deep Learning; Link Prediction; Recommendation; Graphical Model

ACM Reference Format:
Vineeth Rakesh, Suhang Wang, Kai Shu, and Huan Liu. 2019. Linked Variational AutoEncoders for Inferring Substitutable and Supplementary Items. In *The Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19), February 11–15, 2019, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3289600.3290963

WSDM '19, February 11–15, 2019, Melbourne, VIC, Australia

ACM ISBN 978-1-4503-5940-5/19/02…$15.00
https://doi.org/10.1145/3289600.3290963

Figure 1: An example of item recommendation that serve as substitutes and supplements to the main product of interest.

1 INTRODUCTION

Recommender systems have witnessed a significant growth over the last decade. Evolving from simple content and collaborative filtering based techniques to more complex hybrid methods, modern recommender systems are an integral part of e-commerce domain such as Amazon, photo-sharing domains such as Instagram, and location-based social networks such as Yelp. Despite a myriad of research on improving and personalizing recommendation systems [1, 3, 6], only a few have attempted to understand the relationship between items [19, 32, 34]. Also known as *candidate generation*, the objective of this research is to retrieve candidates out of billions of items in order to recommend items that are relevant to a given context. In this paper, we aim to understand the relationship between items to predict whether an item can serve as a substitute or a supplement to another item. Substitutable products are those that are interchangeable, while supplementary products are those that can be purchased together. Figure 1 illustrates a toy example; here, the main product of interest is an LG 4K TV. The substitute is another 4K TV from a different brand and the supplements are streaming devices such as Roku and Apple TV.

Given a collection of items and their reviews, the goal of this paper is to *learn the latent features of items that are indicative of the relationship between items in terms of substitutes and supplements.* Recently, McAuley et al. [19] proposed an algorithm called *Sceptre* that predicts the links between items in Amazon. Here, the authors use latent dirichlet allocation (LDA) [2] to capture the content information of items as a document-topic distribution θ and fit a logistic model over θ to predict the relationship between items. The model is trained by jointly optimizing the LDA and the logistic

parameters. Despite being a successful model, Sceptre suffers from several shortcomings that are detailed as follows: (a) **Content sparsity**: Most reviews are extremely short and lack contextual information. The inferior performance of LDA over short text is a well known problem that has been studied by several researchers [12, 20, 24]. (b) **Limiting Topic Hierarchies**: Sceptre assumes that products are classified into a predefined hierarchy of topics and the authors use this information to limit the number of topics in LDA. However, this method has been widely criticized in the topic modeling literature for overfitting and producing noisy word clusters.

Deep learning has emerged as the state-of-the-art technique for natural language processing, image analysis, and speech recognition due to its ability to learn representative features and capture non-linear relationships in data [21]. Its prowess in feature learning has enabled the applicability of deep learning models to a wide array of problems, including link prediction and recommendation. For example, [10] propose a neural collaborative filtering which integrates matrix factorization with multi-layer perceptron for recommendation; [28] and [15] replace the LDA component of the collaborative topic regression (CTR) model [26] with autoencoders to improve the recommendation performance; Wang et al. [27] investigate relational deep learning for link prediction. Regardless of these contributions, studies that exploit deep learning for inferring item relationships is rather limited.

Therefore, in this paper, we investigate the novel problem of exploiting deep learning algorithms for inferring item relationships. We propose a novel generative deep learning model called Linked Variational Autoencoder (LVA) that predicts the relationship between items in terms of substitutes and supplements. This is achieved by *linking two variational autoencoders [13] and conditioning the feature learning process on the observed link relationship between items*. The proposed model overcomes the drawbacks of Sceptre in the following ways. First, LVA uses VAE to learn the content features of items; this mitigates the problems associated with LDA. Second, unlike Sceptre that alternates between learning the parameters of LDA and fitting a logistic model over the document-topic distribution, LVA follows a full Bayesian approach. This is achieved by directly conditioning the latent features of item-pairs on their observed link relationship, which leads to better approximation of the parameters. In addition to the proposed LVA, we also introduce an extension to our model by integrating LVA with probabilistic matrix factorization (PMF) to create CLVA. CLVA combines collaborative- and content-based information to provide personalized recommendation. The major contributions are summarized as follows:

- We propose a novel deep generative model called Linked Variational Autoencoder (LVA), which can simultaneously capture item features and item relationships to facilitate substitute and supplementary item prediction. LVA accurately predicts the link between items even in the presence of cold start problem.
- We extend LVA by integrating LVA with PMF. The resulting collaborative LVA (CLVA) can model user ratings in addition to item features and item relations, which helps personalized recommendation.
- By conducting extensive experiments on real-world datasets, we demonstrate the effectivenss of LVA in various tasks such as item link prediction, global recommendation and out-of-the-box recommendations.

To the best of our knowledge, we are the first to propose linked VAEs that *learns the latent features of items by conditioning on the observed relationship between items*. These features not only capture the content-representation, but also embed the link between items in-terms of substitutes and supplements.

The rest of this paper is organized as follows. We introduce the generative process of LVA model and the parameter inference in Section 2. We extend LVA by integrating PMF to create the CLVA model in Section 2.2. We conduct experiments in Section 3. Finally, we review related work in Section 4 and conclude our paper with future work in Section 5.

2 THE PROPOSED FRAMEWORK - LVA

Problem Formulation: Let $\{v_i\}_1^V$ be the set of items and $\{X^v\}_1^V$ be the set of reviews for each item. Given a pair of items (X^a, X^b) and an observed label y indicating the relationship (i.e., substitutes and supplements) and the direction of relationship between items a and b, our objective is to learn the latent attributes Z^a and Z^b such that they explain the label y. The label y in our setting has four categories that signifies the type of link and the direction of link. Specifically, labels 1 and 2 denote the substitute link with directions $a \rightarrow b$ and $a \leftarrow b$ respectively and labels 3 and 4 denote supplementary link with directions $a \rightarrow b$ and $a \leftarrow b$ respectively.
The LVA Model: Figure 2 (a) shows the graphical structure of the proposed Linked Variational Autoencoder (LVA), where the grey nodes indicate the observed variables, neural network layer g (i.e., the function approximators) are denoted by the green nodes, and the white nodes denote the unobserved variables. The model has two parts, (1) variational autoencoders (VAE), which is depicted inside the red box and (2) the link predictor part, which is depicted inside the blue box.

Table 1: List of notations used in this paper.

Symbol	Description
U	number of users
V	number of items
$X = \|V\| \times \|D\|$	item-attribute matrix
y	observed label for the latent features (Z^a, Z^b)
Z	latent attributes of item contents
ρ	latent attributes items in CLVA
η	latent attributes users
D	set of item attributes
K	number of latent attributes
$\mathbf{W}$	weights of the neural network
$\mathbf{b}$	bias vector of the neural network
θ	weights of decoder network
ϕ	weights of the encoder network

The latent attributes $\{Z_k^v\}_{k=1}^K$ of item v are learned using (VAE) [13]. An autoencoder (AE) is a neural network trained to learn latent representation that is good at reconstructing its input and VAE is a probabilistic extension of AE, which models the attribute learning process as a generative algorithm. Besides being an effective model for learning feature representations, the probabilistic nature of VAE facilitates seamless integration of other generative models such as

PMF. Algorithm 1 illustrates the generative process of LVA. In the encoder part (lines 7-10), the mean μ and covariance Σ are drawn from a normal distribution parameterized by the neural network g with weights $\mathbf{W}$ and bias vector $\mathbf{b}$ and the latent layer Z is sampled from a normal distribution with parameters μ and Σ. Here, the suffix a/b indicates the sampling operation for both items a and item b. In the decoder part (lines 2-5), the features X^a and X^b are reconstructed and the observed label y is drawn from a categorical distribution parameterized by latent distributions of both Z^a and Z^b. It should be noted that since y is observed, due to the property of *common effect*, there is a flow of influence from Z^a to Z^b and vice versa [14].

Algorithm 1: Generative process of the LVA model

1 **Decoder:**
2 **for** each item pair $(a, b) \in V$ **do**
3 Draw $\mathbf{X}^a \sim \mathcal{N}(\mathbf{g}_{L,a}\ \mathbf{W}_L + \mathbf{b}_L, \lambda_n^{-1}\mathbf{I}_V)$
4 Draw $\mathbf{X}^b \sim \mathcal{N}(\mathbf{g}_{L,b}\ \mathbf{W}_L + \mathbf{b}_L, \lambda_n^{-1}\mathbf{I}_V)$
5 Draw label $y \sim \mathbf{Cat}(Z^a, Z^b; \mathbf{W}_L)$
6 **Encoder:**
7 **for** each item pair $(a, b) \in V$ **do**
8 Draw $\mu_{a/b}, \sim \mathcal{N}(\mathbf{g}_{L,a/b}\mathbf{W}_\mu + \mathbf{b}_\mu, \lambda_n^{-1}\mathbf{I}_K)$
9 Draw $\Sigma_{a/b}, \sim \mathcal{N}(\mathbf{g}_{L,a/b}\mathbf{W}_\Sigma + \mathbf{b}_\Sigma, \lambda_n^{-1}\mathbf{I}_K)$
10 Draw $Z^{a/b} \sim \mathcal{N}(\mu_{a/b}, \Sigma_{a/b})$

2.1 Parameter Inference - LVA

The input to LVA is a pair of item features X^a and X^b, which are the set of reviews for items a and b respectively. The objective is to infer Z^a and Z^b, given the features of items and the label y of the item pairs. In other words, we are interested in learning the probability $p(Z^a, Z^b|X^a, X^b, y)$, where y indicates whether the tuple (X^a, X^b) are supplements or substitutes and the direction of this relation. There are two main challeges to estimating this posterior. First, as with most bayesian models, there is no closed form solution since the normalization factor is intractable to compute. Second, since the label is observed, there is an implicit flow of influence between the latent variables Z^a and Z^b [14]. To overcome these bottlenecks, we approximate the posterior using the method of variational inference; additionally, we decouple the latent variables by assuming that their variational distributions depend only on their respective neural networks. Under these assumptions, the evidence lower bound (ELBO) for our objective is defined as follows:

$$\begin{aligned} log\, Pr(X^a, X^b, y) \geq E_{q_\phi}\big[&log\, Pr_\theta(X^a|Z^a) + \\ &log\, Pr_\theta(X^b|Z^b) + log\, Pr_\theta(Z^a) + log\, Pr_\theta(Z^b) + \\ &log\, Pr_\theta(y|Z^a, Z^b) - log\, q_\phi(Z^a, Z^b|X^a, X^b, y)\big] \end{aligned} \tag{1}$$

the approximate posterior $q_\phi(.)$ is assumed to have a fully factorized form. The RHS of the above expression is expanded as follows:

$$\begin{aligned} &E_{q_\phi(Z^a|.)}\big[log\, Pr_\theta(X^a|Z^a) - \mathcal{D}(\,q_\phi(Z^a|X^a)||\,Pr_\theta(Z^a))\big] \\ &+ E_{q_\phi(Z^b|.)}\big[log\, Pr_\theta(X^b|Z^b) - \mathcal{D}(\,q_\phi(Z^b|X^b)||\,Pr_\theta(Z^b))\big] \\ &+ E_{q_\phi(Z^a, Z^b|.)}\big[logPr_\theta(y|Z^a, Z^b)\big] \end{aligned} \tag{2}$$

The above equation has three parts: the first part is the ELBO for LVA with inputs X^a, the second is the ELBO for X^b and the third is the classifier part of the model. Concisely, the above equation is written as follows:

$$\mathcal{L}(\theta^a, \phi^a) + \mathcal{L}(\theta^b, \phi^b) + E_{q_\phi(Z^a, Z^b|.)}\big[log\, Pr_\theta(y|Z^a, Z^b)\big] \tag{3}$$

We adopt the reparameterization trick over $\mathcal{L}(\theta^a, \phi^a)$ to obtain samples of z from an isotropic normal distribution. This results in the following expression:

$$E_{\epsilon\sim\mathcal{N}(0,I)}\big[log\, Pr_\theta(x^a|z^a)\big] - \mathcal{D}(q_\phi(z^a|x^a)||Pr(z^a)) \tag{3.1}$$

where x^a are the attributes of a single item and $z^a = \mu_\phi(x) + \epsilon \odot \sigma_\phi(x)$, $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. The first part of the above equation is simply the sum squared error. The second part is the KL divergence between two multivariate gaussian distributions, which has a closed form solution defined as follows:

$$\begin{aligned} \mathcal{D}(q_\phi(z^a|x^a)||Pr(z^a)) = -\frac{1}{2}\Big(&tr\big(\Sigma_\phi(x^a)\big) + \\ &\big(\mu_\phi(x^a)\big)^T\big(\mu_\phi(x^a)\big) - log\, det\big(\Sigma_\phi(x^a)\big)\Big) \end{aligned} \tag{3.2}$$

So far, we derived the first term of equation (3). The second term (i.e, variational loss $\mathcal{L}(\theta^b, \phi^b)$) takes the same form as expression (3.1) and the third term is the classifier part of LVA , which can be realized using a softmax function. To be more specific, once we sample the Zs for an item pair (a, b), the latent features become the inputs of the classifier neural network where the label y can be drawn from a softmax function. By substituting equations (3.1), (3.2) along the softmax function in equation (3) the final objective is given by:

$$\begin{aligned} \frac{V}{M}\sum_{i=1}^{M}\Big[&- \frac{1}{2\sigma^2}(x_i^a - \mu_\phi(x_i^a))^2 - \mathcal{D}^a - \frac{1}{2\sigma^2}(x_i^b - \mu_\phi(x_i^b))^2 - \mathcal{D}^b \\ &+ log\left(\frac{e^{\theta^{(y)T}\mathbf{z}}}{\sum_{j=1}^{Y} e^{\theta^{(j)T}\mathbf{z}}}\right)\Big] \end{aligned} \tag{4}$$

where D^a and D^b are the KL divergence terms of items a and b respectively, V is the total number of data points (or items) and M is the set of random samples drawn from V. The LVA can be trained by taking the gradients of the above expressions w.r.t parameters θ and ϕ and updating these parameters over multiple epochs using stochastic gradient-ascent.

2.2 Infusing Personalization

Collaborative LVA: As explained in Section 1, LVA is not personalized model; in the sense, the recommendations provided by LVA is same for all users. Therefore, in Figure 2 (b), we extend LVA to CLVA that incorporates user information in the form of collaborative filtering to personalize the recommendation of substitutes and supplements. This framework adopts the formulation of [28] and [15], where the authors modify the topic modeling part of collaborative topic regression (CTR) [26] with a deep learning model. CLVA embeds the content information of items along with the collaborative filtering by integrating LVA with probabilistic matrix factorization (PMF). In Figure 2 (b), the LVA is depicted inside the red box and the PMF part is depicted inside the blue box. The

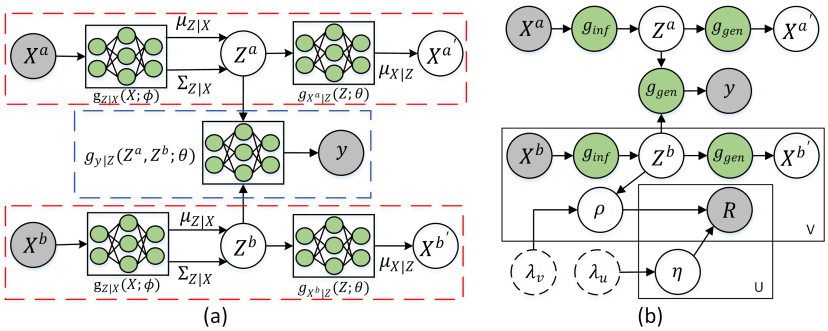


Figure 2: Graphical structure of the proposed models (a) shows the base LVA, where the function $g(.)$ indicates the neural network component and (b) shows the collaborative LVA model with the PMF component.

generative process of LVA remains similar to Algorithm 1, while the generative process of PMF is described as follows:

1. For each item $v \in V$ do:
   (a) draw latent features $Z_v^b \sim \mathcal{N}(\mu, \Sigma)$
   (b) draw offset $\epsilon_v \sim \mathcal{N}(0, \lambda_v^{-1} I_K)$ and set $\rho = \epsilon + Z_v^b$
2. For each user $u \in U$ do: Draw $\eta \sim \mathcal{N}(\lambda_u^{-1} I_k)$
3. For each user-item pair (u, v) do: Draw $R_{u,v} \sim \mathcal{N}(\rho^T \eta, c_{u,v}^{-1})$

Similar to CTR, the key property lies in how the item latent attribute ρ_v is generated. The term $\rho = \epsilon + Z_v^b$ signifies that item latent attribute ρ_v is close to topic proportions Z^b, but could diverge from it if it has to. This divergence is introduced by the offset ϵ, which captures the collaborative filtering-based features.

2.3 Parameter Inference - CLVA

The objective is to infer the posterior $P(Z^a, Z^b, \eta, \rho | X^a, X^b, y)$. Similar to LVA, we proceed by defining the ELBO for CLVA as follows:

$$\mathcal{L}^{MAP^*}(\rho, \eta, \theta, \phi) = E_q\left[log\, Pr(Z^a, Z^b, \eta, \rho, R, y, X^a, X^b) - log\, q(\eta, \rho, Z^a, Z^b)\right] \quad (5)$$

Contrary to LVA, in CLVA, we have a variational distribution over four latent variables. However, since the primary objective of this paper is to introduce a deep generative model for link prediction, we simplify the estimation by considering the variational distribution over Z^a and Z^b. By expanding the likelihood term of the above expression we obtain the following:

$$E_{q_\phi}\left[log\, Pr_\theta(X^a|Z^a) + log\, Pr_\theta(X^b|Z^b) + log\, Pr_\theta(Z^a) + log\, Pr_\theta(Z^b) + log\, Pr_\theta(y|Z^a, Z^b)\right] + E_{q_\phi}\left[log\, Pr_\theta(\rho|Z^b) + log\, Pr(R|\rho, \eta) + log\, Pr(\eta)\right] - E_{q_\phi}\left[log\, q_\phi(Z^a, Z^b|X^a, X^b, y)\right] \quad (6)$$

where $\mathcal{L}$(LVA) is the ELBO of the LVA model from equation (4). From the above expression, one can observe that the evidence lower bound of CLVA remains very similar to that of LVA (see equation 1). The only major difference between these two models is the second term inside the expectation, which embeds the PMF component into the LVA model. Substituting the corresponding distributions in the above equation we obtain the following MAP estimate:

$$\mathcal{L}^{MAP^*}(\rho, \eta, \theta, \phi) = \mathcal{L}(\text{LVA}) - \sum_{u,v} \frac{C_{u,v}}{2}(R_{u,v} - \eta_u^T \rho_v)^2 - \frac{\lambda_\eta}{2}\sum_u ||\eta_u||_2^2 - \frac{\lambda_\rho}{2}\sum_v E_{q_\phi(Z^b|.)}||\rho_v - Z_v^b||_2^2 \quad (7)$$

Readers should note that it is not strictly a MAP estimate since we still infer parameters Z^a and Z^b using a variational distribution. Hence, we denote the above expression as $\mathcal{L}^{MAP^*}$.

3 EXPERIMENTS

In this section, we perform a rigorous series of quantitative and qualitative experiments over various datasets and test cases to evaluate the proposed model. The parameter setting for LVA model is as follows: batch size = 512, latent attributes $Z = 100$, $epochs = 70$ and weights for the VAE network is set as 0.3 and the classifier network as 0.9. For CLVA the model settings for the PMF part are as follows: number of latent user attributes $\eta = 5$, number of item user attributes $\rho = 5$. For all our experiments, we user 80% of the data for training, 10% for validation and 10% for testing. The models are implemented using Keras with Tensorflow as backend. A working code of LVA can be downloaded from our Github repository[1].

3.1 Dataset

For our experiments, we obtain the co-purchase data of items in Amazon from McAuley et al. [19]. The actual co-purchase data comprises of twenty different product categories. Nonetheless, for our experiments, we select products from five different categories that are decided based on the number of reviews: (1) categories with largest collection of reviews namely, Books, Electronics and Movies and (2) categories with sparse set of reviews namely, Men's Clothing and Musical Instruments. Given a pair of products A and B, the dataset defines four types of links between these products. An item B is deemed as a substitute of item A based on the following: (a) users who viewed A also viewed B, or (b) users who viewed item A eventually bought item B. Alternatively, an item B is a supplement of item A based on the following: (a) users who bought A also bought B, or (b) users frequently bought A and B together. To prepare the dataset for our experiments, we perform some basic

[1] https://github.com/VRM1/WSDM19

Table 2: Dataset Statistics of item reviews, links and users used for our experiments.

Dataset	ItmRevs	Links	#Users
Books	966K	10.7M	8M
Electronics	349K	5.6M	4.2M
Men's Clothing	158K	966K	3.1M
Movies	145K	2.8M	2M
Musical Instr	67K	1.1M	339K

pre-processing steps such as lematizing, stemming and removing reviewers which have less than ten words. The statistics of the resulting dataset is shown in Table 2.

3.2 Baselines

We compare the performance of our model with four representative and state-of-the-art baseline methods for item link prediction:

Random: The random baseline is a modification of the Sceptre model [19], where the link probabilities F_β and F_η are replaced with random numbers between 0 and 1. This implies the parameters of the logistic predictor are not learned; thereby making the prediction of substitutes and supplements completely random.

Logit-LDA: Unlike LVA which jointly learns the topic distribution and the link relationship between items, the logistic LDA (abbreviated as Logit-LDA) first learns the item-topic distribution θ of reviews by independently training an LDA model on item reviews. It then trains logistic classifiers on θ to predict the relationship between items. The topic model is trained with K = 150 topics.

RTM: Introduced by Chang et al. [4], the relational topic model (RTM) is a hierarchical model that is specifically designed to infer the relationship between networks of documents. Given a pair of documents, RTM explicitly ties the content of the documents with the connections between them. In other words, the inferred latent topic space of items are conditioned on the observed link relationship; in our case, the relationship is quantified as substitutes and supplements. The number of topics K was set to 20 and hyperparameters α was set to 0.1.

Sceptre: The state-of-the-art model for predicting relationship between products in terms of substitutes and supplements [19]. Sceptre fits a logistic classifier over the topic space of LDA, which not only learns the relationship between items, but also the direction of relationship to predict whether A is a substitute/supplement of B and vise versa. The model has been shown to produce high prediction accuracy while being highly scalable to millions of reviews and product relations.

3.3 Evaluation Methedology

We categorize the evaluation into two tasks: (a) prediction, and (b) recommendation. For the first task, the model should not only predict whether a given pair of items are substitutes or supplements, but also the direction of the link. The test accuracy is determined by comparing the predicted relations with the ground truth to determine the number of true positives and false positives. For the second task, we recommend a limited number of substitutes and supplements. The evaluation of recommendation is divided into two types: (a) using LVA and CLVA, recommend a global set of items that is same for every user in our database (i.e., unpersonalized) and (b) using CLVA, recommend items on a local level where each user in our database gets a personalized suggestion of substitutable and supplementary items. In *global recommendation*, we rank the set of substitutes and supplements according to their probability, and recommend top K items. Both forms of recommendation are evaluated using *precision@K*, which is defined as the fraction of rankings in which the true recommend items are ranked in the top-K positions. Experiments are performed by splitting the data into 80% for training, 10% for validation, and 10% for test. The reported results are based on a 5-fold cross validation technique.

3.4 Link Prediction

Table 3 compares the accuracy scores of LVA along with other baselines. Overall, one can observe that LVA outperforms all other baselines with significant gains over Logit-LDA and RTM. The performance of LVA is also extremely consistent over all datasets with an accuracy of over 90%. While Sceptre produces slightly better results over Men's clothing, LVA reigns superior over rest of the categories. The biggest difference in performance between these two models is exhibited over books and movies with LVA clearly leading Sceptre with a boost of 5%-10% in accuracy. A plausible explanation to this outcome could be the inferior performance of the LDA topic model that in used in Sceptre to learn the content-based features of items. Books and movies are one of the largest review corpus in our database with millions of user reviews. Since most reviews are usually very short, the lack of word co-occurrence severely hampers the learning abilities of LDA by inducing problems such as overfitting, and noise in topic distributions [20]. It is also important to note that both Logit-LDA and RTM are very good in predicting substitutes, but the accuracy is significantly lower for supplementary items. This is because, items that serve as substitutes are usually from the same category (i.e., Xbox could be a substitute for PS4), while items that serve as supplements could be from a broad range of categories. For example, for an Xbox, supplements could be items such as wireless speakers, protection case, game cds, etc. This outcome also co-insides with the paper by Mcgure et al. [19], who reported similar observations. Another interesting outcome is the performance between Logit-LDA and RTM. RTM outperforms Logit-LDA by $2-3\%$ over all datasets. This clearly demonstrates how conditioning the latent space over observed link relationship and jointly learning the label distribution and item-topic distribution is essential for our problem.

3.5 Prediction in Cold Start Scenario

So far, we showed that LVA is capable of predicting substitutes and supplementary links between items with high accuracy. As impressive as it may be, these results are for items that have a certain number of user reviews. However, this is not the case always, when an item is new to the marketplace, the user reviews are completely absent; this translates to the well-known problem of *cold start*. To overcome this bottleneck, we extract other meta-data information about the products. For instance, information such as title and product description from the manufacturer are available for most items in the Amazon database. Therefore, instead of reviews, we use the meta-data as features to train our model. The results of this experiment are shown in Table 4. One can observe that LVA is able to accurately predict the link even with the lack of

Table 3: Performance comparison of LVA in terms of the accuracy scores, where *"All Links"* denote both substitutable and supplementary links. LVA clearly outperforms *Sceptre* in four out of five categories with largest gain over Books and Movies.

Dataset	Accuracy	Random	Logit-LDA	Rel-LDA	Sceptre	LVA
Books	Substitute	65.43%	83.37%	86.07%	93.41%	**95.71%**
	Supplement	55.81%	68.32%	68.91%	86.82%	**92.07%**
	All Links	57.12%	72.28%	74.07%	89.45%	**94.2%**
Electronics	Substitute	64.92%	90.17%	88.75%	95.73%	95.47%
	Supplement	55.38%	65.43%	69.21%	88.11%	**91.18%**
	All Links	58.48%	70.04%	72.27%	90.59%	**92.36%**
Men's Clothing	Substitute	59.44%	72.32%	74.17%	95.63%	92.82%
	Supplement	57.19%	66.04%	68.12%	94.42%	93.18%
	All Links	57.67%	69.39%	71.23%	94.36%	93.11%
Movies	Substitute	-	-	-	-	-
	Supplement	50.32%	55.14%	60.43%	86.01%	**95.63%**
	All Links	50.32%	54.64%	61.21%	86.19%	**95.63%**
Musical Instruments	Substitute	-	-	-	-	-
	Supplement	50.04%	58.19%	60.56%	90.22%	**93.47%**
	All Links	50.04%	57.11%	60.51%	89.84%	**93.05%**
Average Score for All Links		54.72%	64.69%	67.85%	90.08%	**93.67%**
LVA performance gain		**38.98%**	**28.98%**	**25.82%**	**3.59%**	

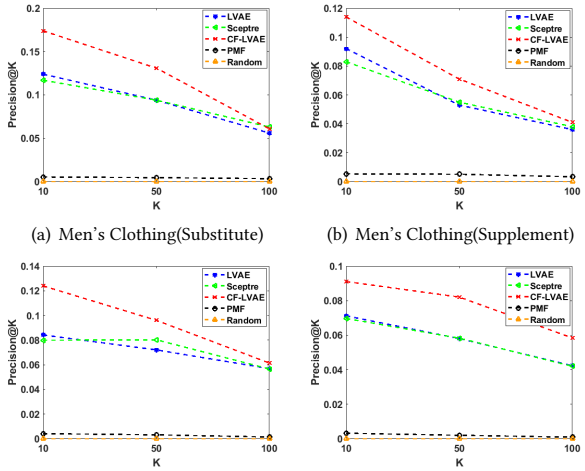
Table 4: Performance of Sceptre and LVA over cold start items. Both models provide a high accuracy despite the lack of reviews.

Dataset	Accuracy	Sceptre	LVA
Books	Substitute	93.41%	93.11%
	Supplement	91.14%	93.51%
Electronics	Substitute	91.35%	94.87%
	Supplement	90.81%	92.02%
Men's Clothing	Substitute	96.42%	95.55%
	Supplement	96.88%	96.19%

reviews. Following a trend similar to Table 3, LVA lags slightly behind Sceptre for men's clothing, but delivers better accuracy over electronics and books. Due space constraints, we only show the results for three categories; however, in our testing, the performance over other categories was in the range of 90-95%.

3.6 Global Recommendation

Recommendation in our setting is incredibly difficult since, for a given item, there are probably thousands of pairs of substitutable and supplementary items. Nevertheless, when recommending items to users, we have to show a limited set of items that are of highest interests to users. Figure 3 reports the precision scores of LVA and CLVA along with other baselines. The candidate links used for training are discarded from the test data. The collaborative filtering (CF) baseline is obtained by independently training the PMF component of CLVA. The random baseline has a precision in the range of 10^{-5} to 10^{-4}, which proves that obtaining high precision scores in our scenario is incredibly difficult. Both LVA and CLVA are better than PMF by more than an order-of-magnitude; this shows that collaborative features captured by PMF is not sufficient to recommend substitutes and supplementary items. Contrary to the results of link prediction (Table 3), the performance of Sceptre is very close to the proposed LVA. LVA is able to achieve a precision of up to 12% for



(a) Men's Clothing(Substitute) (b) Men's Clothing(Supplement)

(c) Electronics(Substitute) (d) Electronics(Supplement)

Figure 3: Precision performance for global recommendation.

men's clothing, while CLVA outperforms its counterpart with a precision of 17%. Although the independent use of PMF did not yield a good performance, integrating collaborative information in LVA certainly boosts its performance. Additionally, the performance of all the models are lower for the electronics domain. A possible reason for this outcome can be attributed to the size of dataset, where the number of unique items in electronics are significantly higher than men's clothing, which in-turns affects the ranking.

3.7 User-Based Recommendation

In the previous section, we reported the results of global recommendation, where every user gets the same set of items irrespective of their personal preference. In other words, the models were trained and tested purely on an item-level that excluded the user

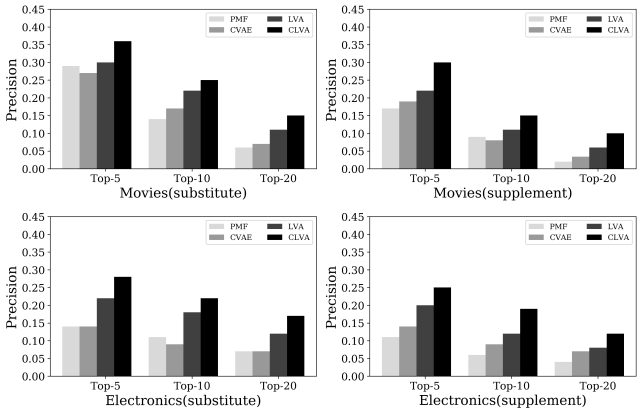


Figure 4: Precision performance for user-based recommendation.

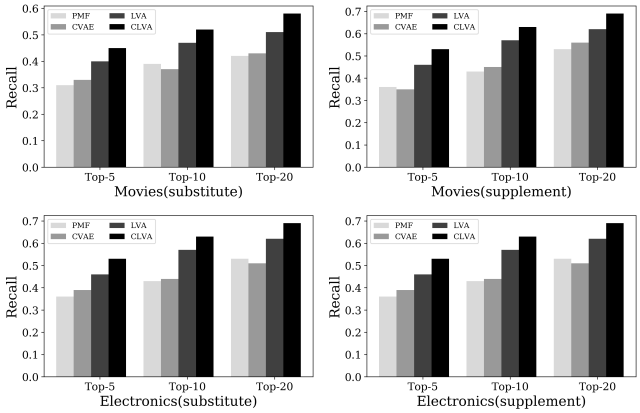


Figure 5: Recall performance for user-based recommendation.

information. Here, we report the performance of user-level (or personalized) recommendation by comparing CLVA with LVA, PMF and CVAE [16]. CVAE is a hybrid generative model which learns deep latent representations from content data in an unsupervised manner and also learns implicit relationships between items and users from both content and rating. It can be treated as a variant of CLVA without considering substitute and supplementary information. Our objective is to see whether integration of collaborative with content-based features and link information improves the recommendation of substitutable and supplementary items. The outcome of this experiment is depicted in Figures 4 and 5 respectively. From the results, it is quite apparent that CLVA leads to better precision and recall across all datasets. Specifically, when it comes to precision, CLVA is atleast an order-of-magnitude better than LVA. Although LVA outperforms PMF, when compared to global recommendation (Figure 4), the difference is not significant. This is because LVA does does not incorporate any form of user information whereas PMF is able to leverage collaborative features to boost its performance. Finally, it is interesting to note that when it comes to recommending substitutes and supplements, the state-of-the-art collaborative variational autoencoder (CVAE) is no better than the traditional PMF.

3.8 Out-of-the-Box Recommendation

Amazon has several million items; nonetheless, only a fraction of them have co-purchase information. Therefore, there are several unexplored set of items that could serve as viable alternative or addition to the main item of interest. Our goal is to recommend substitutes and supplements that are not present in the actual ground truth. To achieve this, given a product of interest, we adopt the following steps: (1) select a set of products V_c from the same category c as that of the main product of interest, (2) remove items from V_c for which there are ground truth information, and (3) use LVA to predict and rank the link probabilities for substitutable and supplementary products. Figure 6 reports the qualitative results of this experiment for three product categories. In block A, the main product of interest is the Canon 5D mark II camera and the supplement is a camera bag and a photo frame, which are very logical and intuitive suggestions. For substitutes, the model recommends cameras from other brands such as Nikon D800 and Sony A7r, which are both recent products when compared to the old 5D mark II. For Men's clothing (block B), the main product of interest is a Jeans pant and the supplements for this item are t-shirt and belt. On the other hand, we observed that Amazon's recommendation for this item is a series of pants of different styles. Contrary to this, our model is able to recommend out-of-box recommendation that are both interesting and meaningful. Additionally, for substitutes, not only does our model recommend jean pants, but also a type that matches with the main product of interest that are signified by the words such "regular fit" and "straight-fit". Finally, in block C, we see the recommendations for Yamaha portable piano where the supplements are MIDI synthesizer and an album of Mozart's solo compositions, both of which are extremely unique and useful.

4 RELATED WORK

The study presented in this paper is related to the following research areas: (1) link prediction and (2) deep hybrid recommendation. We now briefly discuss these topics from a perspective of generative modeling.

Link Prediction Models: Cohn et al. [5] proposed one of the earliest generative models for link prediction in citation networks. Here, the interdependencies between documents is viewed as a mixture model that simultaneously decomposes the contingency tables associated with word occurrences and citations/links into topic factors. In [7], the authors propose a variation of this model by replacing probabilistic latent semantic analysis (PLSA) with latent dirichlet allocation (LDA) and [4] use supervised version of LDA to create relational topic model; [17] and [11] incorporate community information into topic models or predicting hyperlinks between documents. Besides textual content, researchers have also tried incorporating other heterogeneous features such as image, time, and location for link prediction [9, 18, 19, 22, 23, 30]. For example, He et al. [9] learn heterogeneous relationships between items with high-level visual features, [22] exploit both the vertical and horizontal feature hierarchy of items to capture latent relationships that could be used to better characterize user-item interactions.

Deep Hybrid Recommendation Models: The core of all recommender systems is to obtain a utility function that estimates the preference of a user towards an item. Essentially, recommender systems can be divided into three main techniques: content-based,

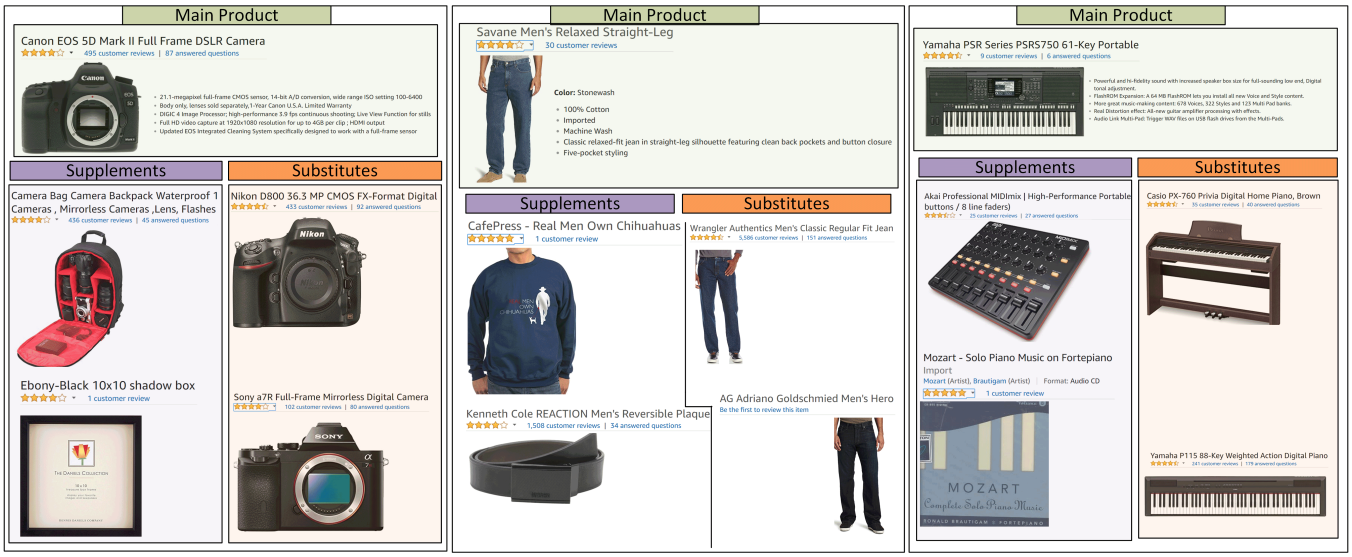


Figure 6: Out-of-Box recommendation of unseen product relations. LVA is able to provide meaningful and interesting items as substitutes and supplements.

collaborative filtering and hybrid methods. For a comprehensive summary on these techniques, readers are suggested to survey the following aritcles [1, 3, 6]. Recent trend in hybrid recommendations involve the integration of deep learning with matrix factorization (MF) models to capture both implicit and explicit features from data [8, 25, 28, 29, 31]. In [28], the authors transform the framework of collaborative topic regression (CTR) model into a deep learning framework by replacing the LDA part of CTR with stacked denoising autoencoders (SDAE) [25]. [15] propose a similar framework, but instead of using SDAE they infer content features using a variational auto encoder (VAE); [16] introduce a CF model that is purely based on VAE. A detailed survey of the latest Deep Learning based techniques for recommender systems can be found in [33].

The research that is closest to our work is a link prediction model called Sceptre that leverages the co-purchasing behavior of users to recommend substitutable and supplementary items [19]. Sceptre learns the content features of items using LDA and fits a logistic function over the document-topic features. The main strength of their model is its high prediction accuracy and scalability. Despite being a successful model, Sceptre suffers from the following shortcomings:

1. LDA is extremely inferior in learning meaningful features from short reviews [12, 24].
2. Sceptre restricts the number of topics in LDA with predefined hierarchy of product categories, which leads to problems such as overfitting and producing noisy word clusters.
3. The proposed model is not personalized; in the sense, the recommendations provided by Sceptre is same for all users.

Our model overcomes the drawbacks of Sceptre in the following ways. First, LVAE uses variational autoencoders (VAE) [13] to learn the content features of items, which overcomes the problems associated with LDA and results in better link prediction accuracy. Second, LVAE is extended to CLVAE, which induces personalization into the prediction model. Third, unlike Sceptre, which alternates between learning the parameters of LDA using variational inference and fitting a logistic model over the document-topic distribution, LVAE follows a full Bayesian approach, which leads to better approximation.

5 CONCLUSION AND FUTURE WORK

In this paper, we understand the relationship between items in Amazon e-commerce domain to recommend auxiliary items that can serve as substitutes or supplements to the main item of interest. Formally, given a network of items, reviews and their relationship (i.e., substitutes and supplements), the goal of this paper is to learn the latent features of items that are indicative of this relationship. To achieve this, we propose a generative deep learning model called Linked Variational Autoencoder (LVA) that predicts the relationship between items in terms of substitutes and supplements. This is achieved by learning the latent features of items by conditioning on the observed relationship between items. We then extend LVA with probabilistic matrix factorization (PMF) to create CLVA model that combines collaborative- and content-based information to provide personalized recommendation. Using a rigorous series of experiments, we show that LVA produces a very high link prediction accuracy of over 92% on all datasets and performs exceptionally well in cold start scenario. Additionally, the personalization induced in the form of collaborative filtering boosts the performance of LVA in recommending substitutes and supplementary items on a user level. In the future, we plan to extend our work by incorporating the following extensions: (a) leverage auxiliary information in the form of images, (b) extend our link prediction task to predict supplements and substitutes based on the style-based features of items and (c) model more complex dependencies to capture sequence of relations between products.

6 ACKNOWLEDGMENTS

This work was supported by the Office of Naval Research (ONR) grant N00014-17-1-2605.

REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. 2015. Context-aware recommender systems. In *Recommender systems handbook*. Springer, 191–226.

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[3] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. 2013. Recommender systems survey. *Knowledge-based systems* 46 (2013), 109–132.

[4] Jonathan Chang and David M Blei. 2009. Relational topic models for document networks. In *International conference on artificial intelligence and statistics*. 81–88.

[5] David A Cohn and Thomas Hofmann. 2001. The missing link-a probabilistic model of document content and hypertext connectivity. In *Advances in neural information processing systems*. 430–436.

[6] Michael D Ekstrand, John T Riedl, Joseph A Konstan, et al. 2011. Collaborative filtering recommender systems. *Foundations and Trends® in Human–Computer Interaction* 4, 2 (2011), 81–173.

[7] Elena Erosheva, Stephen Fienberg, and John Lafferty. 2004. Mixed-membership models of scientific publications. *Proceedings of the National Academy of Sciences* 101, suppl 1 (2004), 5220–5227.

[8] 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.

[9] Ruining He, Charles Packer, and Julian McAuley. 2016. Learning compatibility across categories for heterogeneous item recommendation. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE, 937–942.

[10] 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.

[11] Zhiting Hu, Junjie Yao, Bin Cui, and Eric Xing. 2015. Community level diffusion extraction. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 1555–1569.

[12] Yohan Jo and Alice H Oh. 2011. Aspect and sentiment unification model for online review analysis. In *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 815–824.

[13] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[14] Daphne Koller and Nir Friedman. 2009. *Probabilistic graphical models: principles and techniques*. MIT press.

[15] Xiaopeng Li and James She. 2017. Collaborative variational autoencoder for recommender systems. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 305–314.

[16] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 689–698.

[17] Yan Liu, Alexandru Niculescu-Mizil, and Wojciech Gryc. 2009. Topic-link LDA: joint models of topic and author community. In *proceedings of the 26th annual international conference on machine learning*. ACM, 665–672.

[18] Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 165–172.

[19] Julian McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 785–794.

[20] Vineeth Rakesh, Weicong Ding, Aman Ahuja, Nikhil Rao, Yifan Sun, and Chandan K Reddy. 2018. A Sparse Topic Model for Extracting Aspect-Specific Summaries from Online Reviews. In *Proceedings of the Tenth ACM International Conference on The Web Conference*. ACM, 631–640.

[21] Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural networks* 61 (2015), 85–117.

[22] Zhu Sun, Jie Yang, Jie Zhang, and Alessandro Bozzon. 2017. Exploiting both Vertical and Horizontal Dimensions of Feature Hierarchy for Effective Recommendation. In *AAAI*. 189–195.

[23] Ivan Titov and Ryan McDonald. 2008. A joint model of text and aspect ratings for sentiment summarization. *proceedings of ACL-08: HLT* (2008), 308–316.

[24] Ivan Titov and Ryan McDonald. 2008. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th international conference on World Wide Web*. ACM, 111–120.

[25] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research* 11, Dec (2010), 3371–3408.

[26] Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 448–456.

[27] Hao Wang, Xingjian Shi, and Dit-Yan Yeung. 2017. Relational Deep Learning: A Deep Latent Variable Model for Link Prediction.. In *AAAI*. 2688–2694.

[28] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1235–1244.

[29] 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.

[30] Suhang Wang, Jiliang Tang, Yilin Wang, and Huan Liu. 2015. Exploring Implicit Hierarchical Structures for Recommender Systems.. In *IJCAI*. 1813–1819.

[31] Suhang Wang, Yilin Wang, Jiliang Tang, Kai Shu, Suhas Ranganath, and Huan Liu. 2017. What your images reveal: Exploiting visual contents for point-of-interest recommendation. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 391–400.

[32] Zihan Wang, Ziheng Jiang, Zhaochun Ren, Jiliang Tang, and Dawei Yin. 2018. A Path-constrained Framework for Discriminating Substitutable and Complementary Products in E-commerce. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 619–627.

[33] Shuai Zhang, Lina Yao, and Aixin Sun. 2017. Deep learning based recommender system: A survey and new perspectives. *arXiv preprint arXiv:1707.07435* (2017).

[34] Jiaqian Zheng, Xiaoyuan Wu, Junyu Niu, and Alvaro Bolivar. 2009. Substitutes or complements: another step forward in recommendations. In *Proceedings of the 10th ACM conference on Electronic commerce*. ACM, 139–146.