Generation Meets Recommendation: Proposing Novel Items for Groups of Users

Thanh Vinh Vo
National University of Singapore
vinhvt@comp.nus.edu.sg

Harold Soh
National University of Singapore
harold@comp.nus.edu.sg

ABSTRACT

Consider a movie studio aiming to produce a set of *new* movies for summer release: What types of movies it should produce? Who would the movies appeal to? How many movies should it make? Similar issues are encountered by a variety of organizations, e.g., mobile-phone manufacturers and online magazines, who have to create new (non-existent) items to satisfy groups of users with different preferences. In this paper, we present a joint problem formalization of these interrelated issues, and propose generative methods that address these questions simultaneously. Specifically, we leverage on the latent space obtained by training a deep generative model—the Variational Autoencoder (VAE)—via a loss function that incorporates both rating performance and item reconstruction terms. We use a greedy search algorithm that utilize this learned latent space to jointly obtain K plausible new items, and user groups that would find the items appealing. An evaluation of our methods on a synthetic dataset indicates that our approach is able to generate novel items similar to highly-desirable unobserved items. As case studies on real-world data, we applied our method on the MART abstract art and Movielens Tag Genome datasets, which resulted in promising results: small and diverse sets of novel items.

CCS CONCEPTS

• **Information systems → Recommender systems**; • **Computing methodologies → Learning latent representations**;

KEYWORDS

Novel Item Recommendation; Deep Generative Models; Group Recommendation; Group Formation

1 INTRODUCTION

Automated recommenders [31] have become a mainstay of modern information retrieval and e-commerce systems. By suggesting
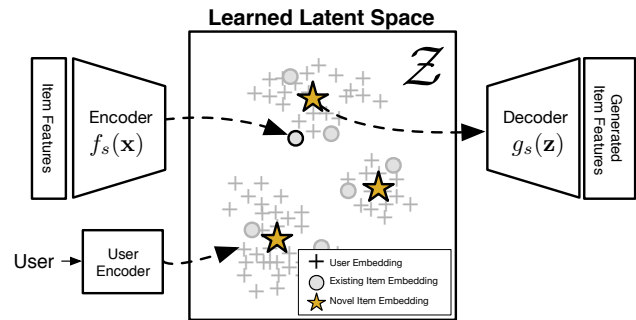
Figure 1: An Overview of Generating New (Non-Existent) Items for Groups of Users. Our approach relies on two related components: (i) a shared latent representation space learnt from user-item rating data and (ii) a pair of functions: the encoder $f_s(\mathbf{x})$ that maps item features x to their respective latent representations/embeddings z, and a decoder $g_s(\mathbf{z})$ that performs the inverse operation, i.e., maps embeddings z to item features x. We exploit this space to obtain novel items with high predicted desirability through greedy weighted maximum coverage, and then generate item features via the decoder.

related items (e.g., news articles [24], music [5], movies [37]), recommender systems filter away irrelevant data, enabling users to efficiently locate items of interest. However, current work has focussed primarily on providing recommendations of *existing items* to consumers (either personalized to individuals or groups [29]). In this work, we take on the problem of recommending or proposing *new* (*non-existent*) items that could be *created* to satisfy users.

New item recommendation would be beneficial for item *makers*—designers, producers, and manufacturers—who generate novel items for the marketplace. For example, mobile phone makers often produce a set of phone models because different users have different preferences; one group of users may prefer a phone with a large touch-screen, whilst another group may prefer a smaller, lighter, device. For item makers, important questions include: Is it possible to craft a set of new phone models that appeal to the widest possible number of customers? Which users would each phone design attract? And how large is each of these user groups? These questions are currently *not* answered by existing recommender systems research and may seem intractable at first glance.

In this paper, we take a first step towards a new class of recommender systems: item recommendation for item makers. We propose a principled approach for addressing the aforementioned questions: first, we formalize the problem and show it to be a variant of maximum coverage, which is NP-hard in general. We then

present an approximate scheme for generating plausible K items that are predicted to be appealing to different groups of users.

Our key insight is to leverage upon low-dimensional latent spaces (Fig. 1), which can be efficiently searched in a greedy manner (with a $1 + \frac{1}{\epsilon}$ approximation guarantee). To learn this low-dimensional space, we propose two models: (i) a linear model similar to matrix factorizers [18] and (ii) deep generative model, specifically, a collaborative variant of the Variational Autoencoder [17, 22] that is trained both to reconstruct ratings as well as observed item features. Both models enable projection of existing items into the latent space, as well as the generation of item features from "imaginary" item representations.

We empirically evaluate our proposed scheme using a synthetic dataset comprising groups of users with different "ideal" items that are removed prior to model training. Results show that the deep model is able to recreate the missing ideal items with higher fidelity compared to the linear model. This results in better eventual coverage of the space. We also present results characterizing the sensitivity of the linear and deep models to the number of existing items, and describe the coverage of generated itemsets of varying sizes. As case-studies using real-world data, we used our methods to propose new art using the MART dataset [39] and movies using the Movielens Tag Genome dataset [37]. The set of resultant movie genomes is intuitive and diverse in nature; the top-3 generated movies are: (i) a sentimental/touching drama, (ii) a narrated social commentary with elements of dark-humor, and (iii) a big-budget action sci-fi film.

To summarize, this work makes the following key contributions:

- A formalization of the problem of recommending new items that can be created to satisfy different groups of users;
- A solution technique using greedy maximum coverage in low-dimensional latent spaces learned via generative models;
- An empirical evaluation of our techniques on synthetic data and a demonstration of the approach on the real-world Movelens Tag Genome dataset.

From a broader perspective, this work takes the first crucial step towards intelligent item proposal for producers rather than consumers, e.g., we envision our proposed method can assist designers in the creative process. We discuss the limitations of the current approach and posit additional future directions in the concluding section of this paper. We hope this work will encourage others in the community to address this key problem, and look forward to a range of interesting solutions.

2 BACKGROUND AND RELATED WORK

Research in recommender systems have flourished in recent years and we refer readers to survey articles [9, 18, 31] for comprehensive reviews. Briefly, recommendation models have been proposed for a variety of items, from movies [4] to tours [2], and in a variety of contexts, e.g., with user and item features [27, 34, 36], and in sequential settings [30, 35]. In this work, we bring together elements from group formation/recommendation and deep learning, which we briefly review below.

Group Recommendation. Unlike standard individual user recommendation, group recommenders focus on suggesting items to *user*

groups [13, 29], e.g., a tour [2] or a movie that all members consume together. Because individual users in the group may have different preferences, group recommenders attempt to balance the interests of all group members. This is typically achieved via two techniques: preference aggregation (e.g., [40]) or score aggregation (e.g., [3]). In the former, recommendation is performed using a single prototype user obtained by combining the ratings of group members. In score aggregation, individual recommendations are combined via aggregation, misery minimization, or game theoretic techniques [1, 13, 29, 42]. Recent work [41] has also applied generative modeling, specifically a latent Dirichlet allocation (LDA), to better capture individual differences and influence.

Group Formation. As a complementary problem to group recommendation, group formation aims to construct groups that would be maximally satisfied with an itemset [26, 32, 38]. Strategic recommendation aware group formation has been shown to be NP-hard under both least misery and aggregate voting criteria [32].

Deep Models for Recommendation. The success of deep learning [7, 21] in a variety of areas (e.g., computer vision [20] and speech recognition [11]) has inspired similar deep neural models for recommendation [15]. Examples include auto-encoders for collaborative filtering [33], and recurrent neural networks for sequential recommendation [35]. Very recent work [22, 23] have independently modified a deep generative model—the Variational Autoencoder (VAE) [17]—for recommendation, but with the purpose of modeling the underlying structure of observed data for existing items.

Summary of Key Differences. Our paper differs from the above prior work in several notable ways. Previous work on collaborative VAEs focus on making better item recommendations (e.g., under implicit feedback) but here, we focus on the *generative capabilities* of collaborative VAEs to *construct* plausible new items; previous recommender systems research has not considered nor evaluated this use-case. Our setting is also unlike prior work in group recommendation: the groups are not known in advance but emerge to comprise users that share an interest in a proposed novel item. As such, it is similar in spirit to group formation and our latent space searching problem is also NP-hard, but our formulation is distinct since we attempt to find novel item embeddings.

3 PROBLEM FORMULATION

In this section, we give a precise problem formalization for novel item generation for item makers; we first present an idealized maximization problem that is impractical to solve. We then make reasonable simplifying assumptions, leading to a practical variant that we address in this work.

Given a set of users $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$, we would like to find a set K of *novel* items $\mathcal{V} = \{v_1, v_2, \ldots, v_K\}$ that will be appealing to the users, i.e., that will maximize user ratings given by a rating function $r_{uv} = r(u, v)$. We assume that $\mathcal{V} \subseteq \mathcal{S}$, i.e., a subset of possible items $\mathcal{S}$. Furthermore, we want novel items that are highly-rated, above a given threshold τ. We say an item *covers* a user u if $r_{uv} > \tau$. Given these elements, we would like to solve the following

optimization problem which aggregates covered user ratings,

$$\text{argmax}_{\mathcal{V}} \sum_{u \in \mathcal{U}} \sum_{v \in \mathcal{V}} r(u, v) c_{uv} \tag{1}$$

subject to:

$$c_{uv} \in \{0, 1\} \qquad \text{(if } c_{uv} = 1 \text{ then } u \text{ is covered by } v\text{)} \tag{2}$$

$$\sum_{v \in \mathcal{V}} c_{uv} = 1 \qquad \text{(each user is covered by one item)} \tag{3}$$

$$|\mathcal{V}| = K \qquad\qquad\quad (K \text{ items are selected}) \tag{4}$$

Attempting to solve (1) above results in immediate practical challenges; the set of potential items $\mathcal{S}$ is unknown and we do not typically have access to the true rating function $r(u, s)$.

3.1 A Latent-Space Formulation

To make progress, we leverage upon latent real vector representation spaces. We assume that each item $s \in \mathcal{S}$ can be represented as a latent real vector (an embedding) $\mathbf{z}_s \in \mathcal{Z} \subseteq \mathbb{R}^{d_z}$. Although $\mathbf{z}_s$ is unobserved, each item is associated with a set of observed features $\mathbf{x}_s \in \mathcal{X} \subseteq \mathbb{R}^{d_x}$. For example, these features could be movie tags/descriptions or mobile phone properties (e.g., CPU, brand, screen size).

We are typically interested in obtaining a novel product's features $\mathbf{x}_v$ rather than its latent representation. As such, we assume that the item features $\mathbf{x}_v$ can be *decoded* or *generated* from $\mathbf{z}_v$ via a function $\mathbf{x}_v = g_s(\mathbf{z}_v)$. In other words, given g_s, finding an item's latent representation is equivalent to finding its features.

Likewise, we let each user be associated with a latent representation sharing the same space $\mathbf{z}_u \in \mathcal{Z}$. Users can also be associated with features $\mathbf{x}_u$ if available, but we do not focus on learning user features in this work. Crucially, the users and items are related by a pre-specified rating function $\hat{r}(\mathbf{z}_u, \mathbf{z}_s)$, e.g., the dot-product $\hat{r}(\mathbf{z}_u, \mathbf{z}_s) = \mathbf{z}_u^\top \mathbf{z}_s$. The key point is that since we do not know the true rating function r, we specify a convenient parametric form and learn the space $\mathcal{Z}$ that complies to observed ratings under $\hat{r}$.

Definition 1 (MCNIP: The Maximum Coverage Novel Itemset Problem). *Given a specified rating function $\hat{r}(\mathbf{z}_u, \mathbf{z}_s)$, and a dataset $\mathcal{D} = \{d^{(k)} = (u, i, \mathbf{x}_i, r_{ui})^{(k)}\}_{k=1}^{|\mathcal{D}|}$ of previously rated existing items $\mathcal{I} \subseteq \mathcal{S}$ with associated features $\mathbf{x}_i$, we seek the latent representations $\mathbf{z}_v$, and the associated features, $\mathbf{x}_v$ for novel covering items $v \in \mathcal{V}$, where $\mathcal{V}$ is the solution to the following optimization problem:*

$$argmax_{\mathcal{V}} \sum_{u \in \mathcal{U}} \sum_{v \in \mathcal{V}} \hat{r}(\mathbf{z}_u, \mathbf{z}_v) c_{uv} \tag{5}$$

retaining the same constraints (2), (3) and (4).

We term this as the *Maximum Coverage Novel Itemset Problem* (MCNIP), due to its close relation to the well-studied maximum coverage problem (MCP) [28]. As a by-product of solving MCNIP, we also obtain user groups, i.e., each group that is covered by an item v. MCNIP can be further generalized [6] (e.g., to incorporate item costs and a total allowable budget) and we discuss possible extensions/variations in Section 6.

4 METHODOLOGY

In this section, we detail our methods for solving MCNIP. At a high-level, our solution scheme consists of three basic steps:

(1) Learn the latent representation or embedding space $\mathcal{Z}$, along with the encoder $f_s(\mathbf{x})$ and decoder $g_s(\mathbf{z})$;
(2) (Approximately) Solve Eq. (5) to obtain a set of novel items $\mathbf{z}_v \in \mathcal{Z}_V$;
(3) Generate novel item features using the learned decoder $g_s(\mathbf{z}_v)$ for each $\mathbf{z}_v \in \mathcal{Z}_V$.

Step (3) is straightforward once $g_s(\mathbf{z}_s)$ is obtained, but steps (1) and (2) require additional elaboration. We describe both steps in detail in the following subsections.

4.1 Learning the Latent Space

Our goal is to learn $\mathcal{Z}$ given pre-specified rating function $\hat{r}(z_u, z_s)$ and a dataset $\mathcal{D}$ of user-item ratings. We also aim to learn an encoding function $f_s(\mathbf{x}; \phi) : \mathcal{X} \rightarrow \mathcal{Z}$ and a decoding function $g_s(\mathbf{z}; \theta) : \mathcal{Z} \rightarrow \mathcal{X}$. To reduce clutter, we drop the explicit dependence on the parameters when it is clear from context. For ease of exposition, the following descriptions assume dot-product-based latent spaces; the predicted ratings are a function f_r of the inner product between the user and item latent representations,

$$\hat{r}(\mathbf{z}_u, \mathbf{z}_s) = f_r(\mathbf{z}_u^\top \mathbf{z}_s). \tag{6}$$

but other reasonable rating functions can be easily substituted into the framework presented. For example, our preliminary experiments have found Euclidean spaces to also work well, $\hat{r}(\mathbf{z}_u, \mathbf{z}_s) = f_r(-\|\mathbf{z}_u - \mathbf{z}_s\|_2)$.

In the following, we describe two models where the encoder and decoder are either both linear or both nonlinear. Conceptually, we find f_s and g_s (or equivalently, the function parameters θ and ϕ) by minimizing a total loss $\mathcal{L}$ that composed of three terms:

$$\mathcal{L} = \text{Rating Loss} + \text{Reconstruction Loss} + \text{Regularizer}$$
$$= \mathcal{L}_r(\theta) + \lambda_1 \mathcal{L}_g(\phi) + \lambda_2 \mathcal{R}(\theta, \phi) \tag{7}$$

where λ_1 and λ_2 weigh the contribution of each loss component. Intuitively, the rating loss captures how useful the latent representations are for rating prediction; the reconstruction loss summarizes how well item features can be reconstructed from the latent vectors; and the last term applies regularization to avoid overfitting the dataset.

4.2 Linear Encoding and Decoding

We begin with a straightforward model whereby both f_s and g_s are linear functions,

$$\mathbf{z}_s = f_s(\mathbf{x}_s) = \mathbf{M}_\theta \mathbf{x}_s \tag{8}$$

$$\mathbf{x}_s = g_s(\mathbf{z}_s) = \mathbf{M}_\phi \mathbf{z}_s \tag{9}$$

and $\mathbf{M}_\theta$, $\mathbf{M}_\phi$ are both real-valued matrices. When the rating function is $\hat{r}(\mathbf{z}_u, \mathbf{z}_s) = \mathbf{z}_u^\top \mathbf{z}_s$, this collaborative linear model (LM) is similar to hybrid matrix factorization (e.g., the Matchbox recommender [36]), but *with an additional generative component*.

The parameters $\theta = \mathbf{M}_\theta$ and $\phi = \mathbf{M}_\phi$ can be obtained by minimizing a loss of the form given in Eq. (7), which we specify in turn. First, the choice of rating loss would depend on the the type of

ratings observed. For example, if $\mathcal{D}$ contained binary ratings (e.g., "likes" or purchases), we could use the cross-entropy loss:

$$\mathcal{L}_r(\theta) = - \sum_{d \in \mathcal{D}} r_{ui} \log(\tilde{r}_{ui}) + (1 - r_{ui}) \log(1 - \tilde{r}_{ui}) \quad (10)$$

where $\tilde{r}_{ui}$ are the predicted ratings for observed items,

$$\hat{r}(\mathbf{z}_u, \mathbf{z}_i) = \sigma(\mathbf{z}_u^\top \mathbf{z}_i) = \sigma(\mathbf{z}_u^\top \mathbf{M}_\theta \mathbf{x}_i) \quad (11)$$

and σ is the sigmoid function. For real-valued ratings (e.g., "stars"), a squared error loss $\mathcal{L}_r(\theta) = \sum_{d \in \mathcal{D}} \|r_{ui} - \tilde{r}_{ui}\|_2^2$ may be more appropriate.

Similarly, the type of observed item features would dictate the form of the reconstruction loss. For example, the cross-entropy loss would apply to item features that are binary or bounded in $[0, 1]$ (e.g., grayscale image pixels):

$$\mathcal{L}_g(\phi) = - \sum_{d \in \mathcal{D}} \sum_j x_{ij} \log(\tilde{x}_{ij}) + (1 - x_{ij}) \log(1 - \tilde{x}_{ij}) \quad (12)$$

where x_{ij} and $\tilde{x}_{ij}$ are the j-th element of the observed $\mathbf{x}_i$ and reconstructed item feature $\tilde{\mathbf{x}}_i = \mathbf{M}_\phi \mathbf{z}_i$, respectively.

As a specific example of combining these elements for a particular application, consider a model for generating Movielens tag genomes (with tag "strengths" in $[0, 1]$) given a dataset of binary (or soft-binary) ratings; we can minimize the loss:

$$\mathcal{L}(\theta, \phi) = \sum_{d \in \mathcal{D}} \Bigg[- \big(r_{ui} \log(\tilde{r}_{ui}) + (1 - r_{ui}) \log(1 - \tilde{r}_{ui}) \big) -$$
$$\lambda_1 \Big(\sum_j x_{ij} \log(\tilde{x}_{ij})(1 - x_{ij}) \log(1 - \tilde{x}_{ij}) \Big) \Bigg] +$$
$$\lambda_2 \Big[\|\mathbf{M}_\theta\|_F + \|\mathbf{M}_\phi\|_F \Big] \quad (13)$$

where we have combined (10) and (12) with a regularizer comprising a Frobenius norm for each matrix parameter, $\|\mathbf{M}\|_F = \sqrt{\sum_i \sum_j |M_{ij}|^2}$.

4.3 Nonlinear Encoding and Decoding with Deep Generative Modeling

In this section, we derive a nonlinear model based on the variational autoencoder (VAE) [17][1]. Although more technically involved than the linear version above, the principal ideas are not entirely dissimilar: the VAE also makes use of an encoder f_s and decoder g_s, trained under a loss function to best reconstruct the items and ratings. A key difference is that these functions are now neural networks, which enable the model to capture more complex nonlinear relationships between the latent embedding $\mathbf{z}_s$ and features $\mathbf{x}_s$.

The second key difference is that, unlike the deterministic linear model in the previous section, the VAE is derived under a probabilistic framework. As a guide, model development begins under the standard generative scheme where the objective is to find parameters θ that maximize the expected log probability of the dataset under the data distribution,

$$\text{argmax}_\theta \mathbb{E}_{\text{data}}[\log p_\theta(d)] \quad (14)$$

where $d = (u, i, \mathbf{x}_i, r_{ui})$ is an observed datum with a user u, an item i with features $\mathbf{x}_i$, and the rating r_{ui}. As we will see, the decoder is

part of the specification of $\log p_\theta(d)$, and the encoder emerges as a means to perform (variational) inference.

To begin, we specify the log-distribution within Eq. (14),

$$\log p_\theta(d) = \log \mathbb{E}_{p(\mathbf{z}_i)}[p(r_{ui}|\mathbf{z}_u, \mathbf{z}_i) p_\theta(\mathbf{x}_i|g_s(\mathbf{z}_i))] \quad (15)$$

where $g_s(\mathbf{z}_i)$ is the decoder neural network, typically with nonlinear layers. Unfortunately, computing $\log p_\theta(d)$ requires marginalizing out the latent variables $\mathbf{z}_i$, which is intractable in this setting [2]. To get around this problem, we introduce the *recognition* or *inference model*, $q_\phi(\mathbf{z}_i|\mathbf{x}_i)$ and perform approximate inference by maximizing the variational or evidence lower bound (ELBO),

$$\ell(\theta, \phi; d) \le \log p_\theta(d), \quad (16)$$

where

$$\ell(\theta, \phi; d) = \mathbb{E}_{q_\phi(\mathbf{z}_i|\mathbf{x}_i)}[\log p(r_{ui}|\mathbf{z}_u, \mathbf{z}_i)] + \mathbb{E}_{q_\phi(\mathbf{z}_i|\mathbf{x}_i)}[\log p_\theta(\mathbf{x}_i|\mathbf{z}_i)]$$
$$- \mathbb{D}_{\text{KL}}[q_\phi(\mathbf{z}_i|\mathbf{x}_i)||p(\mathbf{z}_i)]. \quad (17)$$

The above can be seen as the expectation under the inference model with a KL divergence term that measures how different $q_\phi(\mathbf{z}_i|\mathbf{x}_i)$ from the prior $p(\mathbf{z}_i)$. Alternatively, it is the negative loss per datum; each of the three terms on the RHS correspond to the rating loss, reconstruction loss, and regularizer, in Eq. (7).

In contrast to the linear deterministic model in Sec 4.2 and similar to [34], each item is encoded to a probability distribution, $q_\phi(\mathbf{z}_i|\mathbf{x}_i)$, rather than a vector. Here, we set q_ϕ to be a multivariate normal,

$$q_\phi(\mathbf{z}_i|\mathbf{x}_i) = \mathcal{N}(\mathbf{z}_i|\mu = f_s^\mu(\mathbf{x}_i), \Sigma = \text{diag}(\exp(f_s^\sigma(\mathbf{x}_i)))) \quad (18)$$

where f_s^μ and f_s^σ give the mean and covariance of the distribution respectively, and the function diag($\mathbf{b}$) returns a matrix with $\mathbf{b}$ along the diagonal. The pair f_s^μ and f_s^σ forms the encoder.

Similar to the linear model, different rating and reconstruction likelihoods might be applied depending on the type of ratings and item features. As an example, for binary features, we could apply a Bernoulli likelihood:

$$\log p_\theta(\mathbf{x}_i|\mathbf{z}_i) = \sum x_{ij} \log(\tilde{x}_{ij}) + (1 - x_{ij}) \log(1 - \tilde{x}_{ij}) \quad (19)$$

where $\tilde{\mathbf{x}}_i = g_s(\mathbf{z}_i)$. The log-likelihood above has the same form as the cross-entropy loss given in (12). Applying a Normal likelihood results in a form identical to the squared error loss, up to a multiplicative factor and additional constant terms. The relationship between popular likelihoods and corresponding losses is well-known, and we omit further description here.

To train the model, we use the Stochastic Gradient Variational Bayes (SGVB) estimator, a differentiable unbiased estimator of the lower bound with smaller variance compared to direct Monte-Carlo estimation [17]. Using the SGVB amounts to using an approximate lower bound constructed from samples $\mathbf{z}_i^{(l)}$,

$$\tilde{\ell}(\theta, \phi; d) = \frac{1}{L} \sum_l (\log p_\theta(\mathbf{x}_i|\mathbf{z}_i^{(l)}) + \log p(r_{ui}|\mathbf{z}_u, \mathbf{z}_i^{(l)})) +$$
$$- \mathbb{D}_{\text{KL}}[q_\phi(\mathbf{z}_i|\mathbf{x}_i)||p(\mathbf{z}_i)] \quad (20)$$

where $\mathbf{z}_i^{(l)} = f_s^\mu(\mathbf{x}_i) + \exp(f_s^\sigma(\mathbf{x}_i)) \odot \epsilon_l$ and $\epsilon_l \sim \mathcal{N}(0, I)$. Using the auxiliary variable ϵ_l enables the random variable $\mathbf{z}_i$ to be specified as a deterministic function, and hence allows for differentiation

---

[1]We give a succinct derivation of the collaborative VAE and refer readers wanting more detail on the VAE and an alternative derivation to related work [8, 17, 22].

[2]To simplify exposition, we focus on modeling the item embeddings as random variables and point-optimize the user embeddings $\mathbf{z}_u$; extending the model to treat $\mathbf{z}_u$ as random variables is straightforward given the SGVB sampling estimator.

through the sampling process [17]. This collaborative VAE model can be trained by minimizing the negative log-likelihood across the dataset:

$$\tilde{\mathcal{L}}(\theta, \phi) = - \sum_{d \in \mathcal{D}} \tilde{l}(\theta, \phi; d) \tag{21}$$

To summarize, the inference network pair $f_s^\mu(\mathbf{x}_i)$ and $f_s^\sigma(\mathbf{x}_i)$ forms the encoder, and the network $g_s(\mathbf{z}_i)$ is the decoder. Both are obtained by minimizing the loss function given by Eq. (21) via using stochastic gradient descent. Compared to the linear model, the individual sample losses (20) have the same general form (7), but the encoder/decoder are now neural networks embedded within a probabilistic generative framework.

4.4 Latent Space Searching

Recall that we aim to find novel items that possess maximum coverage with maximum aggregate rating. Unfortunately, solving this problem exactly is impractical for large instances:

REMARK 1. *Given learned latent space $\mathcal{Z}$ and functions g_s and $\hat{r}$, MCNIP with a finite discrete set $\hat{\mathcal{S}} \subseteq \mathcal{S}$ is NP-hard.*

PROOF. (*Sketch*) The optimization problem (5) is exactly weighted MCP (a known NP-hard problem) with sets formed by the covering sets induced by each candidate item $s \in \mathcal{S}$, and weights given by the rating function $\hat{r}$. □

Fortunately, we can leverage upon a greedy algorithm for solving the weighted MCP problem, which achieves an approximation ratio of $1 - \frac{1}{e}$ and is the best possible polynomial-time algorithm, subject to $P \neq NP$ [28]. The greedy algorithm is straightforward: successively pick the candidate item that attains the maximum total rating given by uncovered users until K samples are selected. Note that if we are allowed to generate more then K items, then we can achieve a better approximation ratio relative to the optimal K, e.g., if the greedy algorithm can pick $l > 5K$, the approximation ratio increases to ≈ 0.99 [19, 28].

To utilize the greedy approach, we first discretize $\mathcal{Z}$ to obtain candidate items. This can be achieved using several different techniques. With the collaborative VAE, a simple uninformed method is to sample new items from the prior $p(\mathbf{z}_s)$. Alternatively, if we assume that highly rated novel points will be similar to existing highly items, we can sample T candidate points from $\prod_{t=1}^{T} q(\mathbf{z}_t | \mathbf{x}_t)$ where each $\mathbf{x}_t$ is selected from existing observed items $\mathcal{I}$ based on the rating, e.g., $p_r(\mathbf{x}_t | \mathcal{D}) = \text{Cat}(\boldsymbol{\alpha})$ where $\alpha_i = \frac{exp(\gamma \sum_u r_{ui})}{\sum_j \exp(\gamma \sum_u r_{uj})}$.

As one might expect, higher levels of discretization (more samples) may yield better coverage but incur higher computational cost. In our experiments, we sample candidate items from a generative model trained using the posterior $\mathbf{z}_i$'s to ensure the sampling method was equal for the LM and VAE models. More precisely, we train a Gaussian Mixture Model (GMM) to learn $p(\mathbf{z}_s | \mathcal{Z}_{\mathcal{I}})$ where $\mathcal{Z}_{\mathcal{I}}$ is the latent representation of the observed items $\mathcal{I}$, and sample T candidate items from $p(\mathbf{z}_s | \mathcal{Z}_{\mathcal{I}})$.

5 EMPIRICAL RESULTS

Our primary experimental objective was to evaluate if our proposed approach is able to generate items that are both highly-rated and
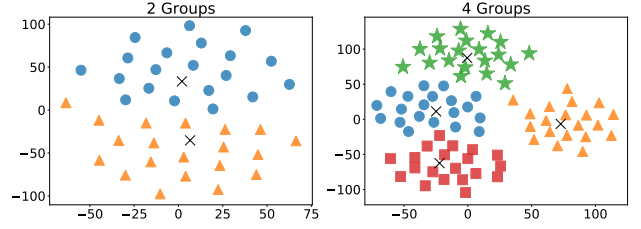

Figure 2: Groups and Observed 20-Dimensional Items in the Synthetic Dataset visualized in 2-D using t-SNE [25]. Ideal items (represented as ×'s) have the highest desirability but are unobserved. Our methods were tasked to recreate these ideal items.

novel (different from existing items). As MCNIP is a new problem, there are currently no established protocols nor datasets to evaluate solutions. Thus, we crafted a synthetic dataset with *unobserved* "ideal" (highly-desirable) items that we could compare our generated items against (described in Section 5.1). Next, to qualitatively evaluate our methods, we applied our approach to generate novel artwork and movie tag genomes [37] as case-studies using real-world ratings and feature data (Sections 5.3 and 5.3).

Model Implementation and Training. We implemented the collaborative Linear Model (LM) and Variational Autoencoder (VAE) models in Tensorflow. All ratings data were rescaled to [0, 1] and we trained both models using the cross-entropy rating loss. We used the ADAM optimizer [16] with mini-batches of size 64, and learning rate of 10^{-3} with early stopping using a held-out validation set.

5.1 Generating Missing Ideal Items

We constructed a synthetic dataset (Fig. 2) with 5000 users split evenly among K groups. For each group, we sampled N_g "observed" items $\mathbf{x}_i \in \mathbb{R}^{20}$ (20-dimensional feature vectors) from a multivariate Gaussian centered at the ideal item for that group, $\mathbf{x}_* \in \mathcal{X}_* \subset \mathbb{R}^{20}$. The observed items were rated differently depending on whether the user was from the same group or otherwise. For same group users, each item received a noisy rating inversely proportional to the distance to the ideal item $\mathbf{x}_*$. Items would obtain low ratings (0.1-0.3) from users belonging to other groups. Each user rated 20 items chosen randomly from the K groups.

To evaluate the sensitivity of our approach to the number of observed items and groups, we varied $N_g = 5, 10, 20, 50, 100$ items and $K = 2, 4$ to generate a variety of synthetic datasets. Note that the number of ratings was fixed across all the datasets since each of the 5000 users rated only 20 items. Using the observed items and ratings, we applied our methods—the underlying model was either the collaborative LM or VAE—to recreate the missing ideal items. The VAE used neural networks comprising a single 14-neuron hidden layer with ReLU activations. Each method was repeated 12 times with different random initialization of their respective parameters. Our greedy algorithm used 2×10^5 candidate points.

We measured performance by computing the RMSE of the generated covering items $\mathbf{x}_v$ to the unobserved ideal items $\mathbf{x}_*$; the matching of generated item to ideal item was performed by solving the standard linear assignment problem via a linear program.
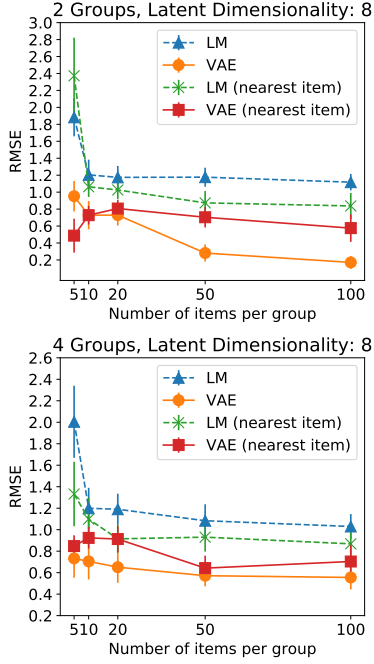
Thanh Vinh Vo and Harold Soh

Figure 3: Novel item generation on synthetic data with two user groups (top) and 4 groups (bottom), and a varying number existing items. For both the collaborative Linear Model (LM) and Variational Autoencoder (VAE) models, the larger the number of existing items, the closer the generated item to the ideal item (as measured by RMSE). Across all the trials, the VAE was able to more accurately recreate the missing items compared to LM. The "nearest item" plots show the RMSE to the nearest existing item; the higher RMSE compared to the latent missing item indicates that the models were not simply re-generating existing items.

Fig. 3 summarizes our results, showing the VAE variant outperforms LM across the different dataset parameters. The VAE is able to generate items that are significantly more similar to the missing ideal items. In addition, for the VAE, the RMSE between the generated item and the ideal item is less than any existing observed item, indicating that the VAE did not simply recreate a highly-rated existing item. Fig. 3 also shows that both models generate better items when there are more observed items, potentially because a larger number of items enables the models to learn better encoding and decoding functions.

Fig. 4 illustrates that the better generation also led to better coverage of the users (threshold $\tau = 0.7$), with VAE achieving a higher coverage than LM. As expected, with higher K, a larger proportion of users were covered, with diminishing returns at higher K. We also observed that the real coverage (computed using the real ratings), was lower than predicted. As the subplots in Fig. 4 (right) show, both models tended to overestimate ratings. It may be possible to correct for this by tuning the loss component weights or using a different ratings loss function. Nevertheless, the results demonstrate that our generation technique using the VAE was able to generate items to satisfy the different groups of users.
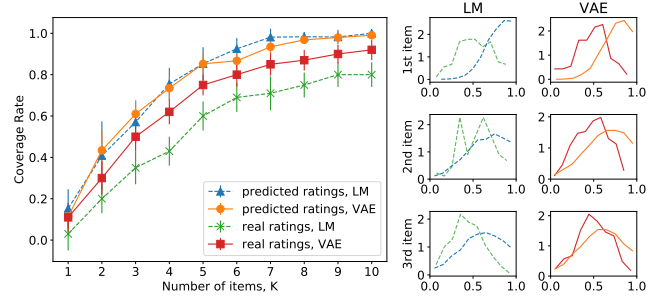

Figure 4: (Left) Coverage proportion on synthetic data (4 groups, 20 items per group) for the collaborative Linear Model (LM) and Variational Autoencoder (VAE). With higher K, a larger proportion of the users were covered ($\tau = 0.7$) with a diminishing rate of coverage. The predicted coverages were higher than the real coverages. (Right Subplots) The distributions of real and predicted ratings for both models show that the LM and VAE overestimated ratings (see left plot legend for line associations). This issue appears more problematic with the LM, and the VAE was able to attain higher real coverages.
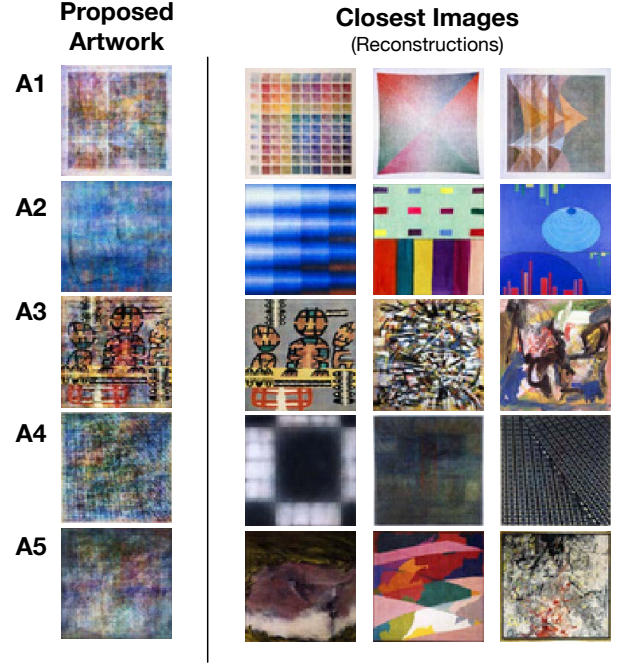

Figure 5: Generated Artwork using the MART dataset [39]. Proposed artwork is shown in the left column, with the three closest artwork (reconstructed from the true images) on the right side.

5.2 Case Study 1: Generating Abstract Art

In this first case study, we applied our methods to generating artwork. In this case, there are no "ideal" novel items for quantitative comparisons and as such, we evaluate our methods qualitatively.

We used the MART dataset [39], which comprises 500 art pieces by professional artists and 10 thousand ratings by 100 people (74 females, ages 18–65). The ratings provided are positive emotion scores on a 7-point Likert scale, which we rescaled via decoupling normalization [14]. The images were cropped and rescaled down to 64x64 pixels.

We trained the collaborative VAE, which comprised three hidden ReLU layers of 512, 768, 384 and 256 neurons respectively, to learn a 128 dimensional latent state space. The method achieves an accuracy of 62.5% on a held-out test set containing 200 ratings. We also trained a linear model for comparisons but the generated images were poor (random looking pixels) and are not shown here.

The generated artwork (with threshold $\tau = 0.9$ with $K = 5$) are shown in Fig. 5. Although each proposed image possessed similarities to the existing artwork, they were also unique; we were unable to find exact matches in the dataset. Overall, the set tends to be colorful—possibly because the ratings are based on positive emotions—but is also visually diverse; A1–A5 have different textures and patterns. The entire set covers 82% of the users, and the coverage of each painting A1 through A5 independently (the count of all users such that $r_{uv} > \tau$) were 61%, 43%, 44%, 17%, and 13% respectively. Overall, we find these results promising; the generated art pieces suggest interesting novel combinations of forms and colors that may be appealing to users.

5.3 Case Study 2: Generating Movie Genomes

This section describes results obtained from applying our approach towards recommending novel movie features. We used the Movielens Tag Genome [37] and MovieLens-1M datasets [12], which contains 3952 movies, 6040 users and 1 million ratings. Each "genome" describes a given movie via 1128 tags with relevance scores between [0, 1]. Here, we aim to generate new movie genomes, thus proposing new movies to create. Similar tags were merged—relevance scores were combined via a weighted mean using popularity scores—and non-informative tags (e.g., 'good', 'bad') were filtered away. After this preprocessing, each movie was represented by 450 tags.

We trained the VAE, which comprised three hidden ReLU layers of 500 neurons, to learn a 128 dimensional latent state space. To ascertain if the model learnt a reasonable representation for rating prediction, we compared the rating performance against matrix factorization (MF) and k-nearest neighbours (KNN). When applied to regular recommendation for existing items, the collaborative VAE achieves similar results to existing recommender systems (RMSE of ≈0.23). We then performed greedy maximum coverage with $K = 5$ and 8×10^5 candidate points. The linear model, while able to produce reasonable ratings, failed to generate interesting items and we focus on the VAE results.

Similar to the previous case-study, we evaluated our method qualitatively by comparing the generated movies against the most similar movies (in the latent space). The top-3 generated tags genomes (threshold $\tau = 0.8$) are shown in Fig. 6 along with similar real movies (as measured by the dot-product in the latent space). Interestingly, we find a diverse set of generated movies genomes: (GM-1) a sentimental/touching drama with a mentor and friendship; (GM-2) a narrated social commentary with elements of dark-humor and loneliness; (GM-3) a big-budget franchise action film with special effects and violence (possibly starring Arnold Schwarzenegger). The two other movies are (GM-4) a beautifully-filmed drama about culture-clash and childhood; (GM-5) an intense visually-appealing sci-fi movie with an emphasis on the future and technology.

The cover for each movie GM-1 to GM-5 independently were 84%, 72%, 36%, 72%, and 50% respectively, suggesting all movies were predicted to be highly rated by large numbers of moviegoers. Each movie additionally covers a portion of the user base that was not previously covered. The predicted coverage of the set is 94% and the top movie already attains a high predicted coverage of the users (84%); all three closest movies to the top generated movie are critically-acclaimed movies that appealed to a large audience, and as such, the model predicted GM-1 to attract most users.

Although the generated items aren't finished movies, they illustrate promising "genomes" for new movies that would appeal to a wide audience. Furthermore, they can be analyzed to determine important features: certain repeated tags across all the generated movies—"excellent script", "mentor", "story"—suggest common elements that highly-rated movies typically share.

6 DISCUSSION: SUMMARY, LIMITATIONS, AND FUTURE WORK

The results shown in the previous section highlight the promise of the overall framework. As with any new venture, there are improvements that can be made to the solution scheme, as well as extensions to the problem formulation. The method is currently sensitive to the discretization procedure and the threshold parameter τ; it is possible to generate a wide variety of interesting novel items by modifying these parameters. While this can be seen as a positive trait—a "brainstorming" process—it would be interesting to explore robust approaches. In the following, we highlight several other potential follow-up research:

Novel Item Generation Under a Budget and other Constraints. As mentioned in Section 3, the maximum coverage problem embedded within MCNIP can be extended. For example, we can consider the item maker to have a maximum budget B, and each candidate item s to incur a cost c_s. This generalization can be accommodated in a straightforward manner by formulating the latent search as a generalized maximum coverage problem and using a modified greedy algorithm [6]. A more difficult issue is to avoid proposing novel items that are infeasible, e.g., that violate physical law or current technological constraints. Potentially, this can be achieved by limiting the candidate set to only feasible items, but doing so in a computationally-efficient manner remains future work.

Complex Item Generation. In this work, we have focussed on generating real vectors/matrices. However, real-world objects (e.g., mobile-phones, recipes) are considerably more complex. Generating structured objects requires advances in generative modeling and we intend to investigate alternative models, e.g., hierarchical VAEs [43] and Generative Adversarial Networks [10]. In addition, this work focussed on using dot-product rating spaces. However, alternative rating functions may be beneficial, e.g., a Euclidean rating function would intuitively result in a more appropriate learned Gaussian spaces $\mathcal{Z}$ for interpolation and generation with VAEs.

Figure 6: Top 20 Tags for the Generated Movie Tag Genomes (left column). The three most similar movies (as computed by the dot-product in the latent space) are shown in the same row as each generated movie. Pink rows indicate tags that are shared with the generated movie. PredR(all) and PredR($> \tau$) are the predicted ratings across all the users and users covered by the movie ($r_{ui} > \tau$) respectively. Likewise, R(all) and R($> \tau$) indicate the real average ratings.

Collaborative Design with a Human-in-the-Loop. Finally, it would be interesting to study how data-driven novel item recommendation can be used for human-machine collaborative design. For example, a recent interactive photo manipulation tool [44] utilized a GAN to generate novel imagery based on human input. Similarly, we envisage that the methods proposed in this study can be utilized in a interactive design tool to help human item makers create desirable novel products. Research into interaction mechanisms and appropriate human-subject experiments could ascertain how such a tool could best aid the creative process.

7 CONCLUSION

In this work, we have taken a first step towards solving the challenging problem of new (non-existent) item generation to satisfy groups of users. Our approach consisted of learning a latent real-vector embedding space using a deep generative model, and applying greedy weighted maximum coverage to locate promising items. Our experimental results on synthetic and real-world data indicate that good novel item features can indeed be produced from rating data. The advancement of these methods would lead to a new generation of recommender systems for item makers, rather than just consumers.

REFERENCES

[1] Sihem Amer-Yahia, Senjuti Basu Roy, Ashish Chawlat, Gautam Das, and Cong Yu. 2009. Group recommendation: Semantics and Efficiency. *Proceedings of the VLDB Endowment* 2, 1 (2009), 754–765.

[2] Aris Anagnostopoulos, Reem Atassi, Luca Becchetti, Adriano Fazzone, and Fabrizio Silvestri. 2017. Tour recommendation for groups. *Data Mining and Knowledge Discovery* 31, 5 (2017), 1157–1188. https://doi.org/10.1007/s10618-016-0477-7

[3] Linas Baltrunas, Tadas Makcinskas, and Francesco Ricci. 2010. Group recommendations with rank aggregation and collaborative filtering. *Proceedings of the fourth ACM conference on Recommender systems - RecSys '10* (2010), 119.

[4] Robert M Bell, Yehuda Koren, and Chris Volinsky. 2007. The BellKor solution to the Netflix Prize A factorization model. *KorBell Teams Report to Netflix* 2 (2007).

[5] O. Celma. 2010. *Music Recommendation and Discovery in the Long Tail.* Springer.

[6] Reuven Cohen and Liran Katzir. 2008. The generalized maximum coverage problem. *Inform. Process. Lett.* 108, 1 (2008), 15–22.

[7] Li Deng and Dong Yu. 2014. Deep Learning: Methods and Applications. *Foundations and Trends® in Signal Processing* 7, 3-4 (2014), 197–387. arXiv:1309.1501

[8] Carl Doersch. 2016. Tutorial on Variational Autoencoders. (2016), 1–23. arXiv:1606.05908 http://arxiv.org/abs/1606.05908

[9] Michael D. Ekstrand, John T. Riedl, and Joseph A. Konstan. 2007. Collaborative Filtering Recommender Systems. *Foundations and Trends in Human-Computer Interaction* 4321, 1 (2007), 291–324. arXiv:ISSN 0018-9162

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

[11] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE international conference on.* IEEE, 6645–6649.

[12] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4, Article 19 (Dec. 2015), 19 pages.

[13] Anthony Jameson and Barry Smyth. 2007. The Adaptive Web. Springer-Verlag, Berlin, Heidelberg, Chapter Recommendation to Groups, 596–627. http://dl.acm.org/citation.cfm?id=1768197.1768221

[14] Rong Jin and Luo Si. 2004. A study of methods for normalizing user ratings in collaborative filtering. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval.* ACM, 568–569.

[15] Alexandros Karatzoglou and Balázs Hidasi. 2017. Deep Learning for Recommender Systems. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17).* ACM, New York, NY, USA, 396–397.

[16] Diederik Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International conference on Learning Representations.* 1–15. arXiv:1412.6980

[17] Diederik P Kingma and Max Welling. 2013. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations (ICLR).* 1–14. arXiv:1312.6114 http://arxiv.org/abs/1312.6114

[18] Y. Koren, R. Bell, and C. Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 42–49. arXiv:ISSN 0018-9162

[19] Andreas Krause and Daniel Golovin. 2014. Submodular function maximization. (2014).

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems.* 1097–1105.

[21] Yann A. LeCun, Yoshua Bengio, and Geoffrey E. Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.

[22] Xiaopeng Li and James She. 2017. Collaborative Variational Autoencoder for Recommender Systems. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '17* (2017), 305–314.

[23] 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 (WWW '18).* International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 689–698. https://doi.org/10.1145/3178876.3186150

[24] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. Personalized news recommendation based on click behavior. *Proceedings of the 15th international conference on Intelligent user interfaces - IUI '10* (2010), 31.

[25] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[26] Hamidreza Mahyar, Elahe Ghalebi K., S Mojde Morshedi, Saina Khalili, Radu Grosu, and Ali Movaghar. 2017. Centrality-based Group Formation in Group Recommender Systems. *WWW (Companion Volume)* (2017), 1187–1196. https://doi.org/10.1145/3041021.3055363

[27] Benjamin Marlin. 2004. Modeling User Rating Profiles For Collaborative Filtering. *Science* 16 (2004), 627–634. https://doi.org/10.1.1.10.1932

[28] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. 1978. An analysis of approximations for maximizing submodular set functions I. *Mathematical Programming* 14, 1 (1978), 265–294.

[29] Mark O'Connor, Dan Cosley, Joseph A. Konstan, and John Riedl. [n. d.]. PolyLens: A Recommender System for Groups of Users. *ECSCW 2001* ([n. d.]), 199–218.

[30] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized Markov chains for next-basket recommendation. *Proceedings of the 19th international conference on World wide web - WWW '10* (2010), 811.

[31] Paul Resnick and Hal R. Varian. 1997. Recommender systems. *Commun. ACM* 40, 3 (1997), 56–58. https://doi.org/10.1145/245108.245121 arXiv:1202.1112v1

[32] Senjuti Basu Roy, Laks V S Lakshmanan, and Rui Liu. 2015. From Group Recommendations to Group Formation. *SIGMOD 2015: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data* (2015), 1603–1616.

[33] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. AutoRec : Autoencoders Meet Collaborative Filtering. In *WWW'15.* International World Wide Web Conferences Steering Committee, Florence, Italy, 0–1.

[34] Harold Soh. 2016. Distance-Preserving Probabilistic Embeddings with Side Information: Variational Bayesian Multidimensional Scaling Gaussian Process. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI.* 2011–2017.

[35] Harold Soh, Scott Sanner, Madeleine White, and Greg Jamieson. 2017. Deep Sequential Recommendation for Personalized Adaptive User Interfaces. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces - IUI '17.* 589–593. https://doi.org/10.1145/3025171.3025207

[36] David Stern, Ralf Herbrich, and Thore Graepel. 2009. Matchbox : Large Scale Online Bayesian Recommendations. *Proceedings of the 18th International Conference on World Wide Web (WWW'09)* (2009), 111–120.

[37] Jesse Vig, Shilad Sen, and John Riedl. 2012. The tag genome: Encoding community knowledge to support novel interaction. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 2, 3 (2012), 13.

[38] Lin Xiao, Zhang Min, Zhang Yongfeng, and Gu Zhaoquan. 2017. Disparity-Aware Group Formation for Recommendation. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems (AAMAS '17).* International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1604–1606.

[39] Victoria Yanulevskaya, Jasper Uijlings, Elia Bruni, Andreza Sartori, Elisa Zamboni, Francesca Bacci, David Melcher, and Nicu Sebe. 2012. In the Eye of the Beholder: Employing Statistical Analysis and Eye Tracking for Analyzing Abstract Paintings. In *Proceedings of the 20th ACM International Conference on Multimedia (MM '12).* ACM, New York, NY, USA, 349–358. https://doi.org/10.1145/2393347.2393399

[40] Zhiwen Yu, Xingshe Zhou, Yanbin Hao, and Jianhua Gu. 2006. TV program recommendation for multiple viewers based on user profile merging. *User Modeling and User-Adapted Interaction* 16, 1 (2006), 63–82.

[41] Quan Yuan, Gao Cong, and Chin-Yew Lin. 2014. COM: a generative model for group recommendation. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14* (2014), 163–172.

[42] Limeng Zhang, Rui Zhou 0001, Haixin Jiang, Hua Wang, and Yanchun Zhang. 2017. Item Group Recommendation: A Method Based on Game Theory. *WWW (Companion Volume)* (2017), 1405–1411. https://doi.org/10.1145/3041021.3051149

[43] Shengjia Zhao, Jiaming Song, and Stefano Ermon. 2017. Learning Hierarchical Features from Generative Models. (2017). arXiv:1702.08396

[44] Jun Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. 2016. Generative visual manipulation on the natural image manifold. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 9909 LNCS. 597–613.