Disentangling Preference Representations for Recommendation Critiquing with β-VAE

Preksha Nema
Google Research
India
preksh@google.com

Alexandros Karatzoglou
Google Research
London, UK
alexkz@google.com

Filip Radlinski
Google Research
London, UK
filiprad@google.com

ABSTRACT

Modern recommender systems usually embed users and items into a learned vector space representation. Similarity in this space is used to generate recommendations, and recommendation methods are agnostic to the structure of the embedding space. Motivated by the need for recommendation systems to be more transparent and controllable, we postulate that it is beneficial to assign meaning to some of the dimensions of user and item representations. Disentanglement is one technique commonly used for this purpose. We present a novel supervised disentangling approach for recommendation tasks. Our model learns embeddings where attributes of interest are disentangled, while requiring only a very small number of labeled items at training time. The model can then generate interactive and critiquable recommendations for all users, without requiring any labels at recommendation time, and without sacrificing any recommendation performance. Our approach thus provides users with levers to manipulate, critique and fine-tune recommendations, and gives insight into why particular recommendations are made. Given only user-item interactions at recommendation time, we show that it identifies user tastes with respect to the attributes that have been disentangled, allowing for users to manipulate recommendations across these attributes.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Disentangling, Critiquing, β-VAE, Recommender Systems

ACM Reference Format:
Preksha Nema, Alexandros Karatzoglou, and Filip Radlinski. 2021. Disentangling Preference Representations for Recommendation Critiquing with β-VAE. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3459637.3482425

1 INTRODUCTION

Recommender systems have become essential tools for exploring content such as music, videos, films, news, merchandise and much more, distributed on apps and websites over the word wide web. The vast majority of these recommender systems act as black boxes and do not allow the user to provide immediate feedback or control the recommendations. In this work we aim to create a new method for providing recommendations that is more transparent and interpretable, while empowering the user by providing more control and the ability to explore and critique recommendations.

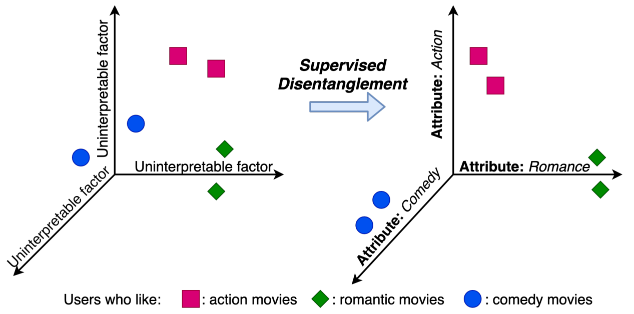


Figure 1: The above figure demonstrates the effect of supervised disentanglement for a toy example of 3-dimensional latent space. The plot on the left shows that though similar users are closer in the embedding space, the dimensions of the latent representation are uninterpretable. With supervised disentanglement we are able to associate some meaning to the dimensions. By adjusting the values in the corresponding dimensions, users will have control over their recommendations.

Recommender systems inherently depend on some kind of representation of users (and items). In state-of-the-art models, these typically come in the form of embeddings, i.e., items are represented with n-dimensional vectors generated by matrix factorization [13, 14], or from the internal hidden state of a deep network (e.g., the hidden state of an RNN trained to predict the next song in a playlist given the last n songs the user has listened to [8]).

While such representations are necessary for optimal recommendation, they have several limitations. For instance, the user cannot tune the recommendations. Standard embeddings cannot, for example, generate recommendations similar to a recipe the system already suggested, but with an ingredient replaced or removed at the user's request. Similarly, it is difficult to accurately explain recommendations, as they live in a latent space where each factor has limited interpretability. User control is desirable for a recommender system, be it through a visual interface or a conversational one, as it facilitates direct user feedback to influence recommendations. This feedback process is often referred to as *critiquing* [2].

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia

ACM ISBN 978-1-4503-8446-9/21/11.
https://doi.org/10.1145/3459637.3482425

We take a step towards flexible critiquing by training user and item representations using *supervised disentanglement.* As demonstrated in Figure 1 supervised disentanglement helps in associating certain aspects of recommendations to the dimensions of learnt user and item representations. Specifically, our model learns representations for users and items that: (i) give users fine-grained control over aspects of recommendations to support critiques (ii) provide users with explanations as to why a given recommendation is made. While novel in the recommendation domain, our work is inspired by disentangled representations in other domains. Related approaches have been used to manipulate generative image models [3] or generate text controlling certain attributes [10].

Variational Autoencoders (VAEs), particularly β-VAEs [9], are generally used to learn disentangled representations. There are two types of disentangling VAEs: *unsupervised* and *supervised.* In the former, the representations are disentangled to explanatory factors of variation in an unsupervised manner, i.e., without assuming additional information on the existence (or not) of specific aspects. The lack of supervision often results in inconsistency and instability [17]. In *supervised* disentangling a small subset of data is assumed to have side-information (i.e. a label or a tag) that is used to disentangle into meaningful factors [17, 18]. As critiquing requires user control using familiar terms/attributes, we follow this path, noting that in most recommendation settings there is sufficient side-information.

Our contribution is a novel recommendation model based on *supervised disentanglement*, modifying the β-VAE loss to learn representations that explicitly capture tangible aspects of the user preference (and item) representation in an independent set of factors. Specifically, we present a disentangling β-VAE collaborative filtering model that provides: (i) fine-grained control over recommendations (ii) explanation of recommendations, (iii) supervised disentangling using a small fraction of labeled items , (iv) state-of-the-art recommendation accuracy.

2 RELATED WORK

Autoencoders and Recommendation. Deep learning based Autoencoder architectures have been extensively adopted for collaborative filtering and recommendation models [16, 26, 32]. In particular, Sedhain et al. [26], Wu et al. [32] adopt denoising autoencoder architectures for this purpose, while Liang et al. [16] use variational autoencoders. Such models often produce state-of-the-art recommendation performance. The latent representations learnt by such architectures are generated through highly nonlinear functions. As a result, it makes standard VAE-based models difficult to control, and they cannot be used to generate explanations of recommendations.

Disentanglement. To improve the explainability and controllability of VAEs, *disentangling* the hidden representations was introduced. One of the first approaches was the β-VAE [1, 9, 29], which essentially enforces a stronger KL divergence between encoding dimensions by way of an additional constraint on the VAE objective. For instance, in the computer vision domain, β-VAE allows image representations to be disentangled across factors such as the color of an object, size of an object, or background color. Such representations are thus also more explainable as compared to VAEs.

One of the drawbacks of β-VAE is that the disentanglement factors cannot be controlled and are relatively unstable, particularly when the factors of variance are subtle [17]. β-VAE results reported in the literature are achieved using many repeated runs [1, 9, 11, 25]. This instability when using unsupervised β-VAE has also been proven theoretically [17]. Therefore subsequent works have proposed *supervised* disentanglement [18]. Either a good set of disentangling dimensions is selected using multiple runs and label information [4], or a supervised loss function is added in the β-VAE objective function [18]. Supervised disentangling methods provide control and are explainable, hence our model is inspired by these techniques, adapting them to the recommendation domain.

Similar approaches have also been applied when disentangling image representations with standard VAEs ($\beta = 1$) by utilizing a discriminator to condition the representation with respect to the explicit features of variations. For instance, Fadder Networks [15] use a discriminator to create representations that are invariant to the supervised features. These features are then concatenated with the representation generated by the autoencoder. A similar VAE-based architecture with a discriminator has been used to control sentiment and tense in text generation [10]. The disentangled features are explicitly modeled using feedback from a discriminator that aims to detect their presence.

Critiquing. A number of methods that use side information to enable critiquing in recommender systems have been recently introduced. These models allow users to tune the recommendations received across a set of provided attributes/dimensions. Notable recent models are augmented with a classifier (or second decoder) of the features over which to control the recommendation [19, 20, 31]. Adjusting the features at the classifier's output modifies the internal hidden state of the model, and leads to the filtering of recommendations to those that either exhibit or do not exhibit the requested attribute. However, this method of critiquing is quite different from our approach: Our method allows for a gradual adjustment of the degree to which the attributes are present in recommendations. Moreover, [19, 20, 31] models are limited to refining recommendations by *disabling* an aspect, whereas ours can *reinforce* an aspect as well. Second, our approach only requires a small fraction of labeled data, while this past work requires a fully labeled dataset.

Unsupervised disentanglement was also recently used to identify, and potentially use, factors of variation from purely collaborative data (i.e., data generated by user interactions with items) [22]. The main aim of this method is to increase recommendation performance. Note that this method does not allow for seamless critiquing as it is essentially an un-supervised disentangling method. As a consequence it is not clear what aspects of the data are disentangled, and a relatively elaborate intermediate step is required to identify the factors that have been. Moreover unsupervised disentangling methods are unstable to repeated runs and small changes in the data [18], hence it cannot be guaranteed that the same attributes will be disentangled, making such methods difficult to apply to real-world use cases.

3 VARIATIONAL AUTOENCODERS

At an intuitive level, user preferences can be seen as a combination of preferences over semantically meaningful attributes (such as

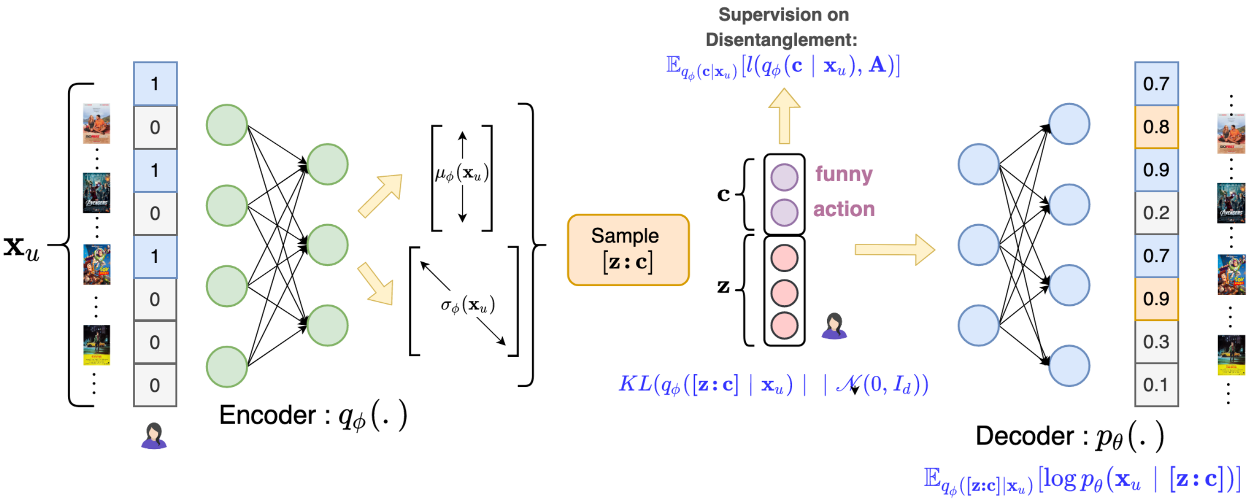

Supervision on Disentanglement:
$\mathbb{E}_{q_\phi(\mathbf{c}|\mathbf{x}_u)}[l(q_\phi(\mathbf{c} \mid \mathbf{x}_u), \mathbf{A})]$
$\mathbf{x}_u$
1
0
1
0
1
0
0
0
Encoder : $q_\phi(.)$
$\mu_\phi(\mathbf{x}_u)$
$\sigma_\phi(\mathbf{x}_u)$
Sample
$[\mathbf{z}:\mathbf{c}]$
c
z
funny
action
$KL(q_\phi([\mathbf{z}:\mathbf{c}] \mid \mathbf{x}_u) \mid \mid \mathcal{N}(0, I_d))$
0.7
0.8
0.9
0.2
0.7
0.9
0.3
0.1
Decoder : $p_\theta(.)$
$\mathbb{E}_{q_\phi([\mathbf{z}:\mathbf{c}]|\mathbf{x}_u)}[\log p_\theta(\mathbf{x}_u \mid [\mathbf{z}:\mathbf{c}])]$

Figure 2: For each user, a bag-of-word representation is passed to the encoder (ϕ), which estimates parameters of the underlying distribution. The representation is then sampled from the learnt distribution. The representation is constrained using KL divergence, and some supervision for certain preferences (e.g., funny, action). The decoder (θ) then generates the probability distribution over the list of items $\mathcal{I}$ using the sampled representation $[z : c]$. During training, we optimize to reconstruct the same set of items as in the input (blue), while at inference new sets of items (orange) that have a high probability score are recommended to the user.

funny movies, or more nuanced such as 30's film-noir) combined with latent factors, they can be learned using *collaborative filtering* with a VAE.

Notation. Let past user-item interactions be captured in matrix $\mathbf{X} \in \mathcal{R}^{|\mathcal{U}| \times |\mathcal{I}|}$, where $u \in \mathcal{U}$ is a user and $i \in \mathcal{I}$ is an item. This matrix contains a nonzero value at $\mathbf{X}_{ui}$ when user u interacted with item i. When only binary user-item interaction data is available, elements of $\mathbf{X}$ are 1 where an interaction occurred, and 0 otherwise.

Given a binary interaction matrix $\mathbf{X}$, VAEs are commonly trained using a bag-of-words representation: Each user u is represented by the items they have interacted with, i.e., row $\mathbf{x}_u$ of matrix $\mathbf{X}$. The autoencoder is then trained to reproduce the input row at the decoder output, with an intermediate d-dimensional embedding.

Encoder. A standard autoencoder is trained to reproduce the input data in an output layer via a compressed latent representation. The first part of the autoencoder which generates the latent representation is termed the *encoder*. This training process ensures that the encoding captures the most relevant information about input $\mathbf{X}$. In contrast, VAEs are based on a generative process: the encoder's aim is to estimate the parameters of the underlying distribution that the input data are sampled from. Each user u is modeled by sampling a d-dimensional latent representation $\mathbf{z}$ from the decoder with a Gaussian prior with 0 mean and diagonal co-variance matrix: $p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I}_d)$ where $\mathbf{I}_d$ is a $d \times d$ diagonal matrix. This process is data driven, with $\mathbf{z}$ being sampled from the distribution provided by the encoder. The true distribution out of which $\mathbf{z}$ is sampled is approximated by a parameterized function $q_\phi(\mathbf{z}|\mathbf{x}_u) = \mathcal{N}(\mu_\phi(\mathbf{x}_u), diag(\sigma_\phi(\mathbf{x}_u))$ as illustrated in Figure 2. Both the estimate of the mean $\mu_\phi(\mathbf{x}_u)$ and standard deviation $\sigma_\phi(\mathbf{x}_u)$ are computed by the encoder $q_\phi(\mathbf{z}|\mathbf{x}_u)$, which is implemented as a feed-forward neural network parameterized by ϕ as also illustrated in the figure 2.

Decoder. In a VAE, the decoder generates the probability distribution over the items $\mathcal{I}$ given $\mathbf{z}$ as an input: $\pi(\mathbf{z}) \propto exp(f_\theta^{dec}(\mathbf{z}))$. The likelihood function used in recommender systems [7, 8, 27, 28] is typically the multinomial likelihood:

$$p_\theta(\mathbf{x}_u|\mathbf{z}) = \sum_i \mathbf{x}_{ui} \log \pi_i(\mathbf{z}) \tag{1}$$

In the recommendation setting, where relevance of the top-k items is critical, the multinomial likelihood performs well: it pushes the outputs of non-zero entries in $\mathbf{x}_u$ towards higher values while restricting the output where $\mathbf{x}_u$ has zeroes.

Learning. The objective optimized by a VAE with respect to the model parameters of the encoder (ϕ) and decoder (θ) is:

$$L(\mathbf{x}_u, \theta, \phi) \equiv \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}_u)}[\log p_\theta(\mathbf{x}_u|\mathbf{z})] - KL\left(q_\phi(\mathbf{z}|\mathbf{x}_u)|p(\mathbf{z})\right) \tag{2}$$

Intuitively, it is the negative reconstruction error, minus the Kullback-Leibler divergence enforcing the posterior distribution of $\mathbf{z}$ to be close to the Gaussian distribution (prior) $p(\mathbf{z})$. The KL divergence is typically computed between the representation generated by the encoder $q_\phi(\mathbf{z}|\mathbf{x}_u)$ and the normal distribution $p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I}_d)$. The diagonal co-variance matrix enforces a degree of independence among the individual factors of the representation.

β-VAE is a variant of VAE that has been successfully used for learning interpretable representations of independent factors of variance in data without supervision [9]. The idea is to strengthen feature independence, increasing the weight of the KL divergence term in the VAE objective with a parameter $\beta > 1$ (a normal VAE being a special case where $\beta = 1$). β balances the latent channel

capacity (i.e., reconstruction accuracy) against independence constraints. The objective function is thus:

$$L(\mathbf{x}_u, \theta, \phi) \equiv \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}_u)}\left[log p_\theta(\mathbf{x}_u|\mathbf{z})\right] - \beta \cdot KL\left(q_\phi(\mathbf{z}|\mathbf{x}_u)|p(\mathbf{z})\right) \quad (3)$$

Supervised β-VAE then adds an additional term to the loss function to encourage factors to have specific meaning. Importantly, only a tiny fraction of preferences are required to have such labels. We present supervised disentanglement using a β-VAE next.

4 MODEL ARCHITECTURE

Our goal is to learn a representation of user preferences that allows user control over recommendations across well-defined attributes. To this end, we adapt β-VAEs. Specifically, if we have $\mathbf{A}$ attributes over which control is desired, the β-VAE is trained such that $\mathbf{A}$ of the d dimensions each map to one attribute. Then, changing the factor corresponding to an attribute leads to controllable variations in the recommended items.

Our supervision approach is a simple and yet very effective extension to VAEs for recommendation, adapted from [18]. Specifically, we modify the β-VAE objective to incorporate a classification loss over a subset of the factors $\mathbf{c}$ that we aim to disentangle over. The classification loss penalizes discrepancies between the prediction coming from factor c_i and the attribute of interest. This nudges the disentanglement for that attribute (label) to happen over the selected factor c_i, mapping this factor to the selected label. The β-VAE representation $\mathbf{z}$ is then essentially replaced by $[\mathbf{z} : \mathbf{c}]$, as illustrated in Figure 2. To elaborate, $\mathbf{c}$ represents the dimensions in the latent representation sampled from $q_\phi(\mathbf{x}_u)$ that are explicitly disentangled, whereas $\mathbf{z}$ constitutes the rest of the dimensions in the sampled latent representation. The input of the encoder remains the movies user u has seen, $\mathbf{x}_u$, and the decoder aims to reconstruct these. To reiterate, $[\mathbf{z} : \mathbf{c}]$ is sampled from the distribution learned by the encoder and thus is the representation of the user.

The objective function of our supervised β-VAE model is thus:

$$\begin{aligned} &L(\mathbf{x}_u, \theta, \phi) \\ &\quad \equiv \mathbb{E}_{q_\phi([\mathbf{z}:\mathbf{c}]|\mathbf{x}_u)}\left[\log p_\theta(\mathbf{x}_u|[\mathbf{z} : \mathbf{c}])\right] \\ &\qquad - \beta KL\left(q_\phi([\mathbf{z} : \mathbf{c}]|\mathbf{x}_u)|p([\mathbf{z} : \mathbf{c}])\right) + \gamma \mathbb{E}_{q_\phi(\mathbf{c}|\mathbf{x}_u)}\left[l(q_\phi(\mathbf{c}|\mathbf{x}_u), \mathbf{A})\right] \end{aligned} \quad (4)$$

where l is a loss function computed on the (limited) preference labels available. We use binary cross entropy loss, i.e. $l(q_\phi(\mathbf{c}|\mathbf{x}_u), \mathbf{A}) = -\sum_{i=1}^{|A|} a_i \log(\sigma(c_i)) + (1 - a_i) \log(1 - \sigma(c_i))$, where σ is the logistic function and (slightly abusing notation) c_i denotes the mean of the representation generated by the encoder for attribute a_i. It is represented in binary form to denote presence or absence of the attribute ($[0, 1]$) label. Typically there are more dimensions in $[\mathbf{z} : \mathbf{c}]$ (usually 10–100) than distinct attributes, so we limit the dimensionality of $\mathbf{c}$ to $|\mathbf{A}|$ dimensions of the representation $[\mathbf{z} : \mathbf{c}]$ for disentangling over $\mathbf{A}$ attributes.

Labels. The training labels a_i are assigned to a fraction of users: Those with particularly distinctive interactions, having predominantly viewed movies with one specific attribute. Alternatively, not explored here, they could be assigned by the users themselves.

Algorithm 1 Training supervised β-Variational Autoencoder

```
1:  Input: Set $\mathbf{X} \in \mathbb{R}^{\mathcal{U} \times \mathcal{I}}$ containing user-item interactions and a
    set of attributes $\mathbf{A}$
2:  Sample one batch (users) from $\mathbf{X}$
3:  for all $u \in \mathcal{U}$ do
4:      repeat
5:          Compute $[\mathbf{z} : \mathbf{c}]$ by sampling the output of the encoder
6:          $\tilde{y} \leftarrow$ Decoder($[\mathbf{z} : \mathbf{c}]$)
7:          if user has label $a$ then
8:              Compute gradients $\nabla L_\phi$, $\nabla L_\theta$ using Equation 4
9:          end if
10:         if user has no label then
11:             Compute gradients $\nabla L_\phi$, $\nabla L_\theta$ using Equation 3
12:         end if
13:         Update model parameters for encoder $\phi$ and decoder $\theta$
14:     until converges
15: end for
```

Learning. Training the supervised β-VAE involves learning both θ and ϕ by maximizing the objective function in Equation 4. The stochastic nature of the representation $[\mathbf{z} : \mathbf{c}]$ does not allow for direct application of backpropagation, hence the re-parametrization trick is used, whereby ϵ is sampled from $\mathcal{N}(0, \boldsymbol{I_d})$ where d the dimensionality of $[\mathbf{z} : \mathbf{c}]$ and the encoder representation is sampled $[\mathbf{z} : \mathbf{c}] = \mu_\phi(\mathbf{x}_u) + \epsilon \odot \sigma_\phi(\mathbf{x}_u)$ [12, 24]. This allows for gradients to flow through the encoder even though $[\mathbf{z} : \mathbf{c}]$ is sampled. While training we utilize a small fraction of labeled items to compute loss l, optimizing over the objective in Equation 4 (as given in Algorithm 1, Line 8). The computational overhead is thus minimal. For the items that have no labels (typically the vast majority) we optimize Equation 3 (as given in Algorithm 1, Line 11). At inference time, when generating recommendations, no label information is used.

5 EVALUATION

We demonstrate the three main goals: (1) *Supervised β-VAE generates disentangled representations without sacrificing performance*; (2) *The representation created by the supervised β-VAE can be used by the user to control and critique recommendations and also explain recommendations across the disentangled attributes*; (3) *Only a tiny fraction of labeled preference data is needed to generate the disentangled representations.* We discuss the metrics utilized.

Ranking Metrics. We utilize two ranking-based metrics, recall@k and normalized discounted cumulative gain (NDCG@k). Both metrics are computed per user. NDCG is rank-sensitive, recall@k considers equally each relevant item in the top-k positions.

$$DCG@k := \sum_{i=1}^{k} \frac{2^{\mathbb{1}[item[i] \in \mathcal{S}]} - 1}{\log(i+1)}$$

$$Recall@k := \frac{\sum_{i=1}^{k} \mathbb{1}[item[i] \in \mathcal{S}]}{min(k, |\mathcal{S}|)}$$

NDCG@k is normalized DCG@k, by dividing it by the largest possible DCG@k. Note that for much of the analysis (with the exception of the accuracy computations) we consider relevant items $\mathcal{S}$ to be the items that contain the attribute/label over which we modify the recommendations.

Dataset	Number of Interactions	Number of Users	Number of Items	Sparsity Rate
Movielens-1m	1,000,209	6,040	3,706	4.468 %
Movielens-20m	9,990,682	136,677	20,720	0.353 %
GR-Books	6,070,472	124,411	104,103	0.046 %
GR-Children	3,371,518	92,993	33,635	0.108 %
GR-Comics	2,705,538	57,405	32,541	0.145 %

Table 1: Dataset statistics (after performing all filtering). The sparsity rate indicates the fraction of cells in the complete user-item matrix with a known value.

Disentanglement Metrics. To evaluate the disentanglement obtained across the given attributes, we use the *Disentanglement* and *Completeness* metrics [5], which are briefly explained below:

Disentanglement (**D**) quantifies the extent to which each dimension in [**z** : **c**] captures at most one factor (attribute). If a single dimension encodes all the factors (attributes) then it's disentanglement score will be 0. It will be 1 if each dimension encodes only one factor (attribute). We measure the *importance* p_{aj} of the a^{th} attribute on the j^{th} dimension of [**z** : **c**] using Gradient Boosting classification. We take the input as the user (or movie) representations [**z** : **c**] and the target as the attributes associated with the respective user (or movie). Along with modeling the classifier for the given target, it measures a dimension's importance to predict an attribute. Based on the p_{aj} scores, **D** can be formally defined as:

$$H_{|\mathbf{A}|}(P_j) = -\sum_{a=0}^{|\mathbf{A}|-1} p_{aj}\log_{|\mathbf{A}|} p_{aj}, \quad D_j = (1 - H_{|\mathbf{A}|}(p_j))$$

$$\mathbf{D} = \sum_{j=0}^{d-1} \rho_j D_j \ \text{ where } \ \rho_j = \frac{\sum_{a=0}^{|\mathbf{A}|-1} p_{aj}}{\sum_{j=0}^{d-1}\sum_{a=0}^{|\mathbf{A}|-1} p_{aj}}$$

D_j is 1 − *entropy*, $(H_{|\mathbf{A}|}(P_j))$ of importance distribution of the a^{th} attribute across all dimensions d in [**z** : **c**]. The disentanglement score of the system is then a weighted average of D_j across all dimensions, where ρ_j is the dimension's relative importance.

Completeness (**C**) measures the extent to which a factor (attribute) is exclusively encoded in a given dimension j of [**z** : **c**]. For instance, for $d = 4$, and two factors ($|\mathbf{A}| = 2$), if the first two encode the first attribute and last two encode the second attribute, **C** will be 0.5 whereas **D** will be 1. Completeness is computed as follows:

$$H_d(P_a) = -\sum_{j=0}^{d-1} p_{aj}\log_d p_{aj}, \quad C_a = (1 - H_d(p_a))$$

$$\mathbf{C} = \sum_{a=0}^{|\mathbf{A}|-1} \rho_a C_a, \quad \rho_a = \frac{\sum_{j=0}^{d-1} p_{aj}}{\sum_{a=0}^{|\mathbf{A}|-1}\sum_{j=0}^{d-1} p_{aj}}$$

5.0.1 Datasets. **Movielens:** We use two subsets of this dataset: Movielens-1m and Movielens-20m [6], consisting of 1 million and 20 million interactions respectively. In the Movielens-20m dataset, we filter out users who have rated fewer than 5 movies, but do not filter any users for Movielens-1m. These cut-off thresholds have been taken from [16]. The statistics are presented in Table 1.

Cluster Label	Number of tagged movies	users	Tags included in cluster
action	1,292	9,932	action, fight-scenes, special-effects
funny	1,326	1,360	comedy, funny, goofy, very funny
romantic	1,061	1,816	destiny, feel-good, love story, romantic
sad	1,620	1,913	bleak, enigmatic, intimate, loneliness, melancholic, melancholy, reflective, sad
suspense	1,172	1,514	betrayal, murder, secrets, suspense, tense, twist-and-turns
violence	1,419	4,282	brutality, cult classic, vengeance, violence, violent

Table 2: Example clusters of tags in Movielens-20m. Tags were clustered using K-Means, as shown in the Tags column. Each cluster was manually assigned a human-readable label. We show how many movies had high relevance score for tags in each cluster, and how many users this mapped to, as predominantly interested by movies in this class.

Attribute Selection. Along with user–movie ratings, the Movielens datasets assigns 10, 381 movies with 1, 127 distinct tags. Example tags include *chase, brutal* and *funny*. Each (tag, movie) pair also has a relevance score – for instance, the movie *Toy Story* has high relevance for *children* (0.96) and for *funny* (0.70), and low relevance for *gory* (0.02) and *horror* (0.04). We extract the 100 tags with the highest mean relevance score across all movies. Within these 100 tags, a number of tags are highly correlated (such as *sad* and *melancholy*), and thus arguably have very similar semantics to users. Therefore we cluster these tags into 20 clusters using K-Means clustering (representing each tag by its relevance score $s_t \in \mathbf{R}^{10381}$).

For each movie and cluster, a new clustered-relevance score is taken as the average of relevance scores for all the tags present in the corresponding cluster. We label the movie with the cluster if its clustered-relevance score is greater than a threshold M, which is a hyperparameter. Finally, we transitively apply cluster tags to *users*: A user is assigned to a given cluster if at least half of the movies rated by this user are labeled by the respective cluster. A user is labeled as e.g. *action* if more than 50% of the movies they rated are labeled as *action movies*. Example clusters, along with the number of movies and users assigned to each, are shown in Table 2.

GoodReads: This dataset [30] contains approximately 230 million interactions from 876, 145 users' public bookshelves with 2.3 million books. Goodreads-(Children,Comics) [30] consist of similar interactions, but restricted to books in the children and comics genres respectively. As in [21], we filter out interactions with a rating of less than 4 as non-relevant, then remove users with fewer than 10 rated books, and vice versa for the books rated by fewer than 10 users for Goodreads-(Children, Comics) and 15 users for Goodreads Books. The final dataset statistics are in Table 1.

Attribute Selection. In addition to user-book interactions, the GoodReads dataset contains metadata for each book (book language, top user-generated shelf names, etc.). We use the top user-generated shelf names (such as *to-read*, *horror* and *fiction*) as attributes for disentangling. We extract the 100 most popular names, then filter these to remove uninformative tags (e.g. *one-word-title* is not an informative keyword for a book, while *humor* is informative). A shelf name was considered uninformative if it was marked as such independently by all authors of this paper. We list the attributes kept for all three GoodReads datasets in Table 3. To label users, we perform the same process as described above for Movielens-20m:

Dataset	Attribute	Number of tagged		Dataset	Attribute	Number of tagged	
		books	users			books	users
Books	adventure	5895	2951	Children	horror	1204	370
	crime	3698	606		humor	10587	24958
	fantasy	8017	11655		mystery	4094	7293
	horror	2730	232		romance	1582	704
	humor	4504	1085	Comics	adventure	9339	22,809
	mystery	5887	2110		horror	6239	9,725
	romance	5922	4100		humor	9,473	15,216
	sci-fi	4040	721		mystery	5,916	7,259
	thriller	3691	746		romance	8,625	10,184
					sc-fi	9,038	16,791

Table 3: Attributes selected for disentangling for GoodReads-(Books, Children and Comics dataset). We list the number of users, and books for which the attribute was listed in the popular shelf.

Features	Critiquing Based Methods [19, 20, 31]	Disentanglement for Recommendations [22]	Ours
Critique representations:	Binary	Continuous	Continuous
Partially-labeled dataset	No	No	Yes
Type of User control	Binary	Continuous	Continuous
Pre-defined User controls	Yes	No	Yes

Table 4: Qualitative Comparison with Critiquing and Disentanglement based methods.

That is, we label the user with a given shelf-name if at least half of the books (rated positive) by him or her belong to that shelf-name. The number of positive labels has been listed in Table 3 for both books and users. We select the negative labels using the same method as described for the movies domain.

5.0.2 Implementation Details. We divide the users into train, validation and test splits: Validation and test splits consist of 10% of the users, across all datasets. For each user in the validation and test split, we use only 80% of the items rated by them to learn the user representation. The remaining 20% is used to evaluate the model's performance. This strategy is similar to that used by [16]. For all experiments, the user's latent representation is restricted to 16 dimensions. The encoder and decoder consist of two layers with $[500, 300]$ and $[300, 500]$ hidden units respectively, each with ReLu activation. We conduct hyper-parameter tuning to identify β and γ values from $[1, 10, 50, 100]$ and $[200, 500, 1000]$ respectively. The threshold M to identify movies where the attribute is present for Movielens-20m , and MovieLens-1m is taken as 0.5 and 0.4 respectively. All the models are run up to 50 epochs. We select the best model, based on its performance on the validation dataset for both NDCG@100 and Disentanglement score. We select less than 1% of users for supervised β-VAE using stratified sampling. We present the results and analysis on the test dataset, across all datasets in the subsequent sections. Also, for the baselines we take the same hyperparameters as mentioned in [16].

5.1 Qualitative Comparison

As discussed earlier, our model uses supervised disentanglement to enable critiquing for VAE based models for collaborative filtering. Therefore, we compare our approach to recent work on disentanglement for recommendations [22] and critiquing based systems [19, 20, 31] across desired features as given in Table 4.

First, note that [22] aims to disentangle user behavior and preferences in a hierarchical and unsupervised fashion, leading to state-of-the-art recommendation performance and interpretable representations. In comparison to [22], we primarily disentangle user representations to provide controllable levers to users to critique the representations: The ability to encode *specific* subtle preferences to enable user control is particularly crucial for critiquing. For example, in the case of movies, it is crucial to disentangle concepts like funniness, as comedy preferences are very common. Such subtle preferences are not guaranteed to be disentangled in [22]. Thus we use supervision to disentangle predefined attributes. We achieve this with a tiny fraction of labeled-data.

Methods such as [31] are primarily focused on keyword-based critiques with binary nature, i.e., if recommended items should, or should not, have the given feature. Such methods could limit usefulness of critiquing in domains like movies or books, where continuous attributes are common. For example, consider "action movies": There is Bad Boys (light action) to John Wick (intense action), and these may need different treatment. Their critiqued recommendations depend on the energy redistribution function (re-computing the latent representation). However, in our method, dimensions are associated with specific meanings so the change in latent representation is achieved by simply adjusting the associated dimension, reducing inference overhead. To reiterate, we achieve meaningful representations with very few labels. Moreover, in past work critiquing has only been explored in one direction (disabling the feature). We enable the user to critique in both directions. Note that though there is scope for incremental critiquing [23], but we leave this for future work, focusing on fine-grained one-step critiquing here.

5.2 Disentanglement Performance

We now quantify the disentanglement achieved across the attributes studied, as described in Section 5. We infer from Table 5 that supervised β-VAE outperforms the baselines (Multi-VAE and Multi-DAE) and unsupervised β-VAE. It also achieves a high disentanglement and completeness score. Recall that these indicate the extent to which an attribute a is encoded only in the corresponding dimension c_a in the latent representation. Thus they measure whether adjusting c_a is sufficient to critique attribute a.

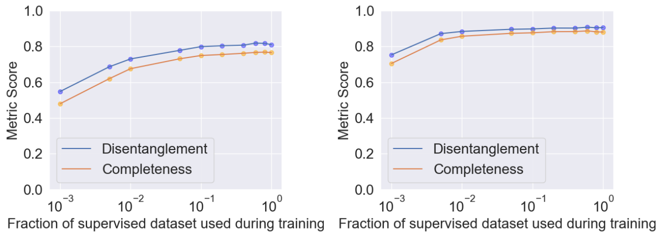
We further study the variation in disentanglement performance with the amount of labeled data used. We train our model for different fractions of the total labelled users, ranging from [0.1 to 100]%, which corresponds to [51 to 39857] labels for MovieLens-20m and [125 to 44928] labels for GoodReads-Comics. We can infer from Figure 3[1] that even with a few[2] supervised labels in the training set, we obtain Disentanglement scores of 0.73 and 0.75 for Movielens-20m and GoodReads-Comics respectively, which is close to the performance when trained with all labeled users, (0.81, 0.90 respectively). We observe a similar trend for completeness as well. This is one of the key findings: we achieve significant disentanglement for user representation, with only around 100 labels per attribute.

[1] Note that each point in the graph is an average of five runs.

[2] 0.5% (250 points) and 0.1% (125 points) of the labeled dataset for Movielens-20m, Goodreads-Comics respectively.

Model	ML-1m		ML-20m		GR-Comics		GR-Children	
	D	C	D	C	D	C	D	C
Multi-DAE [16]	0.381	0.312	0.317	0.238	0.182	0.183	0.287	0.243
Multi-VAE [16]	0.361	0.294	0.245	0.271	0.164	0.184	0.308	0.263
β-VAE	0.745	0.473	0.501	0.298	0.329	0.227	0.484	0.303
Supervised β-VAE	**0.825**	**0.656**	**0.801**	**0.719**	**0.902**	**0.857**	**0.823**	**0.728**

Table 5: Disentanglement performance of baselines, β-VAE and Supervised β-VAE across Movielens(1m, 20m)and Goodreads(Comics, Children)[Books Domain]. D stands for Disentanglement score and C for Completeness



(a) Movielens-20m

(b) GoodReads-Comics

Figure 3: The trend for Disentanglement and Completeness metrics when fewer labels are used for Movielens-20m and Goodreads-Books. It is evident that comparable scores are obtained even with very few labels on both datasets.

Algorithm 2 Control/Critiquing recommendations simulation

```
Input: Given a user $u$ represented by the items they have interacted with $\mathbf{x}_u$ and attribute $a$ that will be adjusted.
$[\mathbf{z} : \mathbf{c}]$ = encoder($\mathbf{x}_u$)
$ranking\_scores$ = []
for $g \in \{-8, ..., 8\}$ do
    $old\_c_a = c_a$
    //Only adjust the $a^{th}$ dimension of user representation's subset c
    $c_a = g \times c_a$
    $\tilde{y} \leftarrow$ Decoder($[\mathbf{z} : \mathbf{c}]$)
    //$\mathcal{I}_a$: Relevant items where attribute a is present
    $ranking\_scores$.insert($recall@k(\tilde{y}, \mathcal{I}_a)$)
    $c_a = old\_c_a$
end for
plot $ranking\_scores$ against $[-8, 8]$ for attribute $a$
```

This enables learning representations even for rare attributes. The results in Table 5 are reported with 1% of the labelled dataset.

5.2.1 *Control and critiquing of recommendations.*

One of the primary aims of our model is to provide refined control over recommended items. For instance, if a user asks for *more romantic* movies, the model should rank movies with the *romantic* attribute higher. We use algorithm 2 to study the critiquing and control abilities of the supervised β-VAE model. Specifically, we retrieve the user representations based on the items rated using the encoder then adjust the factor value c_a associated with attribute[3] a by multiplying it by a factor g (ranging from $[-8, 8]$, which we empirically found to be a range that fully demonstrates the effect. Beyond this range, the impact is consistent with the ranking performance near the endpoints -8 and 8 for both β-VAE and supervised β-VAE). We then retrieve the items recommended using the adjusted user representation in the decoder and evaluate them against the known relevant items (items where the attribute a is present). We compute recall@k with respect to these relevant items.

Attribute	User-A (Children genre)		User-B (Action genre)	
	Movie Name	*Genre*	*Movie Name*	*Genre*
None	Yellow Submarine	Adventure, Animation, Comedy	Mission: Impossible	Action, Adventure
Sad	Alice in Wonderland	Adventure, Animation, Children	Blade Runner	Action, Sci-Fi
Action	Jurassic Park	Action, Adventure, Sci-Fi	Men in Black	Action, Comedy
Suspense	Lord of the Rings	Adventure|Fantasy	Ronin	Action, Crime, Thriller
Romance	Beauty and the Beast	Animation, Children, Fantasy	Forrest Gump	Comedy, Drama
Violence	Dark Crystal, The	Adventure, Fantasy	Twelve Monkeys	Mystery,Sci-Fi
Funny	Inspector Gadget	Action, Adventure, Children	From Dusk Till Dawn.	Action, Comedy

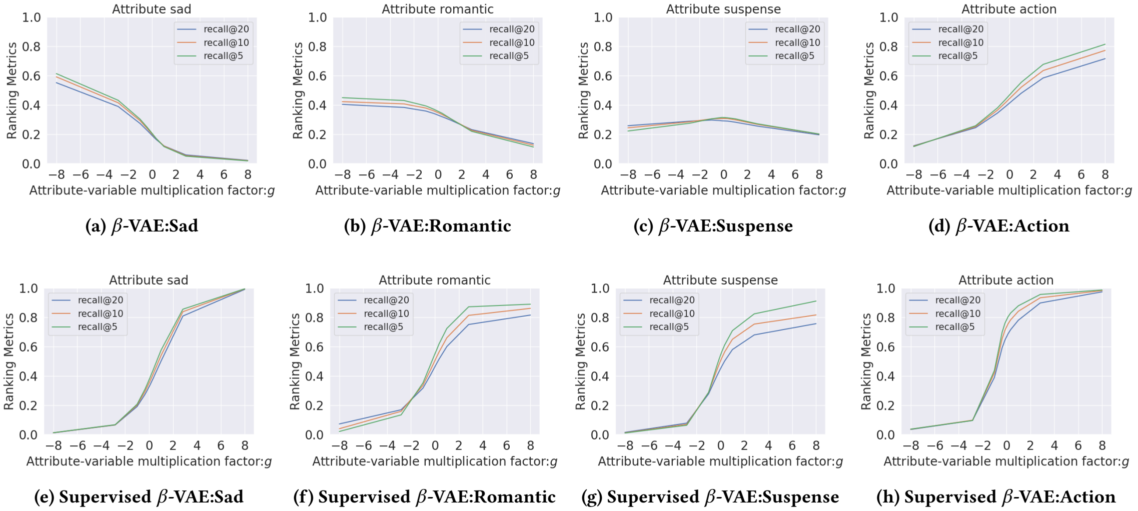
Table 6: Comparison between the top recommendation for User-(A and B), when they prefer attribute given in Col. 1

We show the impact of multiplying with the factor g, on a selection of disentangled attribute factors c_a, against the ranking metrics in Figures 4 and 5, comparing unsupervised and supervised β-VAE. From a comparative analysis between the top and bottom rows of the Figures, we can infer that supervised β-VAE provides better control over the presence (or absence) of the desired attributes in the recommendations, as the range of recall@k is substantially higher as g is adjusted. This effectively means that a user can tune her recommendations to contain between no recommendations of items with an attribute (recall@k $\approx$ 0) up to all items in her recommendation list with the corresponding attribute (recall@l $\approx$ 1). Even for attributes like *suspense* in Figure 4, and *mystery* in Figure 5, for which the normal β-VAE model does not provide much control (Fig (c)), we see significant improvement with our model (Fig (g)) in Figures 4 and 5. Note that the range of recall@k values is consistently higher across all the disentangled-dimensions than the corresponding disentangled dimensions in β-VAE for the predefined attributes. Also, we note that each point in the graphs is an average across all the users present in the test split.

5.2.2 *Case Study: User Control with Disentanglement.*

We study how adjusting profile attributes changes the recommendations for two example user profiles: We created two synthetic user profiles, where each has rated 20 movies. `User-A` rated movies mostly from the *Children* genre, such as *Toy Story, Snow White.* `User-B` rated movies with major *Action* content such as *Batman, Mortal Kombat.* We hypothesize that when `User-B` asks for more *funny* movies, the system should move towards *Action Comedy* movies rather than classic comedies. Similarly, if more *violent* movies are asked for by `User-A`, we would expect recommendations would still be less violent than those returned for `User-B`.

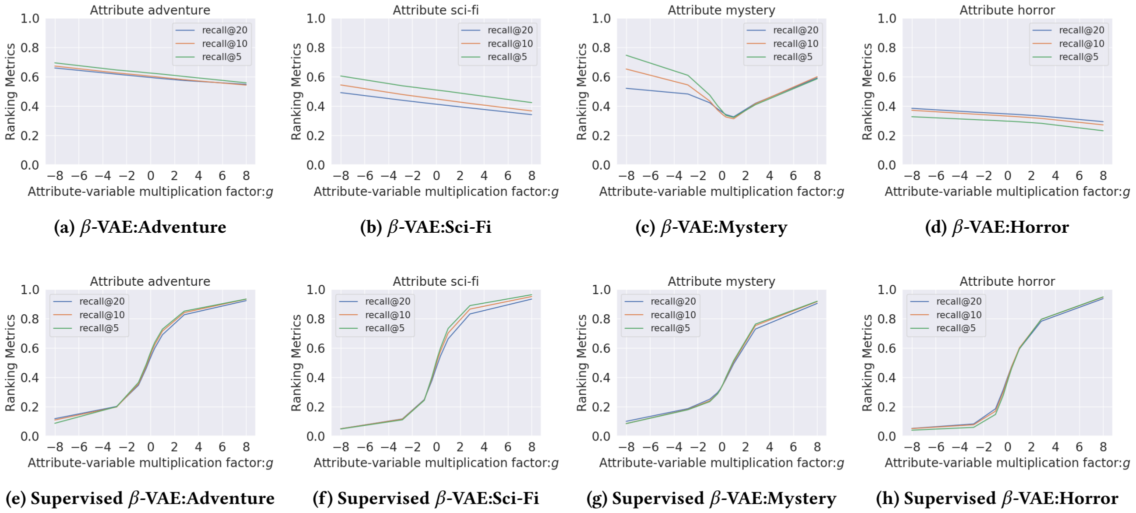
[3] For (unsupervised) β-VAE, we adjust the dimension with highest feature importance score computed using a Gradient Boosting Classifier for attribute a.



(a) β-VAE:Sad **(b) β-VAE:Romantic** **(c) β-VAE:Suspense** **(d) β-VAE:Action**

(e) Supervised β-VAE:Sad **(f) Supervised β-VAE:Romantic** **(g) Supervised β-VAE:Suspense** **(h) Supervised β-VAE:Action**

Figure 4: Control over recommendations when factor-value c_a is adjusted by multiplicative factor $g \in [-8, 8]$. Recommendation lists are evaluated by recall@(5,10,20). Relevance is determined by the presence of attribute a in the retrieved item. We compare β-VAE (top) with supervised β-VAE (bottom) for sad, romantic, suspense and action attributes on Movielens-20m.



(a) β-VAE:Adventure **(b) β-VAE:Sci-Fi** **(c) β-VAE:Mystery** **(d) β-VAE:Horror**

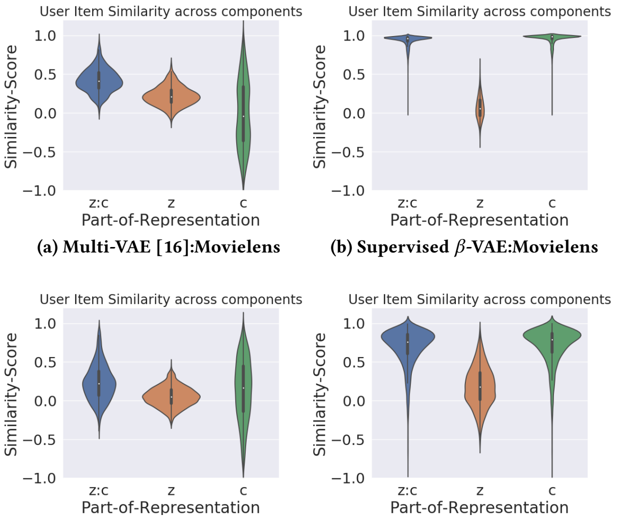
(e) Supervised β-VAE:Adventure **(f) Supervised β-VAE:Sci-Fi** **(g) Supervised β-VAE:Mystery** **(h) Supervised β-VAE:Horror**

Figure 5: We compare β-VAE (top) with supervised β-VAE (bottom) for adventure, sci-fi, mystery and horror attributes for Goodreads-Comics for the same analysis as in Figure 4.

We obtain revised recommendations as follows: (i) we obtain user representation by passing $\mathbf{x}_u$ in Figure 2 to the encoder. Note that $\mathbf{x}_u$ is set to 1 for the handpicked 20 movies. (ii) We adjust the value of the attribute in the encoded user representation to a higher value. (iii) This adjusted user model is then passed as input to the decoder, to generate revised recommendations. We show the top-1 movie recommended in Table 6. Notice that even changing intense attributes like *suspense* and *violence* have an appropriate impact on `User-A` recommendations. For `User-B`, when attributes such as *suspense* or *funny* are increased, recommendations contain a blend of *Action* and the requested attribute. The model provides control over recommendations keeping the user profile in perspective.

5.2.3 *Explainable Representations.* Here we show that with supervised disentanglement the similarity between the user and item representations is dominated by the disentangled part of the representation **c**. As we have associated predefined preferences to **c**, the items recommended could be explained with **c**.

We compute user representations using our model encoder. We obtain item representations by encoding one item at a time in the one-hot encoding in Figure 2 at the encoder. The representations



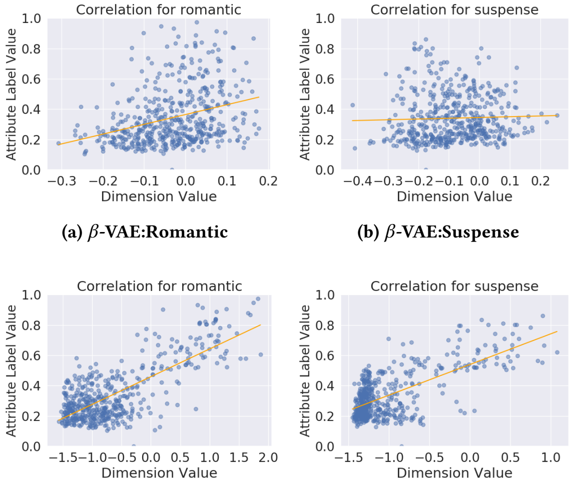
(a) Multi-VAE [16]:Movielens

(b) Supervised β-VAE:Movielens

(c) Mutil-VAE [16]:Comics

(d) Supervised β-VAE: Comics

Figure 6: Distribution of cosine similarities between user and item representations for Multi-VAE and Supervised β-VAE, for various components of the representation.



(a) β-VAE:Romantic

(b) β-VAE:Suspense

(c) Supervised β-VAE:Romantic

(d) Supervised β-VAE:Suspense

Figure 7: Correlation between factor value in item-representation and attribute label value for Movielens-20m for *romantic* and *suspense*. Our model achieves higher correlation (bottom-row) as compared to β-VAE (top-row).

consist of latent dimensions (**z**) and supervised disentangled dimensions (**c**). We compute the cosine similarity between user and items (seen by the corresponding user), using (i) the whole representation [**z** : **c**], (ii) only entangled dimensions (**z**), (iii) only disentangled dimensions (**c**). Figure 6 shows the similarity score distributions using the three settings for Multi-VAE [16] and Supervised β-VAE.

From Figure 6 (a,c), we see that for Multi-VAE [16] the similarity using the whole representation is not being dominated either by entangled or disentangled dimensions. We also observe that β-VAE has a similar trend. With both algorithms, the complete representation is necessary to capture the interaction between user and item. As discussed earlier, as none of the dimensions in Multi-VAE can be strongly associated with any of the preferences, it is difficult to explain the items recommended using only the representation. In comparison to Multi-VAE, the cosine similarity in representations generated with supervised β-VAE using dimensions [**z** : **c**] , is dominated by the disentangled dimensions **c** (Figure 6(b,d)). As we have associated predefined preferences with dimensions in **c**, we could explain the correlation between the user and an item using only **c**. This behavior is consistent across the movie and book domains. We also compare the learned dimension values c_a associated with an attribute a in item's representation with the ground truth relevance scores associated with a movie and tag (such as *suspense, thriller, etc.*)[4]. We see in Figure 7 that the value of the associated dimension with attribute *romantic, suspense* in (a,b) and the relevant tag score are much less correlated for β-VAE, whereas in (c,d), the correlation improves with our model.

Model	ML-1m		ML-20m		GR-Comics		GR-Children	
	R@50	N@100	R@50	N@100	R@50	N@100	R@50	N@100
MultiDAE[16]	0.467	0.404	0.530	0.419	0.510	0.401	0.591	0.425
MultiVAE[16]	0.458	0.405	0.519	0.411	0.550	0.439	0.577	0.404
β-VAE	0.454	0.411	0.516	0.407	0.549	0.443	0.586	0.415
Supervised β-VAE	0.445	0.415	0.515	0.403	0.538	0.430	0.581	0.413

Table 7: Recommendation performance across models on Movielens(1m, 20m) and Goodreads(Comics, Children). We omit error bars as confidence interval is in the 4th digit.

5.3 General Recommendation Performance

Finally, we compare our models β-VAE and supervised β-VAE against MultiDAE and MultiVAE [16] on the Movielens and GoodReads datasets. We see in Table 7 that the performance of our model and the baseline models on ranking-based metrics (recall@k, and NDCG@k) on the test split are comparable across all datasets. On some datasets supervised β-VAE outperforms the baselines, while on others it is slightly behind (We believe that this might be due to the constraints on the representation imposed by the increased β parameter). However, we argue that these small drops in performance are compensated for by the significant additional flexibility and interpretability provided by our model.

6 CONCLUSION

The supervised β-VAE recommendation model allows for disentangled representation of user preferences and hence allows for critiquing of the recommendations provided. The analysis shows that the model can effectively learn the attribute representations using only a tiny fraction of labeling information and map them on individual attributes of the user representations. Recommendations can then be generated, explained and controlled over these attributes without the use of any label or attribute information. Finally the overall recommendation accuracy is on par with state-of-the art collaborative filtering methods.

[4] The (movie,tag) association is present in Movielens-20m [6]

REFERENCES

[1] Christopher P. Burgess, Irina Higgins, Arka Pal, Loïc Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. 2018. Understanding disentangling in β-VAE. *ArXiv* abs/1804.03599 (2018).

[2] Li Chen and Pearl Pu. 2012. Critiquing-based recommenders: survey and emerging trends. *User Modeling and User-Adapted Interaction* 22, 1 (2012), 125–150. https://doi.org/10.1007/s11257-011-9108-6

[3] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In *NIPS*. 2172–2180.

[4] Sunny Duan, Loic Matthey, Andre Saraiva, Nick Watters, Chris Burgess, Alexander Lerchner, and Irina Higgins. 2020. Unsupervised Model Selection for Variational Disentangled Representation Learning. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SyxL2TNtvr

[5] Cian Eastwood and Christopher KI Williams. 2018. A framework for the quantitative evaluation of disentangled representations. (2018).

[6] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm Transactions on Interactive Intelligent Systems (TIIS)* 5, 4 (2015), 1–19.

[7] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent Neural Networks with Top-k Gains for Session-Based Recommendations. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*. Association for Computing Machinery, New York, NY, USA, 843–852. https://doi.org/10.1145/3269206.3271761

[8] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *International Conference on Learning Representations (ICLR '16)*.

[9] Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *ICLR*. OpenReview.net.

[10] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward Controlled Generation of Text. In *ICML (Proceedings of Machine Learning Research)*, Vol. 70. PMLR, 1587–1596.

[11] Hyunjik Kim and Andriy Mnih. 2018. Disentangling by Factorising. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Jennifer Dy and Andreas Krause (Eds.), Vol. 80. PMLR, Stockholmsmässan, Stockholm Sweden, 2649–2658. http://proceedings.mlr.press/v80/kim18b.html

[12] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes.. In *ICLR*, Yoshua Bengio and Yann LeCun (Eds.). http://dblp.uni-trier.de/db/conf/iclr/iclr2014.html#KingmaW13

[13] Yehuda Koren. 2008. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*. Association for Computing Machinery, New York, NY, USA, 426–434. https://doi.org/10.1145/1401890.1401944

[14] Y. Koren, R. Bell, and C. Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (Aug 2009), 30–37. https://doi.org/10.1109/MC.2009.263

[15] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic DENOYER, and Marc' Aurelio Ranzato. 2017. Fader Networks:Manipulating Images by Sliding Attributes. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 5967–5976.

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

[17] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem. 2019. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations. In *Proceedings of the 36th International Conference on Machine Learning (ICML) (Proceedings of Machine Learning Research)*, Vol. 97. PMLR, 4114–4124.

[18] Francesco Locatello, Michael Tschannen, Stefan Bauer, Gunnar Rätsch, Bernhard Schölkopf, and Olivier Bachem. 2019. Disentangling Factors of Variation Using Few Labels. arXiv:cs.LG/1905.01258

[19] Kai Luo, Scott Sanner, Ga Wu, Hanze Li, and Hojin Yang. 2020. Latent Linear Critiquing for Conversational Recommender Systems. In *Proceedings of The Web Conference 2020 (WWW '20)*. Association for Computing Machinery, New York, NY, USA, 2535–2541. https://doi.org/10.1145/3366423.3380003

[20] Kai Luo, Hojin Yang, Ga Wu, and Scott Sanner. 2020. Deep Critiquing for VAE-Based Recommender Systems. Association for Computing Machinery, New York, NY, USA, 1269–1278. https://doi.org/10.1145/3397271.3401091

[21] Chen Ma, Peng Kang, and Xue Liu. 2019. Hierarchical gating networks for sequential recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 825–833.

[22] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning Disentangled Representations for Recommendation. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 5711–5722. http://papers.nips.cc/paper/8808-learning-disentangled-representations-for-recommendation.pdf

[23] James Reilly, Kevin McCarthy, Lorraine McGinty, and Barry Smyth. 2005. Incremental Critiquing. In *Research and Development in Intelligent Systems XXI*, Max Bramer, Frans Coenen, and Tony Allen (Eds.). Springer London, London, 101–114.

[24] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *Proceedings of the 31st International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Eric P. Xing and Tony Jebara (Eds.), Vol. 32. PMLR, Bejing, China, 1278–1286. http://proceedings.mlr.press/v32/rezende14.html

[25] P. K. Rubenstein, B. Schölkopf, and I. Tolstikhin. 2018. Learning Disentangled Representations with Wasserstein Auto-Encoders. In *Workshop at the 6th International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=Hy79-UJPM

[26] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. AutoRec: Autoencoders Meet Collaborative Filtering. In *Proceedings of the 24th International Conference on World Wide Web (WWW '15 Companion)*. Association for Computing Machinery, New York, NY, USA, 111–112. https://doi.org/10.1145/2740908.2742726

[27] Elena Smirnova and Flavian Vasile. 2017. Contextual Sequence Modeling for Recommendation with Recurrent Neural Networks. In *Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems (DLRS 2017)*. Association for Computing Machinery, New York, NY, USA, 2–9. https://doi.org/10.1145/3125486.3125488

[28] Harald Steck. 2015. Gaussian Ranking by Matrix Factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys '15)*. Association for Computing Machinery, New York, NY, USA, 115–122. https://doi.org/10.1145/2792838.2800185

[29] Sjoerd Van Steenkiste, Francesco Locatello, Jürgen Schmidhuber, and Olivier Bachem. 2019. Are Disentangled Representations Helpful for Abstract Visual Reasoning? In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 14222–14235.

[30] Mengting Wan and Julian J. McAuley. 2018. Item recommendation on monotonic behavior chains. In *RecSys*. ACM, 86–94.

[31] Ga Wu, Kai Luo, Scott Sanner, and Harold Soh. 2019. Deep Language-Based Critiquing for Recommender Systems. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys '19)*. Association for Computing Machinery, New York, NY, USA, 137–145. https://doi.org/10.1145/3298689.3347009

[32] Yao Wu, Christopher DuBois, Alice X. Zheng, and Martin Ester. 2016. Collaborative Denoising Auto-Encoders for Top-N Recommender Systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining (WSDM '16)*. Association for Computing Machinery, New York, NY, USA, 153–162. https://doi.org/10.1145/2835776.2835837