Learning Disentangled Representations for Recommendation (NeurIPS 2019)

Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, Wenwu Zhu

https://arxiv.org/abs/1910.14238

Basia Rychalska

27.04.2020

Why did I choose this topic?

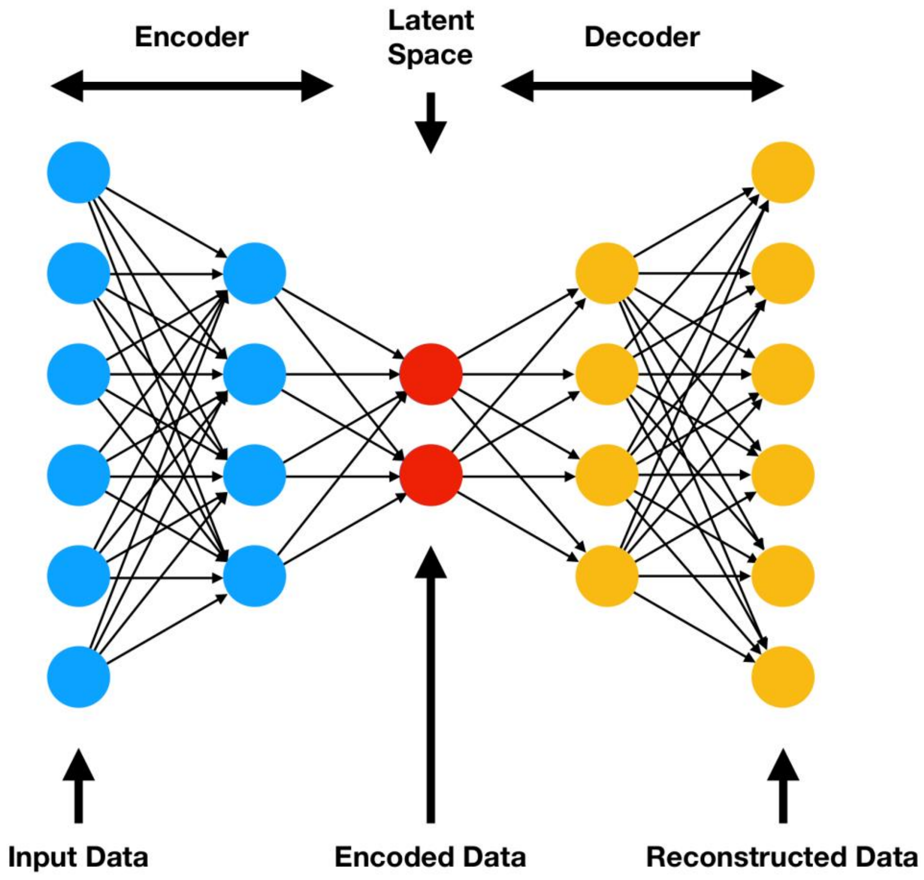
- It is an important paper (SOTA results, part of a trend)
- It is a surprising application of a deep learning algorithm
- It is interpretable

Autoencoder

- Goal: minimize difference between Input and Reconstructed Data (Output)

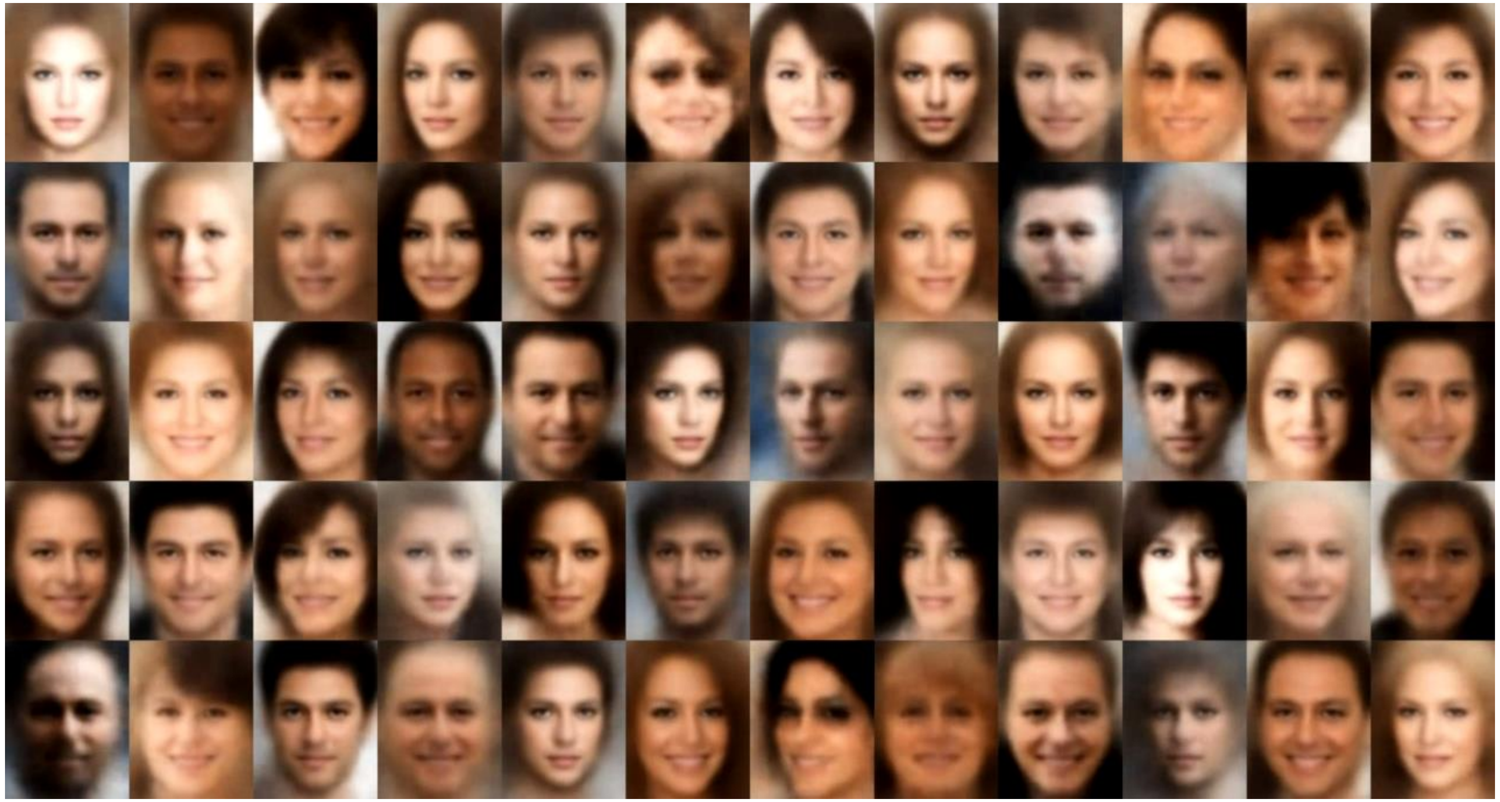
$G = min|x - x'|$

- Main usage is obtaining an encoded smaller representation of input



Variational Autoencoder

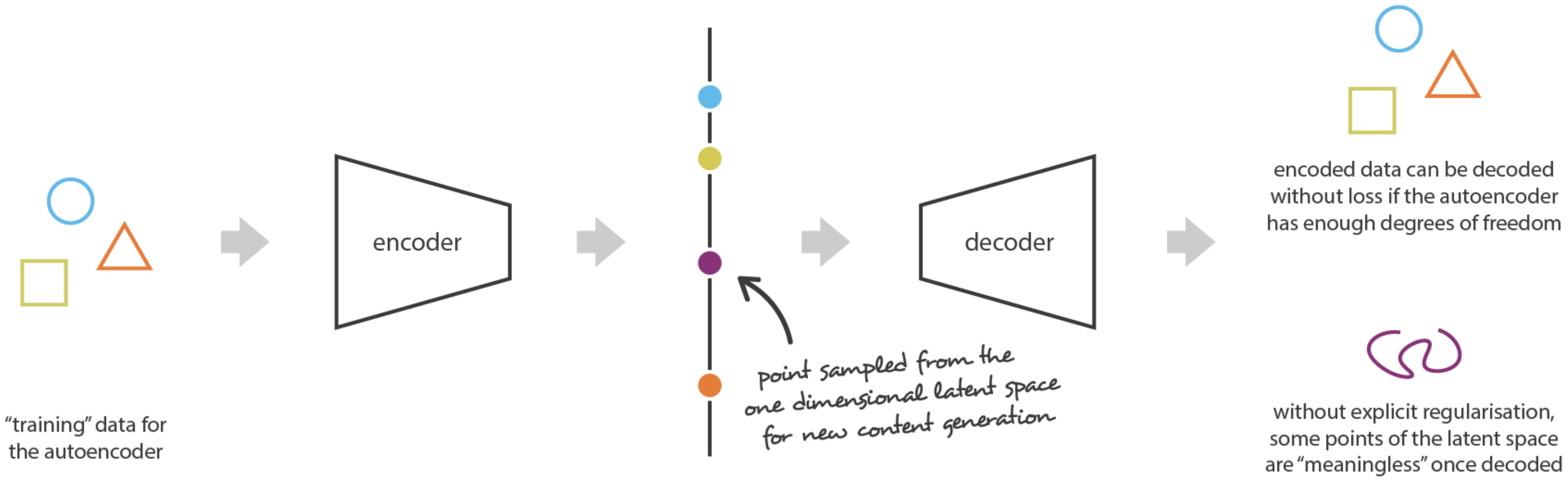
- An autoencoder which can generate new meaningful content



https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

Why not a regular Autoencoder?

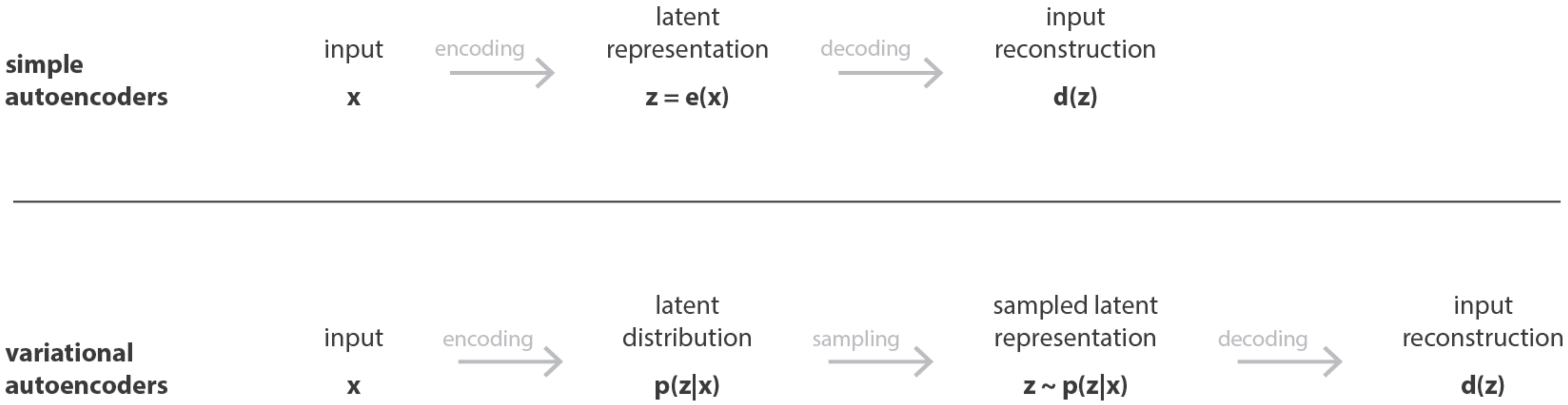
Autoencoders overfit too much to be able to generate new meaningful content.



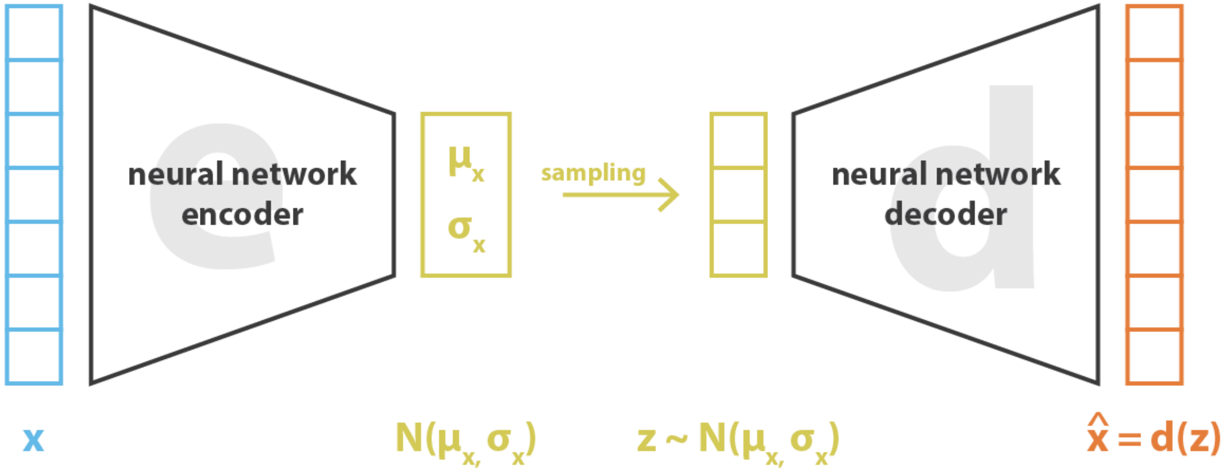
https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

Variational Autoencoder

- It is a marriage between a Graphical Model and a Neural Network
- It is an autoencoder which is **regularised to avoid overfitting and ensure that the latent space has good properties that enable generative process.**

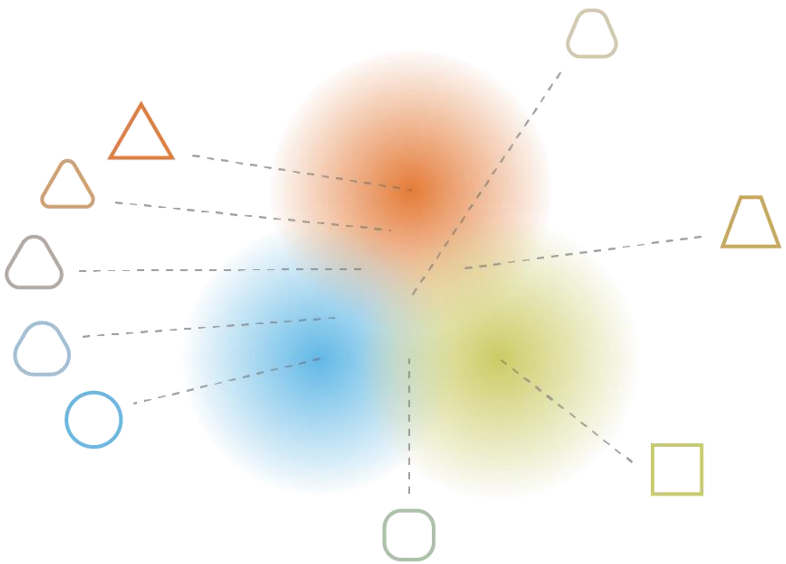


Variational Autoencoder



$$\text{loss} = \| x - \hat{x} \|^2 + \text{KL}[\, N(\mu_x, \sigma_x), N(0, I) \,] = \| x - d(z) \|^2 + \text{KL}[\, N(\mu_x, \sigma_x), N(0, I) \,]$$

Desired effect of normalization:



MacridVAE – NeurIPS 2019

Users \ *Items*	*1*	*2*	...	*i*	...	*m*
1	5	3		1	2	
2		2				4
:			5			
u	3	4		2	1	
:					4	
n			3	2		

- Many latent factors behind the users' decision making processes
- The factors can be multilevel:
  - Macro: I want to buy a bag/clothes/toys/electronics ....
  - Micro: I want to buy a small/green/leather etc. bag

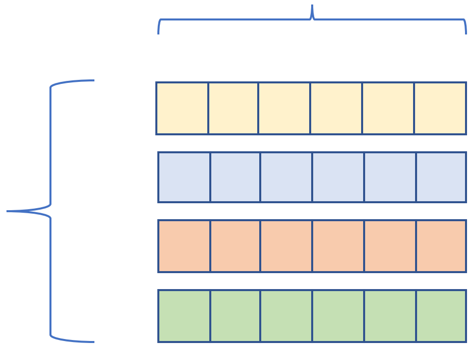
MacridVAE

- Goal is to learn how to represent a user representation **z** and vector **C**

z: For each user

K macro level concepts (clothes, electronics, ...)

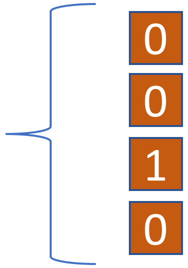
d micro level concepts (color, shape, ...)



- They choose a constant K to define the number of macro (high level) concepts and d for the number of micro concepts
- User interests consist of K vectors of dimensionality d

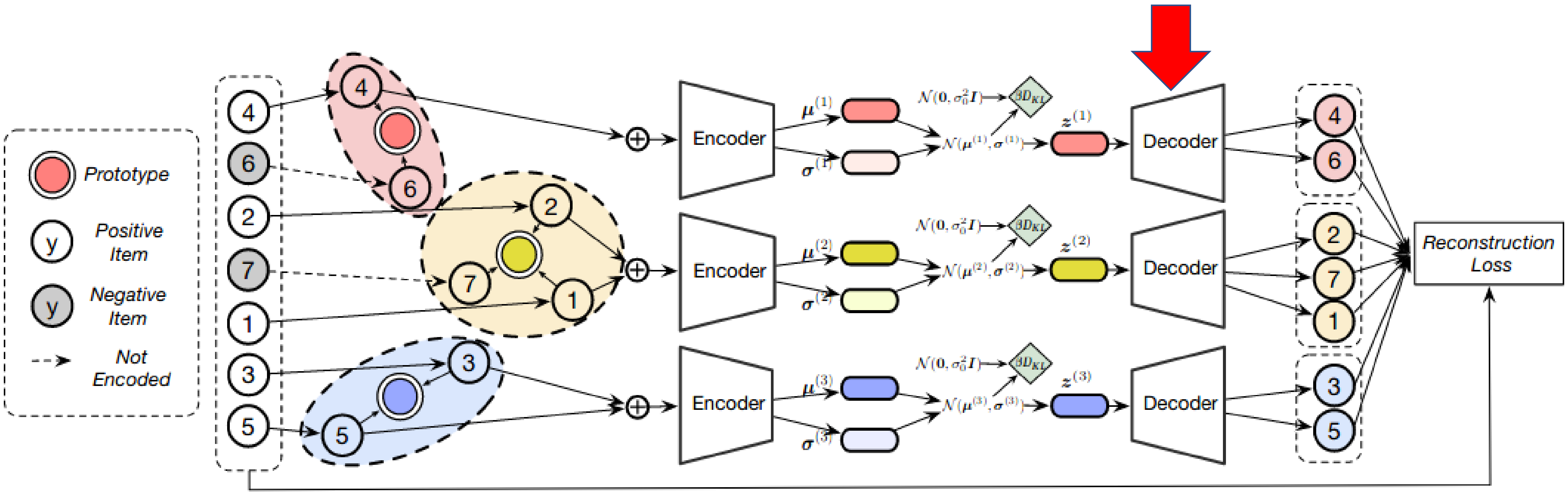
C: For each item

K macro level concepts (clothes, electronics, ...)



MacridVAE

Note that here the Decoder does not know what items the user did not like (the only input it has is the **z** vector)



- They exploit a regularization term based on KL divergence which encourages independence between learned vectors ('disentanglement')

MacridVAE – why is there a VAE at all?

- Obtaining a high quality squashed representation is generally the solution to recommender problems
- Authors of an older recommender VAE paper (MultVAE https://arxiv.org/pdf/1802.05814v1.pdf) observed a long time ago that regularization of the latent representation is key (http://www.cs.toronto.edu/~lcharlin/papers/cofactorization.pdf)

The distinguishing feature of Co-Factor is that the regularization comes from a deterministic non-linear transformation of the original user-item preference data (…). This kind of regularization (either through incorporation of side information or re-encoding the input as in CoFactor) can alternatively be viewed as enforcing more complex structure in the prior of the latent factors. For example, Ranganathet al.[17] find that imposing a deep exponential family prioron the matrix factorization model, which implicitly conditions on consumption counts in a non-linear fashion, can be helpful in reducing the effect of extremely popular (or extremely unpopular) items on held-out recommendation tasks. This is analogous to our findings with the CoFactor model.

Previous work [27] has demonstrated that adding carefully crafted non-linear features into the linear latent factor models can significantly boost recommendation performance.

MacridVAE – Interpretability

(a) Bag size. (b) Bag color. (c) Styles of phone cases.

(d) Bag size. The same dimension as Figure 3a. (e) Bag color. The same dimension as Figure 3b. (f) Chicken → beef → mutton → seafood.

Figure 3: Starting from an item representation, we gradually alter the value of a target dimension, and list the items that have representations similar to the altered representations (see Subsection 2.4).

MacridVAE – Results

Table 1: Collaborative filtering. All methods are constrained to have around $2Md$ parameters, where M is the number of items and d is the dimension of each item representation. We set $d = 100$.

Dataset	Method	Metrics		
		NDCG@100	Recall@20	Recall@50
AliShop-7C	MultDAE	0.23923 (±0.00380)	0.15242 (±0.00305)	0.24892 (±0.00391)
	β-MultVAE	0.23875 (±0.00379)	0.15040 (±0.00302)	0.24589 (±0.00387)
	Ours	**0.29148** (±0.00380)	**0.18616** (±0.00317)	**0.30256** (±0.00397)
ML-100k	MultDAE	0.24487 (±0.02738)	0.23794 (±0.03605)	0.32279 (±0.04070)
	β-MultVAE	0.27484 (±0.02883)	0.24838 (±0.03294)	0.35270 (±0.03927)
	Ours	**0.28895** (±0.02739)	**0.30951** (±0.03808)	**0.41309** (±0.04503)
ML-1M	MultDAE	0.40453 (±0.00799)	0.34382 (±0.00961)	0.46781 (±0.01032)
	β-MultVAE	0.40555 (±0.00809)	0.33960 (±0.00919)	0.45825 (±0.01039)
	Ours	**0.42740** (±0.00789)	**0.36046** (±0.00947)	**0.49039** (±0.01029)
ML-20M	MultDAE	0.41900 (±0.00209)	0.39169 (±0.00271)	**0.53054** (±0.00285)
	β-MultVAE	0.41113 (±0.00212)	0.38263 (±0.00273)	0.51975 (±0.00289)
	Ours	**0.42496** (±0.00212)	**0.39649** (±0.00271)	0.52901 (±0.00284)
Netflix	MultDAE	0.37450 (±0.00095)	0.33982 (±0.00123)	0.43247 (±0.00126)
	β-MultVAE	0.36291 (±0.00094)	0.32792 (±0.00122)	0.41960 (±0.00125)
	Ours	**0.37987** (±0.00096)	**0.34587** (±0.00124)	**0.43478** (±0.00125)