Variational Autoencoders for Sparse and Overdispersed Discrete Data

He Zhao[1], Piyush Rai[2], Lan Du[1], Wray Buntine[1], Dinh Phung[1], Mingyuan Zhou[3]

[1]Monash University, Australia

[2]IIT Kanpur, India

[3]The University of Texas at Austin, USA

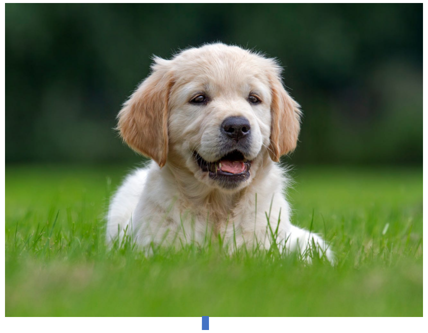
Image from https://www.aistats.org

VAEs for Sparse and Overdispersed Discrete Data

VAEs for Sparse and Overdispersed Discrete Data

Background and motivations: Variational autoencoders

VAEs for images



Encoder

$$z \sim \mathcal{N}(\mu, \quad \sigma^2)$$

Decoder

Image from:
https://www.insider.com/most-popular-dog-breeds-2019-google-search

Large-scale sparse discrete data

Count-valued data

$$[0, \cdots, 0, 0, 10, 0, 3, \cdots, 0, 5, \cdots]$$

E.g., word occurrences for a document

Binary data

$$[0, \cdots, 1, 1, 0, 1, \cdots, 0]$$

E.g., a user's buying history

VAEs for discrete data [1,2,3]

$\boldsymbol{y}$

Encoder

$$z \sim \mathcal{N}(\mu, \quad \sigma^2)$$

Decoder

softmax

$$\boldsymbol{y} \sim \text{Multinomial}(\boldsymbol{\theta})$$

[1] Neural variational inference for text processing, ICML 2016
[2] Variational autoencoders for collaborative filtering, WWW 2018
[3] On the chal- lenges of learning with inference networks on sparse, AISTATS 2018

Background and motivations: Issues with the multinomial likelihood for VAEs

Insufficient capability of modelling overdispersion in count-valued data

Overdispersion: the sample variance exceeds the sample mean in the data distribution

Example: Word burstness in documents

dog cat puppy

$$[0, \cdots, 0, 0, 10, 0, 3, \cdots, 0, 5, \cdots]$$

Self-excitation Cross-excitation

- A few bursty words occur multiple times while other words only show up once or never
  - high variance in the word counts of the document
- Multinomial models usually have insufficient capability of modelling self- and cross-excitations

Model misspecification for binary data

✅

$$[0, \cdots, 0, 0, 10, 0, 3, \cdots, 0, 5, \cdots]$$

❌

$$[0, \cdots, 1, 1, 0, 1, \cdots, 0]$$

Both issues can be tackled by replacing the multinomial likelihood with the negative binomial likelihood

Background and motivations: Multinomial V.S. Negative-Binomial

Multinomial distribution

$$\boldsymbol{y} \sim \text{Multinomial}(\boldsymbol{\theta})$$

$$p(\boldsymbol{y}) = \frac{\Gamma(\sum_v y_v + 1)}{\prod_v \Gamma(y_v + 1)} \prod_v \theta_v^{y_v}$$

- $\boldsymbol{\theta}$ is normalised
- Multinomial is a joint multivariate distribution
  - The dimensions of it are tied up

(Multivariate) Negative-Binomial distribution

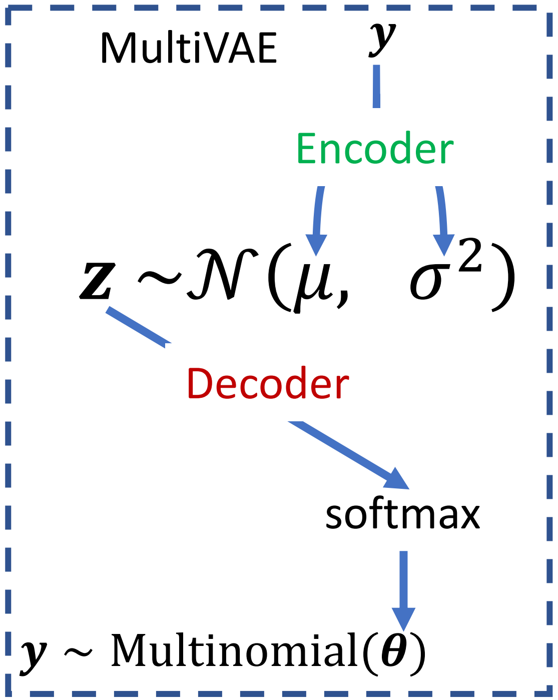
$$\boldsymbol{y} \sim \text{NegativeBinomial}(\boldsymbol{r}, \boldsymbol{p})$$

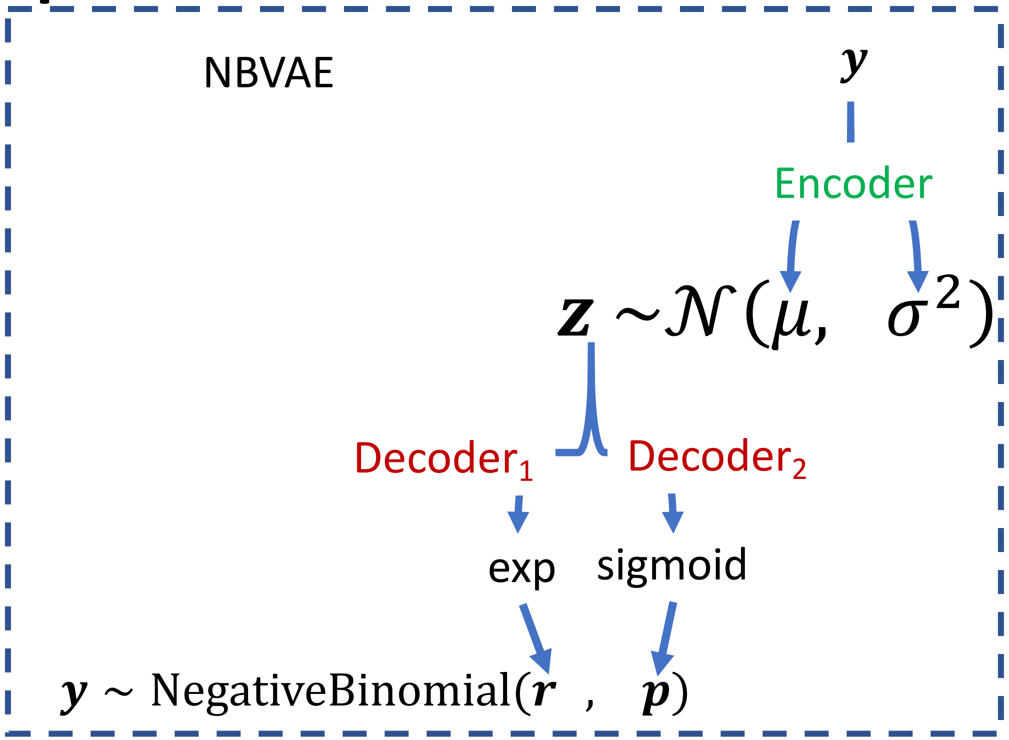
$$p(\boldsymbol{y}) = \prod_v \frac{\Gamma(r_v + y_v)}{y_v!\,\Gamma(r_v)} p_v^{y_v} (1 - p_v)^{r_v}$$

- $\boldsymbol{r}$ is positive but unnormalised
- $p_v \in (0,1)$
- Each dimension of (Multivariate) Negative-Binomial is independent

VAEs for Sparse and Overdispersed Discrete Data

Proposed approach that models overdispersion for count-valued data





Predictive probability of a word in a document, given $\boldsymbol{z}$
e.g., the probability of a word being "dog" given the content of document

$$\text{softmax}\left(\boxed{\text{decoder}}(\boldsymbol{z})\right)_{\text{dog}}$$

Both self- and cross-excitations

$$\frac{\left(\boxed{y_{\text{dog}}}+e^{\boxed{\text{decoder}_1(z)}_{\text{dog}}}\right)}{1+e^{\boxed{-\text{decoder}_2(z)}_{\text{dog}}}}$$

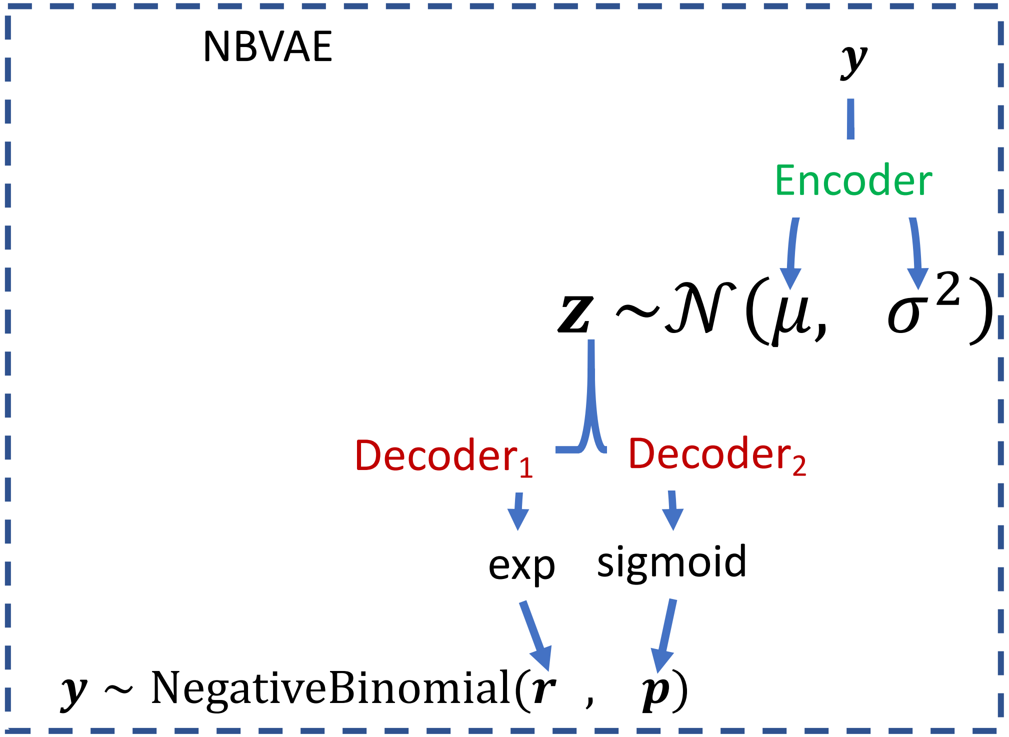
Self-excitation

Cross-excitation

Better capacity of handling self- and cross excitations

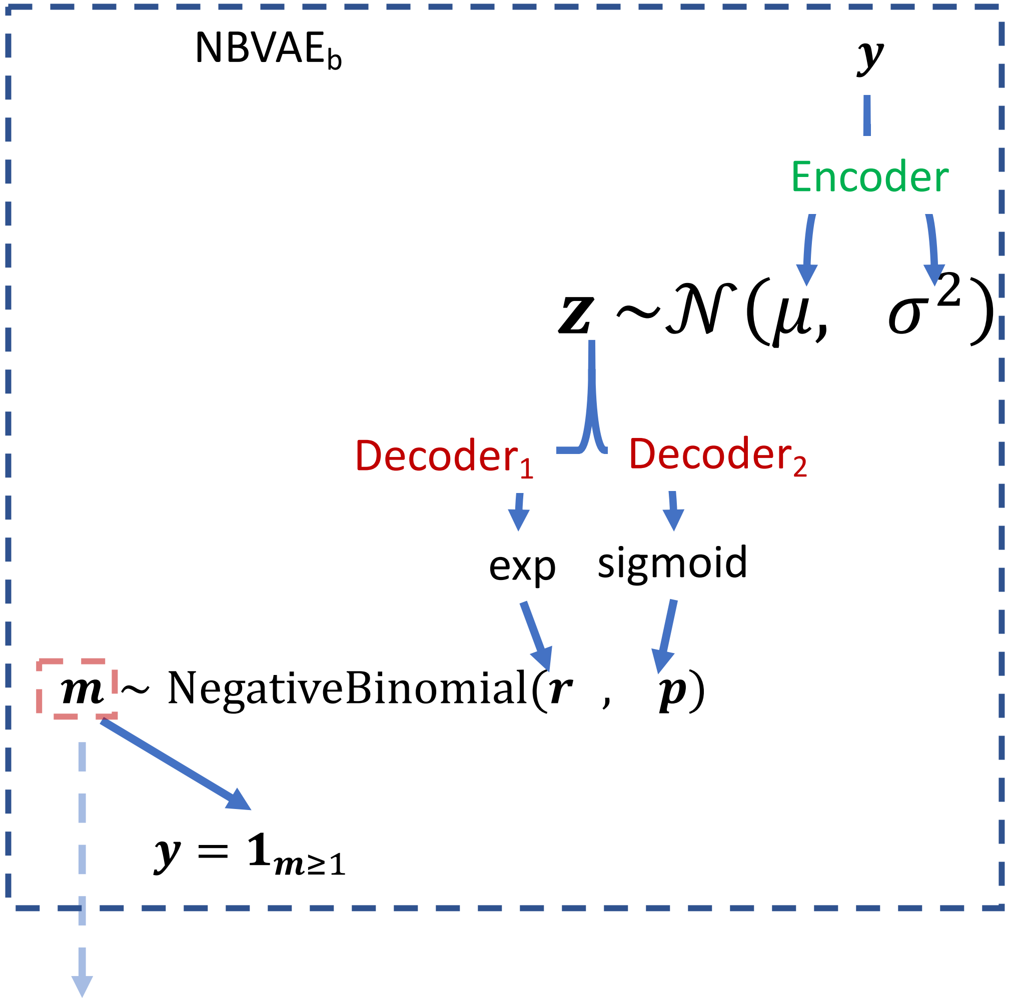
Better capacity of handling overdispersion

NBVAE for binary data



$$[0, \cdots, 1, 1, 0, 1, \cdots, 0]$$

Example: A user's buying history of items



- m_v can be interpreted as the latent interest of the user on item v
- The user will buy this item if and only if $m_v > 0$

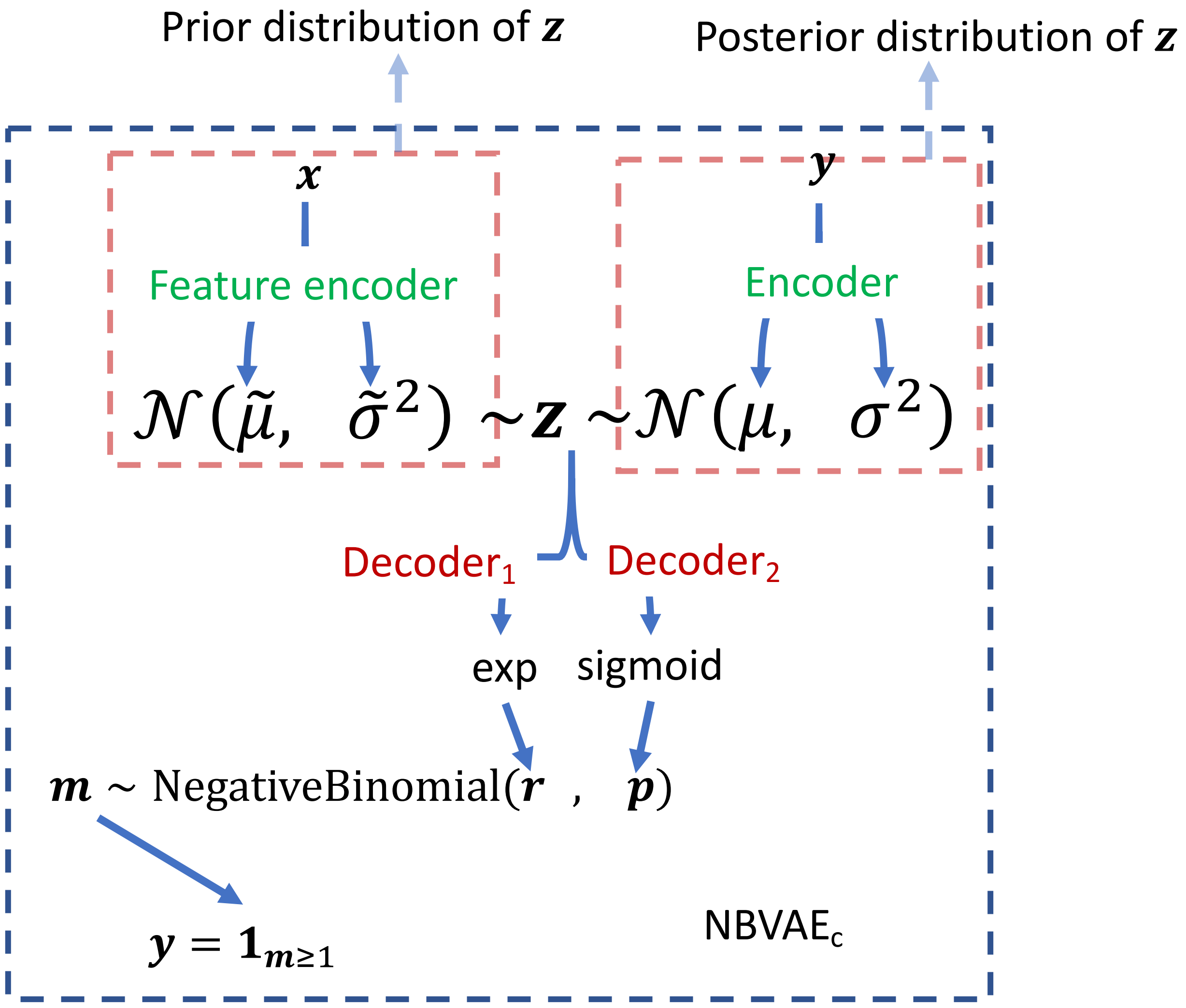
NBVAE in supervised cases: Multilabel learning as an example

Feature vector of a data sample

$$\boldsymbol{x} = [0.02, \cdots, -0.58, 0.75, 0.04, 0.11, \cdots, -0.89]$$

Label vector of the data sample

$$\boldsymbol{y} = [0, \cdots, 1, 1, 0, 1, \cdots, 0]$$



VAEs for Sparse and Overdispersed Discrete Data

Experiments on count-valued data: Text analysis as an example

- Task: predict the heldout words of a test document
- Metric: Perplexity
  - Similar to inverse model likelihood
  - Lower is better

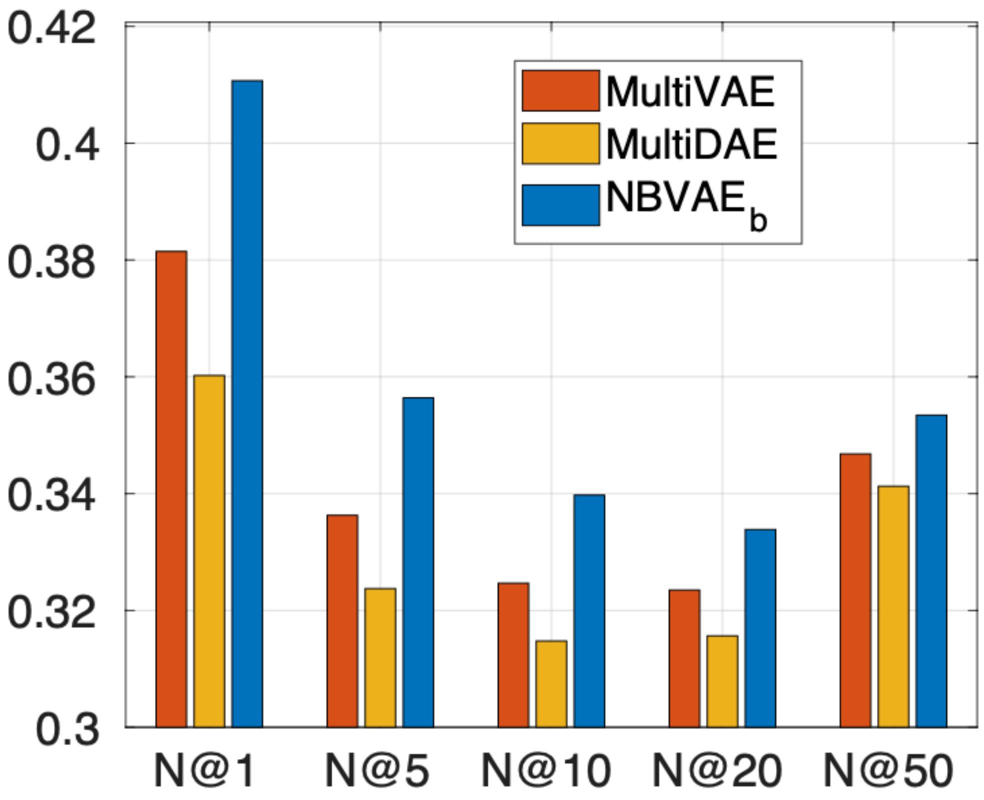
Model	Inference	Layers	20NG	RCV	Wiki
DLDA	TLASGR	128-64-32	757	815	786
DLDA	Gibbs	128-64-32	752	802	-
DPFM	SVI	128-64	818	961	791
DPFM	MCMC	128-64	780	908	783
DPFA-SBN	Gibbs	128-64-32	827	-	-
DPFA-SBN	SGNHT	128-64-32	846	1143	876
DPFA-RBM	SGNHT	128-64-32	896	920	942
NBFA	Gibbs	128	690	702	-
MultiVAE	VAE	128-64	746	632	629
MultiVAE	VAE	128	772	786	756
NBVAE	VAE	128-64	**688**	**579**	**464**
NBVAE	VAE	128	714	694	529

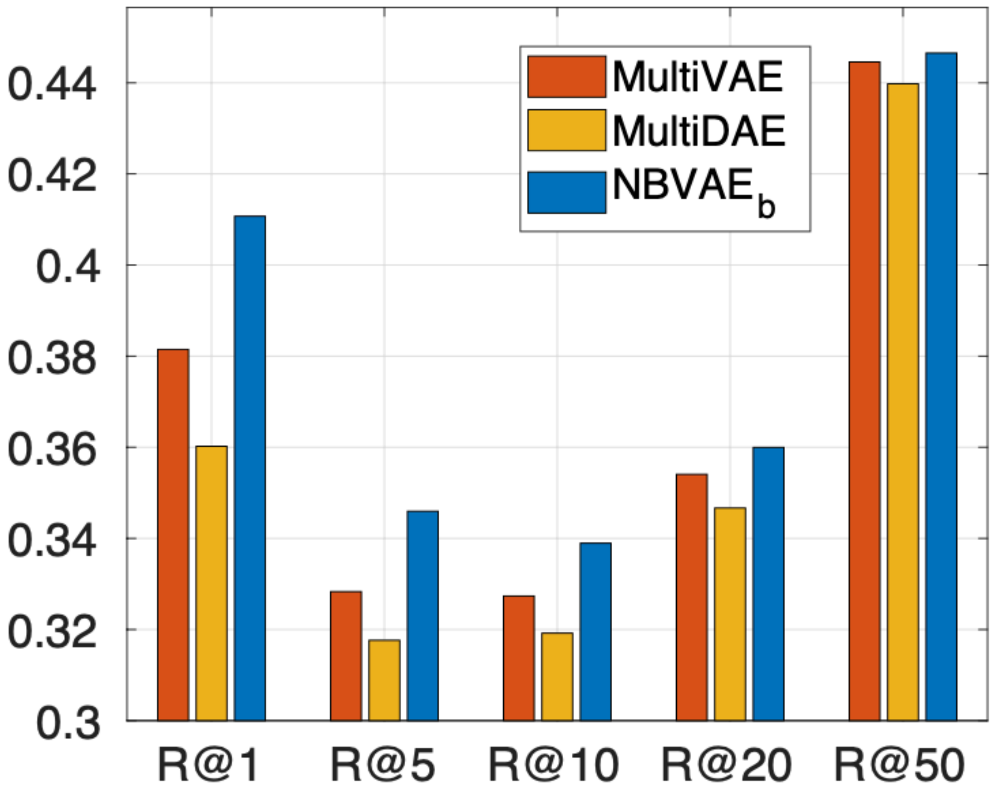
Wiki → 10 million documents

Experiments on binary data: Collaborative filtering as an example

- Task: recommend items to users using their clicking history
- Metric: Recall@R and the truncated normalized discounted cumulative gain (NDCG@R)
  - Widely-used in information retrieval and collaborative filtering
  - Higher is better

Netflix data with 0.3 million users and 40 thousand movies





Experiments on discrete data with supervisions: Multilabel learning as an example

- Task: predict a data sample’s labels given its features
- Metric: Precision@R
  - Widely-used in multilabel learning
  - Higher is better

Num of unique labels	Datasets	Metric	LEML	PfastreXML	PD-Sparse	GenEML	$NBVAE_c$
983	Delicious	P@1	65.67	67.13	51.82	-	**68.49**±0.39
		P@3	60.55	**63.48**	46.00	-	62.83±0.47
		P@5	56.08	**60.74**	42.02	-	58.04±0.31
101	Mediamill	P@1	84.01	83.98	81.86	87.15	**88.27**±0.24
		P@3	67.20	67.37	62.52	69.9	**71.47**±0.18
		P@5	52.80	53.02	45.11	55.21	**56.76**±0.26
3993	EURLex	P@1	63.40	75.45	76.43	77.75	**78.28**±0.49
		P@3	50.35	62.70	60.37	63.98	**66.09**±0.17
		P@5	41.28	52.51	49.72	53.24	**55.47**±0.15

Scalability

Table 2: Statistics of the datasets in collaborative filtering. N_{train}: number of training instances, N_{test}: number of test instances. The number of nonzeros and density are computed of each whole dataset.

Dataset	N_{train}	N_{test}	V	#Nonzeros	Density
ML-10M	49,167	10,000	10,066	4,131,372	0.0059
ML-20M	116,677	10,000	20,108	9,128,733	0.0033
Netflix	383,435	40,000	17,769	50,980,816	0.0062
MSD	459,330	50,000	36,716	29,138,887	0.0014

Table 5: Running time (seconds) per iteration on the collaborative filtering datasets.

Model	ML-10M	ML-20M	Netflix	MSD
MultiVAE	2.91	14.79	46.90	105.33
NBVAE	3.47	17.54	52.12	124.43

(c) NDCG@20 on ML-10M over iterations.

(d) NDCG@20 on ML-10M over seconds.

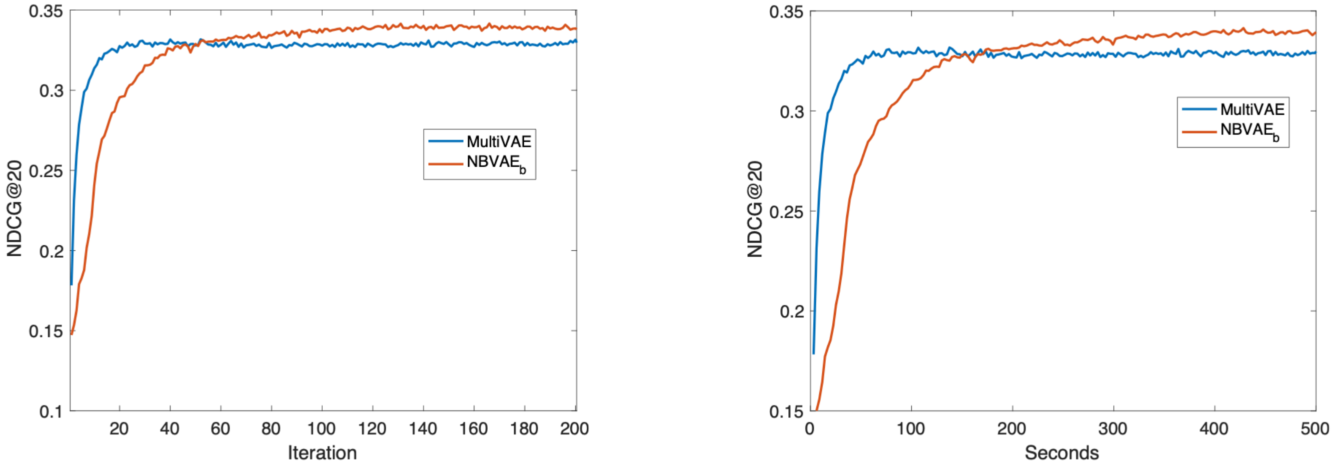


Figure 1: Performance of NBVAE and MultiVAE on the validation set during training.


Much more experiments in the paper
(a) Perplexity on 20NG over iterations.
(b) Perplexity on 20NG over seconds.
Figure 1: Comparisons
Performance of NBVAE and MultiVAE on the validation set during trainin
MultiVAE
NBVAE
MultiDAE
Perplexity
Iteration
Seconds
NDCG@20
Table 5: Running
collaborative filtering
Model
ML-10M
ML-20M
Netflix

VAEs for Sparse and Overdispersed Discrete Data

Conclusion

- Simple approach to boost the performance of VAEs on discrete data
- Can be used in many applications: text analysis, collaborative filtering, multi-label learning, …

https://github.com/ethanhezhao/NBVAE

Thanks for watching!