October 2019 1–8

Project report: "Learning latent subspaces in variational autoencoders"

Artem Shafarostov
Marina Pominova
Alexander Lyzhov
Elizaveta Lazareva

Github link: https://github.com/nikkou/latent-subspaces

1. Introduction

We consider a generative modeling problem in which objects have different labels (e.g. classes). Two possible problem objectives are 1) decreasing reconstruction error, or improving sampling quality and 2) recovering rich features correlated with each label for more controllable and interpretable generation.

Conditional subspace VAE (CSVAE) [4] aims to solve the second problem with solution grounded in the variational autoencoder - a Bayesian model which can generate the data from latent representations. The paper introduces new latent space structure and optimization scheme to solve the problem stated above.

Our general goal is to reimplement the experimental part of the paper and make the code publicly available. We also reimplement competing approaches and compare against them. We use image datasets as an easy and standard benchmark.

2. Method

The traditional variational auto-encoder (VAE) [2] is trained to maximize a lower bound on the marginal log-likelihood $\log p_\theta(x)$ over the data by utilizing a learned approximate posterior q_ϕ:

$$\log p_\theta(x|z) \geq \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p(z)) \tag{1}$$

This model maps the input to the vector of the lower dimensional latent space.

Conditional VAE [3] introduces a method of structuring the latent space. By encoding the data and modifying the variable y before decoding it is possible to manipulate the data in a controlled way. The objective for Conditional VAE is a lower bound of marginal log-likelihood:

$$\log p_\theta(x|z) \geq \mathbb{E}_{q_\phi(z|x,y)}[\log p_\theta(x|z,y)] - D_{KL}(q_\phi(z|x,y)||p(z)) \tag{2}$$

Our project revolves around replication of Conditional Subspace VAE (CondVAE) architecture and replication of the results of its benchmark. Conditional subspace VAE (CSVAE)

[4] is a model that was introduced in "Learning Latent Subspaces in Variational Autoencoders" (NIPS'18) and is based heavily on VAE [2]. The main idea of CSVAE is to restrict the latent space to only allow a single part of it to correlate with an external feature associated with an object as opposed to the restriction on a single feature in latent space as was done with CondVAE-info architecture [1]. Authors claim that this could give the model a boost in structuring the latent space and give it the ability to automatically find richer features correlated with labels.

Technically, this is done by minimizing mutual information (MI) between a label associated with an object and everything else in latent space besides the required subspace. Let's denote object and its (single, for simplicity) label (real or binary) as x and y. Let's also denote part of the latent space that we don't want to contain information about y as z, and all othes parts of the latent space as w. For this, we need proxy models: posterior on another part of the latent space (z) and a separately optimized decoder which restores the label (y) from a part of the latent space (w). The latter is needed to maximize the conditional entropy $H(Y|Z)$ taking proxy models into account to, in turn, minimize, the mutual information.

Using this notation, we define the joint probabilistic modeel:

$$\log p_{\theta,\gamma}(\mathbf{x},\mathbf{y},\mathbf{w},\mathbf{z}) = \log p_\theta(\mathbf{x}|\mathbf{w},\mathbf{z}) + \log p(\mathbf{z}) + \log p_\gamma(\mathbf{w}|\mathbf{y}) + \log p(\mathbf{y})$$

We also consider additional information minimization terms. Two resulting loss functions can be described by following equations:

$$\begin{aligned}
&-\beta_1 \mathbb{E}_{q_\phi(\mathbf{z},\mathbf{w}|\mathbf{x},\mathbf{y})}\left[\log p_\theta(\mathbf{x} \mid \mathbf{w},\mathbf{z})\right] + \beta_2 D_{KL}\left(q_\phi(\mathbf{w} \mid \mathbf{x},\mathbf{y}) \parallel \log p(\mathbf{w} \mid \mathbf{y})\right) \\
&+\beta_3 D_{KL}\left(q_\phi(\mathbf{z} \mid \mathbf{x},\mathbf{y}) \parallel p(\mathbf{z})\right) + \beta_4 \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})\mathcal{D}(\mathbf{x})}\left[\int_Y q_\delta(\mathbf{y} \mid \mathbf{z}) \log q_\delta(\mathbf{y} \mid \mathbf{z})\, d\mathbf{y}\right] \\
&-\log p(\mathbf{y}) \\
&\qquad - \beta_5 \mathbb{E}_{q(\mathbf{z}|\mathbf{x})\mathcal{D}(\mathbf{x},\mathbf{y})}\left[\log q_\delta(\mathbf{y} \mid \mathbf{z})\right].
\end{aligned}$$

The first loss is composed of, consecutively, data term, 2 KL terms corresponding to the single KL term in the usual VAE, MI minimization term and log prior density which is not accounted for during training. MI minimization term is only optimized with regard to ϕ (encoder parameters). MI minimization term acts as a generator loss in the adversarial procedure.

The second loss is for y to be reconstructed better from z and is only optimized with regard to δ (decoder parameters). It acts as a discriminator loss in the adversarial procedure.

3. Experiments

3.1. CSVAE: toy data

Toy data experiments were neeeded for validation of model architecture that was built in the course of the project.

The CSVAE paper is not clear on network architectures, on connections between networks and on optimizers that they used. We decided to implement the networks for CSVAE-like modeling as follows.

Usual VAE encoder is split in two (to discriminatively model z and w) and while distribution on part w of latent space is inferred based on both x and y (object and its label), distribution on z is only inferred based on object x (because information about y ideally shouldn't leak there).

There are 2 decoders in our implementation: one is usual VAE decoder and another is a part that tries to guess label y from part w of latent space along with the adversarial part of the loss that tries to hamper its progress by maximizing the entropy of predictive distribution on y.

We use swiss roll from sklearn [7] for our simple and cheap experiments of model validation.

Object label y is binary in the case of this dataset and is artificially generated deterministically from object coordinates. Latent space is 4-dimensional: z1, z2, w1, w2. w1, w2 should have information about Y (to enrich reconstruction), but z1, z2 shouldn't have it.

We trained CSVAE on 10000 train samples from the dataset. To validate the structure of the latent space, we projected 10000 test samples on the latent space and also estimated KL-divergence between distributions of latent codes for different label values on the specified plane. KL-divergence was estimated from samples using a method described in [8] which was based on KD-Tree.

Figures 1, 2, 3 show that only the complete CSVAE with adversarial components out of all networks that were trained on the toy data structures the latent space in the way we want to structure it: z plane representation does not distinguish between different labels and all the information about distinguishing them for reconstruction is located in w plane. This corroborates the hypothesis that this model structures the latent space in the desired way.

3.2. CelebA evaluation

CelebA [6] dataset consists of 200,000 images of celebrity faces with 40 labelled attributes. For the experiments we used just 19 most visually distinguishable attributes, such as 'Eyeglasses', 'Wearing Hat', 'Mustache', 'Male'. For quantitative estimates of generation performance we trained the model for classification of attributes and estimated the performance with a proxy measure of accuracy of the classifier on samples with chosen attributes.

Performance of this discriminative model on ground truth images (which actually possess the required attributes) is listed in the table below.

Accuracy of trained classifier on ground truth images	
Attribute name	Accuracy
Heavy Makeup	0.90
Wearing Hat	0.99
Male	0.97
Smiling	0.91
Eyeglasses	0.99

Figure 1: Latent space projections of VAE

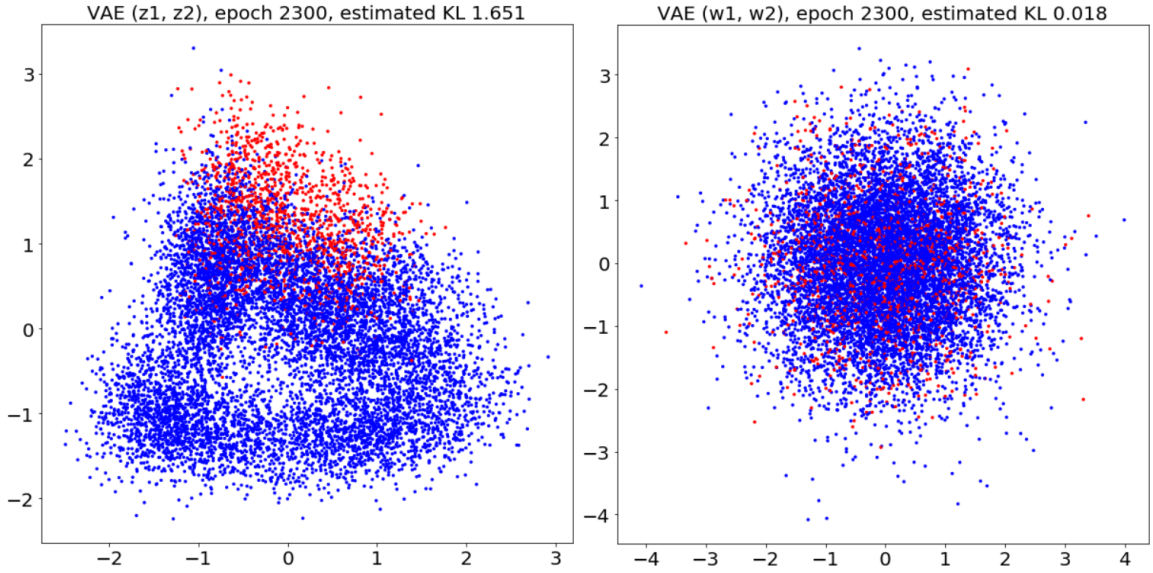
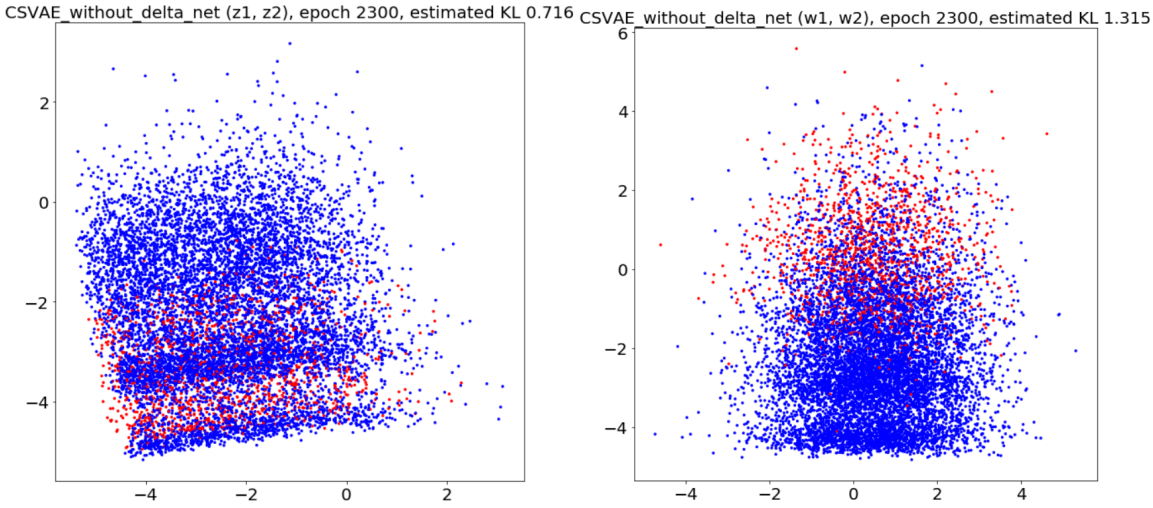


Figure 2: Latent space projections of CSVAE without adversarial component



3.2.1. CondVAE

We first trained base VAE. We used same architecture and pretrained weights for CSVAE and all other experiments. As in CSVAE paper, we used an effective convolutional VAE architecture described in another paper [5] for all our CelebA experiments.

The examples of samples generation with different attributes sets are shown in Figure 5. The accuracy of the generation for four chosen attributes are presented in the Table 3.2.3.

Figure 3: Latent space projections of CSVAE

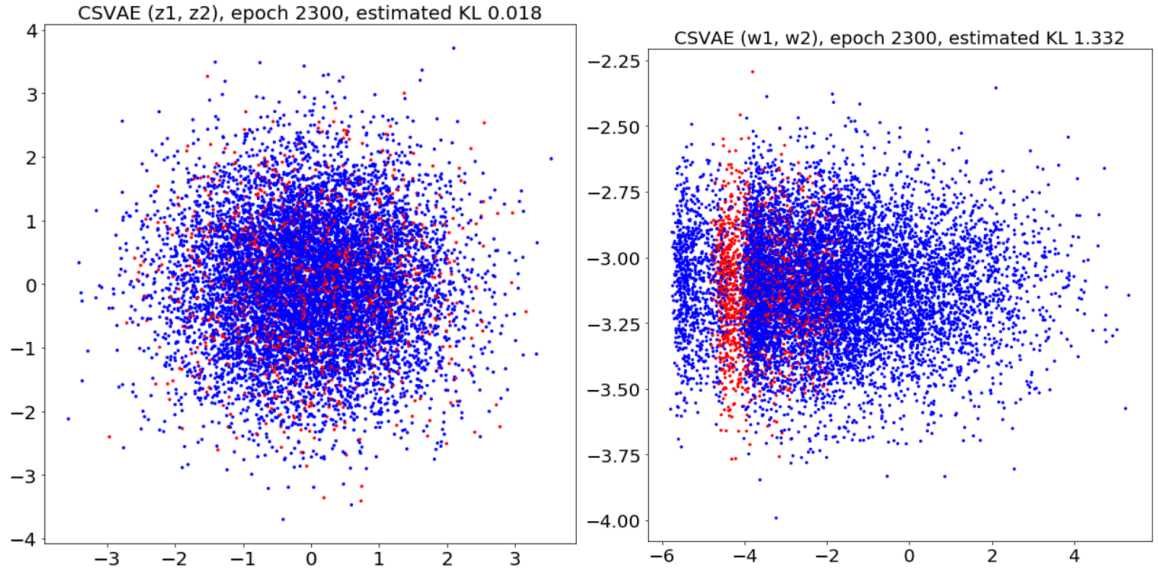


Figure 4: CondVAE: the examples of generation images with attributes: left - 'Male', center - 'Eyeglasses' and 'Smiling', right - 'Heavy Makeup'.

Figure 5: CondVAE: the examples of interpolation between sample without attribute 'Smiling' and sample with this attribute.

The interesting thing is that even the generation of samples with 'Eyeglasses' visually looks good, the accuracy of such generation is low. The very good results were shown by generating samples with "Male" and "Smiling attributes".

3.2.2. CondVAE-info

We also compared CSVAE with CondVAE-info model in the form of an architecture called IFcVAEGAN [1]. During training we used all the hyperparameters that were described in the article, however, we failed to obtain a qualitatively good result. During training, the generation slipped into the average value and the same face was generated, as shown in 6. Perhaps, if we wait a little longer, we will get a better result, but due to the huge number of hyperparameters it takes a lot more time.

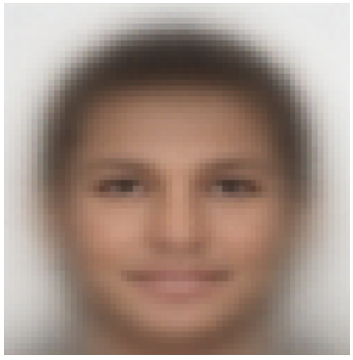


Figure 6: IFcVAEGAN generation sample

3.2.3. CSVAE

We obtained quantitative estimations of the performance of generative model by training an accurate classification model for attributes and subsequently measuring the accuracy on generated samples. We quantified the ability of generation of samples with specific attributes as well as attribute transfer.

The results indeed show higher reconstruction quality for CondVAE which suggests a possible trade-off between reconstruction quality and imposing a desirable latent space structure as one explanation.

Accuracy of the classifier on generated samples		
Attribute name	CondVAE	CSVAE
Wearing Hat	0.256	0.097
Eyeglasses	0.3639	0.121
Smiling	0.877	0.826
Male	0.754	0.680
Heavy Makeup	0.625	0.444

The plate clearly shows that the generation quality of method CondVAE is higher than method CSVAE. However, for many attributes, the generation accuracy is rather poor-quality, this may be due to the fact that the generated images are rather blurry and our classifier is overfitting on the celebA dataset domain.

Figure 7: CSVAE: the examples of generation images with attributes: left - 'Male', center - 'Eyeglasses' and 'Smiling', right - 'Heavy Makeup'. Visually the results of CondVAE can give an impression for being better and we trained a classifier for corroboration of this result.

4. Conclusions

CondVAE can provide more accurate samples and the model is easier. Despite this, CSVAE has a more controllable latent space with richer attributes. This comes at the expense of increasing difficulty of tuning, training and building models. CSVAE turned out to be more difficult to train on complex tasks (e.g. CelebA).

5. Contributions

Artem Shafarostov: experiments with CondVAE and implementation and experements with CondVAE-info, style transfer

Marina Pominova: experiments with CSVAE, CondVAE and alternative architectures on CelebA, tuning and interpretation

Alexander Lyzhov: implementation CSVAE with adversarial components (taking the most primitive VAE model as a base to build on), testing on toy data, visualizations of latent space, their qualitative and quantitative measurement and interpretation.

Elizaveta Lazareva: implementation VAE and CondVAE architectures, training accurate VAE for the further experiments, visualization of the results of the conditional generation and style transfer, training the attribute classifier for quantitative estimation of models performance.

References

[1] Antonia Creswell et al. "Adversarial Information Factorization". In: *arXiv preprint arXiv:1711.05175* (2017).

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

[3] Durk P Kingma et al. "Semi-supervised learning with deep generative models". In: *Advances in neural information processing systems*. 2014, pp. 3581–3589.

[4] Jack Klys, Jake Snell, and Richard Zemel. "Learning latent subspaces in variational autoencoders". In: *Advances in Neural Information Processing Systems*. 2018, pp. 6444–6454.

[5] Anders Boesen Lindbo Larsen et al. "Autoencoding beyond pixels using a learned similarity metric". In: *arXiv preprint arXiv:1512.09300* (2015).

[6] Ziwei Liu et al. "Deep learning face attributes in the wild". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 3730–3738.

[7] Fabian Pedregosa et al. "Scikit-learn: Machine learning in Python". In: *Journal of machine learning research* 12.Oct (2011), pp. 2825–2830.

[8] Qing Wang, Sanjeev R Kulkarni, and Sergio Verdú. "Divergence estimation for multidimensional densities via k-nearest-neighbor distances". In: *IEEE Transactions on Information Theory* 55.5 (2009), pp. 2392–2405.