Semi-Supervised Latent Tree Variational AutoEncoders with Learnable Dependencies

Abhinav Jain
Department of Computer Science
Rice University
aj70@rice.edu

Abstract

With the prevalence of Variational Auto-Encoders in learning the representation of observed data in the latent space, much of the recent prior-art has been dedicated to organizing the structure of latent space. The highly-opted approach has been to constrain how the variables in the latent space interact with each other. In this paper, we relax the constraints and make such interactions learnable. In other words, we let the latent space organize itself with the observed data. Particularly, we demonstrate this on CELEB-A dataset where we show how the latent space organisation affects the classification performance under different supervised settings.

1 Introduction

Classic Auto-Encoders aim to learn the distribution of observed data in latent space. Variational Autoencoders are special class of auto-encoders that encode inputs as distributions instead of points and whose latent-space organization is regularised by constraining latent distribution returned by the encoder to be close to standard Gaussian. They solve $arg_{\phi} \min\ KL(q_{\phi}(z|x)||p(z|x))$ by looking for an efficient encoding-decoding scheme where for given x, we want to maximize the probability of x_{recons} = x when we sample z from $q_{\phi}(z|x)$ and then sample x_{recons} from the generative distribution $p_{\theta}(x|z)$, where the posterior and generative distribution are approximated by the encoder-decoder respectively. The objective that comes out is the Evidence Lower Bound on log-likelihood of the observed data i.e. $ELBO$ = $\max \mathbb{E}_{q(z|x)}[\frac{p(x,z)}{q(z|x)}]$.

Over the last decade, various efforts have been put by the scientific community to better structure the latent space in order to learn meaningful representations of observed data. For instance, [6] assumes the latent space to posses a Ladder-like graphical structure while [1] assumes a fully-connected Bayesian latent graph with learnable dependencies. Many frameworks [4], [2] have also tried to associate certain latent factors with discrete labels to cluster different characteristics of observed data within categorical variables. They organize latent space into a hierarchy of z and y where z is the immediate latent space learned from x and y clusters z into different facets. [4] uses GMM to do unsupervised clustering of z while [2] follows one-to-one mapping between z and y to force each dimension of z to learn a feature relevant to a particular label. Moreover, some frameworks like [3], [2] use limited observed labels to supervise the learning of VAEs in a semi-supervised setting.

In this paper, we relax the assumptions made in the prior-art that enforce certain structure within the latent space and gate the dependencies between characteristics-capturing latent variables/attributes z and the discrete labels y. This is different from [2] where each attribute z is forced to contribute to only one label y. Contrary to this, by letting z capture attributes of x and learning which attributes are relevant to classifying x into particular label y, we allow multiple attributes to contribute to each y and also allow simultaneous sharing of attributes across labels (See 1). We demonstrate this on CELEB-A dataset [5] in a semi-supervised setting where some images carry multiple-labels while some do not have any label information.

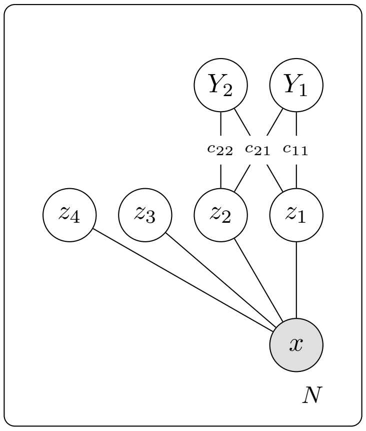

Figure 1: Graphical Model for gated LT-VAE

2 Methodology

With gated-LT-VAE, we want to learn certain characteristics of x through z_i and then associate them with labels Y_j, instead of directly inferring z as labels unlike [3] (See 1). We further want to learn which characteristics are important for predicting a certain label i.e. we want to learn the dependencies in the latent graph comprising z_i and Y_j in semi-supervised setting. Here, **z** acts as continuous latent variables capturing characteristics of x and **Y** are the discrete latent variables [4] acting as labels of x. We introduce binary variables $c_{i,j}$ to gate the dependencies between (Y_j, z_i) [1]. Each $c_{i,j}$ is sampled independently from a Bernoulli Distribution with learnable mean $\boldsymbol{\mu_{i,j}}$ using the Gumbel-Softmax Trick. Additionally, we also retain z with no connections to any Y to capture style attributes of data that can not be explained by given labels similar to [2].

In the semi-supervised objective we have the following components of the overall ELBO that we are trying to maximise:-

$$\max ELBO = \sum_{x} L_u(x) + \sum_{x,y} L_s(x,y) \tag{1}$$

2.1 Unsupervised Objective

In this setting, (z, y, c) are latent and x is observed. Thus we have the following ELBO:-

$$\begin{aligned} ELBO &= \mathbb{E}_{q(z,c,y|x)} \log \frac{p(x,z,c,y)}{q(z,c,y|x)} \\ &= \mathbb{E}_{q(z|x)q(c|z,x)q(y|c,z)} \log \frac{p(y)p(c)p(z|c,y)p(x|z)}{q(z|x)q(c|z,x)q(y|c,z)} \end{aligned}$$

Since sampling c establishes the attribute-label graph topology, we argue it to be indep. of x and we sample c once every inference-generation cycle to retain the same dependency between z, y Thus, $q(c|z,x) = q(c) = p(c)$

$$\begin{aligned} \implies ELBO &= \mathbb{E}_{p(c)}[\mathbb{E}_{q(z|x)q(y|c,z)} \log \frac{p(y)p(z|c,y)p(x|z)}{q(z|x)q(y|c,z)}] \\ &= \mathbb{E}_{c' \sim p(c)}[\mathrm{L}_{u_{c'}}] \end{aligned}$$

where, $\mathrm{L}_{u_{c'}}$ is the unsupervised ELBO for 1 with the fixed latent dependency as specified by c'.

Gating Scheme	$f = 1.0$	$f = 0.5$	$f = 0.2$
one-one	**81.4**	**74.8**	68.0
inferred	76.2	74.6	65.9
learnable	75.5	74.0	**72.8**

Table 1: Test Accuracies on CELEB-A for each gating scheme across different supervision settings

2.2 Supervised Objective

In this setting, (z, c) are latent and (x, y) are observed. Thus we have the following ELBO:-

$$\begin{aligned}
ELBO &= \mathbb{E}_{q(z,c|x,y)} \log \frac{p(x,z,c,y)}{q(z,c|x,y)} \\
&(\because q(z,c|x,y) = \frac{q(x,z,c,y)}{q(x,y)} = \frac{q(z|x)q(y|c,z)p(c)}{q(y|x)}) \\
&= \mathbb{E}_{\frac{q(z|x)q(y|c,z)p(c)}{q(y|x)}} \log \frac{p(y)p(z|c,y)p(x|z)q(y|x)}{q(z|x)q(y|c,z)} \\
&= \log p(y) + \log q(y|x) + \mathbb{E}_{p(c)q(z|x)} \frac{q(y|c,z)}{q(y|x)} \log \frac{p(x|z)p(z|c,y)}{q(z|x)q(y|c,z)} \\
&= \mathbb{E}_{p(c)} [\mathbb{E}_{q(z|x)} \frac{q(y|c,z)}{q(y|x)} \log \frac{p(x|z)p(z|c,y)}{q(z|x)q(y|c,z)}] + \log p(y) + \log q(y|x) \\
&= \mathbb{E}_{c' \sim p(c)} [\mathrm{L}_{s_{c'}}] + \log p(y) + \log q(y|x)
\end{aligned}$$

where, $\mathrm{L}_{u_{c'}}$ is the supervised ELBO for 1 with the fixed latent dependency as specified by c'. We can also see that classification probability $q(y|x)$ also comes out naturally.

3 Experiments

In this section, we highlight the experiment details and ablation study on the CELEB-A Dataset which consists of 202.6k images with 40 binary labels for each sample. We selected 18 easy labels to keep the association of attributes and labels simple.

3.1 Gating Schemes

Let's call edges between (y_i, z_i) as '*straight*' and $(y_i, z_j),\ i \neq j$ as '*diagonal*' $\forall 1 \leq i, j \leq L$ (For the remainder of the paper we will be ignoring the style capturing z unconnected to any Y for discussion purposes). Now we can realise different latent gating schemes and categorise them as follows :- (a) *fixed, one-to-one*: In this gating scheme, we connect each label y to a single z unit using straight edges (we can call them straight wlog) to obtain fixed μ_c (with diagonal entries as 1). (b) *fixed, inferred*: In this scheme, we infer the μ_c from observed labels as explained in the next section and keep it fixed. (c) *learnable*: In this scheme, we initialise μ_c similarly but fine-tune it while learning the parameters of LT-VAE. Here, we also add L1-regularisation i.e. $\lambda_c ||\mu_c||_1$ to the overall loss in order to encourage sparsity among gates which ensures the gates between un-associated label-attribute pairs get closed.

3.2 Estimating Initial Probability of Gating Variables

To estimate the initial (gating) probability, we obtain the co-occurrence matrix of observed labels i.e. if (y_i, y_j) tend to co-occur across the dataset, they are more likely to have certain attributes z of x in common. We then normalise the matrix with total number of samples to obtain co-occurrence probabilities. Finally, we set diagonal entries to 1 to ensure that each label is always connected to its unique attribute which can be shared with other labels. Formally we will always have a mapping (a straight edge) between (z_i, y_i) with $\mu_{i,i} = 1$ while the mapping (a diagonal edge) between $(z_j, y_k),\ j \neq k$ i.e. $\mu_{j,k} = cooccur(y_j, y_k)$ is learned using co-occurrence of label pairs.

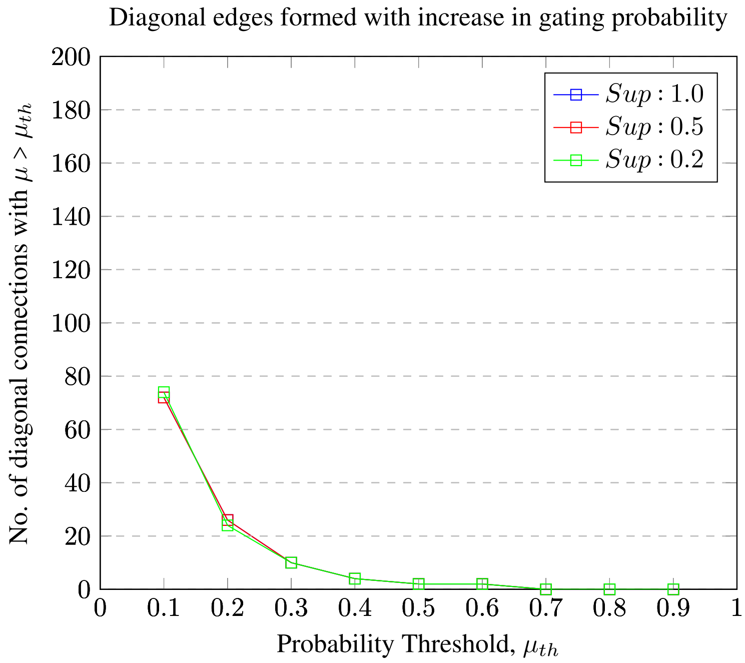

Figure 2: Plot for the gating probabilites in 'fixed-inferred' gating case

3.3 Implementation Details

In this section, we provide the necessary implementation details. We experiment with three supervision settings $f = 1.0, 0.5, 0.2$. For each setting, only the observed labels are used to calculate the $\mu_{c_{init}}$. The implementation is done in Tensorflow 2.0, for which we have made the code publicly available[1].

3.3.1 Probability Measures

We realise each probability measure identified in the overall objective as - (a) The prior over the latent gates is a Bernoulli distribution, $Bernoulli(\mu_c)$; (b) the posterior over $z, q(z|x) = Gaussian(\mu_\phi(x), \sigma_\phi(x))$, where ϕ are the parameters of the Encoder Network and $z \sim q(z|x)$ is done using the reparameterisation trick; (c) the posterior over binary labels $y, q(y|z,c) = Bernoulli(\pi_\Phi(z * c))$, where Φ are the parameters of the Classifier Network; (d) the generative distribution over z $p(z|y,c) = Gaussian(\mu_\omega(y * c), \sigma_\omega(y * c))$ where ω are the parameters of the Conditional Prior Network; (e) the generative distribution over $x, p(x|z) = Laplacian(\mu_\theta(z))$ where θ are the parameteres of the Decoder Network. Finally, prior over labels $p(y) = Uniform(0.5)$.

3.3.2 Network Architecture

While the encoder-decoder networks can be realised using convolutional and transposed convolutional layers respectively, the classifier and conditonal prior networks have to be custom-designed to incorporate opening/closing of gates. In the classifier network, a dense layer maps z_i to y_j $\forall 1 \leq i, j \leq L$, however since the input to compute each y_j i.e. $z_{i,j} * c_{i,j}$ is changing with j, we compute the logits for y_j as $\sum_i W_{\Phi_{i,j}} * z_{i,j} * c_{i,j} + b_{\Phi_j}$ by replicating z such that $z_{i,j} = z_i$ $\forall j$, where '$*'$ denotes element-wise multiplication. We similarly do this in the reverse direction for Conditional Prior Network i.e. $\mu_{\omega_i} = \sum_j W_{1_{i,j}} * c^T_{i,j} * y_{i,j} + W_{2_{i,j}} * c^T_{i,j} * (1 - y_{i,j})$ and $\sigma_{\omega_i} = \sum_j W_{3_{i,j}} * c^T_{i,j} * y_{i,j} + W_{4_{i,j}} * c^T_{i,j} * (1 - y_{i,j})$, where $y_{i,j} = y_j$ $\forall i$.

[1] Code: https://github.com/jabhinav/Semi-Supervised-Gated-LT-VAE/tree/master

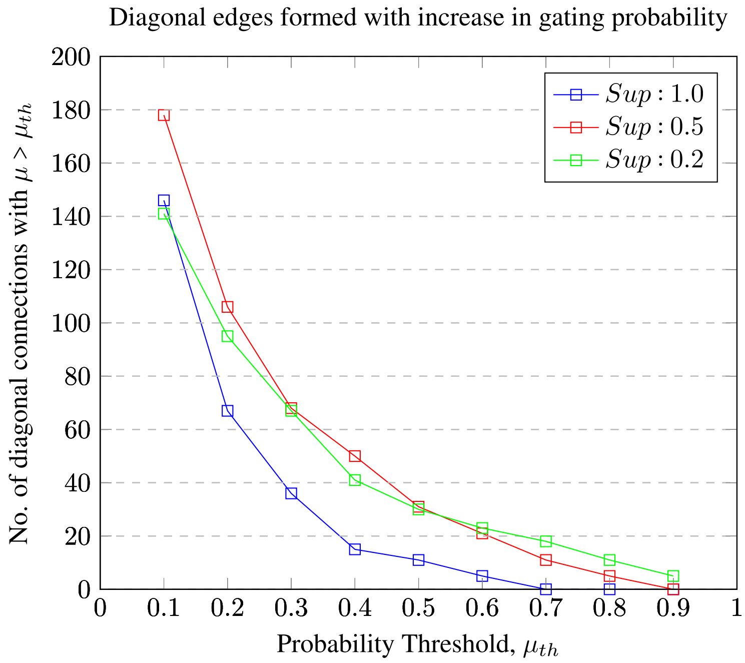

Figure 3: Plot for the learned gating probabilites in the 'learnable' gating case

3.3.3 Training Configuration

Due to limited computation resources, we were only able to train the model for 50 epochs. We use the 80-10-10 split of the dataset and save the model with the best validation accuracy. The dimension of z is set to be 45 where, $z_{classify}$ = 18 = $NumLabels$ and z_{style} unconnected to any y becomes 27. We use the Adam optimizer with initial learning rate of 0.0001 which is annealed at the rate of 0.00003 per epoch. The initial gating temperature for sampling gates using the gumbel softmax is set to 1.0 under the 'learnable' gating scheme and to 0.3 under the 'fixed' gating scheme. In the learnable case, the temperature is reduced each epoch to recover binary values for c yielding the desired discrete sampling of dependency structures. Lastly, the images are normalised and resized to $(64, 64)$.

4 Results

In Table 1 we present the results of evaluating the best model with highest validation accuracy on the test set. We can observe that the one-one gating scheme with no diagonal connections performs the best in settings with high supervision. However, the gap starts to diminish with drop in supervised data until the learnable gating scheme starts to perform better (f = 0.2). To make better sense of this, we plot number of number of diagonal gating connections that start to appear with high probability (see Fig. 2 vs Fig. 3).

We can see that the inferred gating probability matrix is almost similar across all supervision settings. However, when fine-tuned, the model starts to assign higher probabilities to many diagonal connections. We can argue that there might be some attributes of the observed data that should be shared across some labels, which intuitively makes sense. However, in fully-supervised setting some learned connections might be superfluous which could explain the drop in accuracy. On the other hand, in limited supervision, such connections might be required to better explain the observed data which might be contributing to higher accuracy. Importantly in limited supervision, enforcing one-one gating constraints is probably limiting the expressability of the model and thus degrading the overall performance.

Another observation that we made was regarding the symmetricity of the gating matrix. The fixed-inferred gating matrix is symmetric i.e. if attribute z_i is shared between y_i with a straight connection and y_j with a diagonal connection, then so is attribute z_j having straight connection with y_j and diagonal with y_i. For instance, there is attribute sharing between label - *Young* and *No_Beard*. However, the learned gating matrix adds asymmetric connections such as attribute for predicting *Wavy_Hair* is shared with label *Blonde_Hair* but not vice-versa. One way of interpreting this would be that predicting label *Blonde_Hair* requires its own unique attribute as well as information available in the attribute for *Wavy_Hair* but not vice-versa.

5 Conclusion

In this paper, we formulated a formal approach to gating connections between attributes and labels of the given data in the latent space. After testing the performance of various gating schemes under different supervised settings, we concluded that attribute sharing between labels is indeed useful for the LT-VAE model to perform better, however controlling the opening-closing of gated connections better than simply using L_1 norm is desirable and left as part of future work.

References

[1] Jiawei He, Yu Gong, Joseph Marino, Greg Mori, and Andreas Lehrmann. Variational autoencoders with jointly optimized latent dependency structure. In International Conference on Learning Representations, 2018.

[2] Tom Joy, Sebastian Schmon, Philip Torr, N Siddharth, and Tom Rainforth. Capturing label characteristics in vaes. In International Conference on Learning Representations, 2020.

[3] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In Advances in neural information processing systems, pages 3581–3589, 2014.

[4] Xiaopeng Li, Zhourong Chen, Leonard KM Poon, and Nevin L Zhang. Learning latent superstructures in variational autoencoders for deep multidimensional clustering. arXiv preprint arXiv:1803.05206, 2018.

[5] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In Proceedings of the IEEE international conference on computer vision, pages 3730–3738, 2015.

[6] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. Advances in neural information processing systems, 29:3738–3746, 2016.