Appendix of Commutative Lie Group VAE for Disentanglement Learning

Xinqi Zhu   Chang Xu   Dacheng Tao

1. Proof of Proposition 1

Proposition 1. *If* $A_iA_j = A_jA_i, \forall i, j$, *then*

$$\begin{aligned} &\exp(t_1A_1 + t_2A_2 + ... + t_mA_m) \\ &= \exp(t_1A_1)\exp(t_2A_2)...\exp(t_mA_m). \end{aligned} \quad (1)$$

Proof. By (Hall, 2013) Proposition 2.3, we have that if $XY = YX$, $\exp(X + Y) = \exp(X)\exp(Y) = \exp(Y)\exp(X)$.

We also have:

$$\begin{aligned} &t_1A_1(t_2A_2 + t_3A_3 + ... + t_mA_m) \\ &\quad - (t_2A_2 + t_3A_3 + ... + t_mA_m)t_1A_1 && (2) \\ &= (t_1A_1t_2A_2 - t_2A_2t_1A_1) + ... \\ &\quad + (t_1A_1t_mA_m - t_mA_mt_1A_1) && (3) \\ &= t_1t_2(A_1A_2 - A_2A_1) + ... \\ &\quad + t_1t_m(A_1A_m - A_mA_1) && (4) \\ &= 0, && (5) \\ &\Rightarrow t_1A_1(t_2A_2 + t_3A_3 + ... + t_mA_m) && (6) \\ &= (t_2A_2 + t_3A_3 + ... + t_mA_m)t_1A_1 && (7) \end{aligned}$$

Then we have:

$$\begin{aligned} &\exp(t_1A_1 + t_2A_2 + ... + t_mA_m) \\ &= \exp(t_1A_1)\exp(t_2A_2 + ... + t_mA_m) && (8) \\ &= \exp(t_2A_2 + ...t_mA_m)\exp(t_1A_1) && (9) \\ & && (10) \end{aligned}$$

Apply this to all terms (index > 1), we have:

$$\begin{aligned} &\exp(t_1A_1 + t_2A_2 + ... + t_mA_m) \\ &= \exp(t_1A_1)\exp(t_2A_2)...\exp(t_mA_m) \\ &= \prod_{\text{perm}(i)} \exp(t_iA_i). && (11) \end{aligned}$$

□

2. Proof of Proposition 2

Proposition 2. *If* $A_iA_j = 0, \forall i, j$, *then*

$$H_{ij} = \frac{\partial^2 g(\boldsymbol{t})}{\partial t_i \partial t_j} = 0, \quad (12)$$

where $g(\boldsymbol{t}) = \exp(t_1A_1 + t_2A_2 + ... + t_mA_m)$.

Proof. By (Rossmann, 2002) Theorem 5, we have:

$$\frac{d}{dt}\exp(X) = \exp(X)\frac{1 - \exp(-\text{ad}_X)}{\text{ad}_X}\frac{dX}{dt}, \quad (13)$$

where $X = X(t)$ is any matrix-valued differentiable function of a scalar variable t, and

$$\frac{1 - \exp(-\text{ad}_X)}{\text{ad}_X} = \sum_{k=0}^{\infty} \frac{(-1)^k}{(k+1)!}(\text{ad}_X)^k, \quad (14)$$

and $\text{ad}_X Y = [X, Y] = XY - YX$ is the adjoint action on Lie algebra (note that $(\text{ad}_X)^k Y = [X, [X, ..., [X, Y]]]$).

By denoting $\boldsymbol{A} = t_1A_1 + t_2A_2 + ... + t_mA_m$, we then have:

$$\begin{aligned} \frac{\partial g(\boldsymbol{t})}{\partial t_i} &= g(\boldsymbol{t})[\frac{(-1)^0}{1!}(\text{ad}_{\boldsymbol{A}})^0 + \frac{(-1)^1}{2!}(\text{ad}_{\boldsymbol{A}})^1 + && (15) \\ &\quad \frac{(-1)^2}{3!}(\text{ad}_{\boldsymbol{A}})^2 + ...]\frac{\partial(t_1A_1 + ... + t_mA_m)}{\partial t_i} && (16) \\ &= g(\boldsymbol{t})[A_i + (-\frac{1}{2})[\boldsymbol{A}, A_i] + \frac{1}{6}[\boldsymbol{A}, [\boldsymbol{A}, A_i]] + ...]. && (17) \end{aligned}$$

Since $A_iA_j = 0, \forall i, j$, we have:

$$\begin{aligned} &[\boldsymbol{A}, A_i] = t_1[A_1, A_i] + ... + t_m[A_m, A_i] && (18) \\ &= t_1(A_1A_i - A_iA_1) + ... + t_m(A_mA_i - A_iA_m) && (19) \\ &= 0 && (20) \\ &\Rightarrow \frac{\partial g(\boldsymbol{t})}{\partial t_i} = g(\boldsymbol{t})A_i, && (21) \\ &\Rightarrow H_{ij} = \frac{\partial^2 g(\boldsymbol{t})}{\partial t_i \partial t_j} = \frac{\partial}{\partial t_j}\Big(\frac{\partial g(\boldsymbol{t})}{\partial t_i}\Big) && (22) \\ &\quad = \frac{\partial g(\boldsymbol{t})A_i}{\partial t_j} = g(\boldsymbol{t})A_jA_i = 0. && (23) \end{aligned}$$

□

3. Introduction of Variational Autoencoder

The variational autoencoder (VAE) is a latent variable model to maximize the evidence lower bound (ELBO) of the intractable marginal log-likelihood of the training data:

$$\begin{aligned} &\log p(x) \geq \mathcal{L}(x, z) \\ &= \mathbb{E}_{q(z|x)}\log p(x|z) - KL(q(z|x)||p(z)), \end{aligned} \quad (24)$$

where an inference network $q(z|x)$ is introduced to estimate the posterior distribution $p(z|x)$. The $q(z|x)$ and $p(x|z)$ are constructed as two networks combined to form an stochastic autoencoder, where the $q(z|x)$ is usually modeled as a Gaussian distribution with means and standard deviations being the outputs of a neural network. The prior distribution $p(z)$ is usually fixed to be a standard Gaussian distribution. The first term in Eq. 24 is modeled as a reconstruction loss of the input x, and the second term is the KL divergence computed usually in a closed form if both distributions are Gaussian distributions.

4. Proof of Proposition 3

Proposition 3. *Suppose two latent variables z and t are used to model the log-likelihood of data x, then we have:*

$$\begin{aligned}
&\log p(x) \geq \mathcal{L}_{bottleneck}(x,z,t)\\
&= \mathbb{E}_{q(z|x)}\mathbb{E}_{q(t|x,z)}\log p(x,z|t)\\
&\quad - \mathbb{E}_{q(z|x)}KL(q(t|x,z)||p(t)) - \mathbb{E}_{q(z|x)}\log q(z|x) \qquad (25)\\
&= \mathbb{E}_{q(z|x)q(t|z)}\log p(x|z)p(z|t)\\
&\quad - \mathbb{E}_{q(z|x)}KL(q(t|z)||p(t)) - \mathbb{E}_{q(z|x)}\log q(z|x), \qquad (26)
\end{aligned}$$

where Eq. 26 holds because we assume Markov property: $q(t|z) = q(t|x,z)$, $p(x|z,t) = p(x|z)$.

Proof.

$$\log p(x) = \log \int_z \int_t p(x,z,t) \tag{27}$$

$$= \log \int_z \int_t p(x,z,t)\frac{q(t|x,z)q(z|x)}{q(t|x,z)q(z|x)} \tag{28}$$

$$\geq \int_z q(z|x) \log \int_t p(x,z,t)\frac{q(t|x,z)}{q(t|x,z)q(z|x)} \tag{29}$$

$$= \int_z q(z|x) \log \int_t p(x,z,t)\frac{q(t|x,z)}{q(t|x,z)} - \mathbb{E}_{q(z|x)}\log q(z|x) \tag{30}$$

$$\geq \int_z q(z|x) \int_t q(t|x,z) \log \frac{p(x,z,t)}{q(t|x,z)} - \mathbb{E}_{q(z|x)}\log q(z|x) \tag{31}$$

$$= \int_z q(z|x) \int_t q(t|x,z) \log \frac{p(x,z|t)p(t)}{q(t|x,z)} - \mathbb{E}_{q(z|x)}\log q(z|x) \tag{32}$$

$$= \mathbb{E}_{q(z|x)}\mathbb{E}_{q(t|x,z)}\log p(x,z|t) - \mathbb{E}_{q(z|x)}\int_t q(t|x,z)\log\frac{q(t|x,z)}{p(t)} - \mathbb{E}_{q(z|x)}\log q(z|x) \tag{33}$$

$$= \mathbb{E}_{q(z|x)}\mathbb{E}_{q(t|x,z)}\log p(x,z|t) - \mathbb{E}_{q(z|x)}KL(q(t|x,z)||p(t)) - \mathbb{E}_{q(z|x)}\log q(z|x) \tag{34}$$

$$= \mathbb{E}_{q(z|x)q(t|z)}\log p(x|z)p(z|t) - \mathbb{E}_{q(z|x)}KL(q(t|z)||p(t)) - \mathbb{E}_{q(z|x)}\log q(z|x), \tag{35}$$

where Eq. 29 and 31 are due to the Jensen's inequality, and Eq. 35 holds because we assume Markov property: $q(t|z) = q(t|x,z)$, $p(x|z,t) = p(x|z)$. □

Encoder-64
$64 \times 64 \times$ n_channel
4×4 Conv. 32, ReLU, Stride 2
4×4 Conv. 32, ReLU, Stride 2
4×4 Conv. 64, ReLU, Stride 2
4×4 Conv. 64, ReLU, Stride 2
Flatten + FC. 256, ReLU
FC. group_size
FC. group_size $\times 4$, ReLU
FC. $2 \times$ n_latent

Table 1. Encoder architecture on Dsprites, 3DShapes, CelebA and 3DChairs datasets.

Decoder-64
Latent code $\in \mathbb{R}^{\text{n\_latent}}$
Multiply Lie Algebra Basis + Matrix Exponential
Flatten + FC. 256, ReLU
FC. $4 \times 4 \times 64$, ReLU
4×4 Deconv. 64, ReLU, Stride 2
4×4 Deconv. 32, ReLU, Stride 2
4×4 Deconv. 32, ReLU, Stride 2
4×4 Deconv. n_channel, Sigmoid, Stride 2

Table 2. Decoder architecture on Dsprites and 3DShapes, CelebA and 3DChairs datasets.

5. Implementation Details

The Lie Group VAE Encoder and Decoder architectures used on all datasets are shown in Table 1, 2, 3, and 4. These architectures all inherit from the architectures proposed in FactorVAE (Kim & Mnih, 2018) with small modifications to support the integration of Lie Group constraints proposed in this paper. Specifically, two FC layers are added in the encoder to obtain the inferred group representation and latent code, while Lie Algebra multiplication and matrix exponential mapping are added in the decoder. The convolution layers are kept the same as in (Kim & Mnih, 2018). The $loss_{\text{rec\_group}}$ is scaled by 0.1 on all datasets. On 3DShapes and CelebA the group size is set to 400, while on other datasets we use 100.

Encoder-32
$32 \times 32 \times$ n_channel
4×4 Conv. 32, ReLU, Stride 2
4×4 Conv. 32, ReLU, Stride 2
4×4 Conv. 64, ReLU, Stride 2
Flatten + FC. 256, ReLU
FC. group_size
FC. group_size $\times 4$, ReLU
FC. $2 \times$ n_latent

Table 3. Encoder architecture on Mnist dataset.

Decoder-32
Latent code $\in \mathbb{R}^{\text{n\_latent}}$
Multiply Lie Algebra Basis + Matrix Exponential
Flatten + FC. 256, ReLU
FC. $4 \times 4 \times 64$, ReLU
4×4 Deconv. 32, ReLU, Stride 2
4×4 Deconv. 32, ReLU, Stride 2
4×4 Deconv. n_channel, Sigmoid, Stride 2

Table 4. Decoder architecture on Mnist dataset.

References

Hall, B. C. *Lie Groups, Lie Algebras, and Representations*. Springer New York, New York, NY, 2013.

Kim, H. and Mnih, A. Disentangling by factorising. In *ICML*, 2018.

Rossmann, W. *Lie groups : an introduction through linear groups*. Oxford graduate texts in mathematics ; 5. Oxford University Press, Oxford ;, 2002. ISBN 0198596839.