A Remedy For Negative Monte Carlo Estimated Values Of KL-divergence

Jiaxu Liu

A report submitted for the course
COMP4560
Supervised by: Tom Gedeon, Zhenyue Qin
The Australian National University

June 2022

Except where otherwise indicated, this report is my own original work.

Jiaxu Liu
6 June 2022

Acknowledgments

I owe many debts of thanks. First to my project supervisors, Professor Tom Gedeon and Mr. Zhenyue Qin, I thank Tom for taking me under his wing and giving me great freedom to explore, I thank Zhenyue for sharing thoughtful and calm guidance throughout, also he offered me opportunities to collaborate with wonderful people, which I felt I had been privileged to.

Among the many teaching staff that have brightened my time at Canberra, I thank Professor Hanna Kurniawati, who offered me the opportunity to teach in the Algorithm class at ANU, which was a terrific experience.

To the guy who shared a dormitory with me - Ruikai Cui - and the guy who is keen to gossip with me - Yukang Liu - thank you for your generosity, patience and friendship. To everyone else that has offered me great support during my stay in Australia - and all of my classmates from Shandong University - thank you all for your help and many valuable discussions.

I thank my most cherished friends. Jiyu Qi, from Chinese Academy of Sciences, majored in pure mathematics, the only genius I know, who has offered me countless guidance on mathematics, which I appreciate so much. Rui Qian, who was my teammate in the ACM school team, a thinker, a scholar. Tianze Wang, with whom we shared a common perspective, but headed down different paths. Haonan Sun, from Zhejiang University, Boyang Hui, from Beijing Jiaotong University, these two guys have always been there for me since childhood, whether we live in the same city or in a different country. I owe them a lot.

I thank all those who have brought me pain and happiness, and, most importantly, I thank all my family for their constant patience, wisdom and warm support. To my father Liu-Yingwu and my mother Wang-Jin, I am grateful to them as always, they used to live tough life.

For my family

Abstract

It is of great significance in the domain of machine learning research to learn independent, separable and disentangled latent representations of data. For instance, the Variational Auto-Encoder (VAE) is a scalable model for learning latent variable of complex data, and is attracting increasing attention. The objective function in such models usually employs the Kullback-Leibler divergence (KL-divergence) for implementable optimization. The KL-divergence has a broad range of applications in information theory, statistics and machine learning. For example, the KL-divergence is applicable in speech recognition, multimedia generation, hypothesis testing, text classification, just to name a few.

Although KL-divergence is widely used as an objective function in machine learning, computing its analytical form is challenging. Hence, Monte Carlo techniques are standard approaches to estimate the KL-divergence. However, the unbiased estimation of Monte Carlo methods requires unrealistically infinite available data samples. In contrast, standard small sample sizes can cause negative estimation of KL-divergence, whereas KL-divergence is non-negative by definition. Thus, such non-negativity of estimated KL-divergence results in severe noise and high variance. In this thesis, we propose new estimators of KL-divergence, namely absolute and tangent estimators. We formulate new divergence using these estimators and propose AbsoluteKL-divergence and TangentKL-divergence, abbreviated as AKL- and TKL-divergence. They are new quantities to measure the difference between two distributions, with a similar form to KL-divergence. The experimental results demonstrate that the Monte Carlo estimations of our proposed divergence maintain non-negativity, facilitating more accurate approximations than with KL-divergence.

Moreover, we show that AKL-divergence is a semi-metric. Therefore, it is a rational quantity for measuring the difference between two distributions. We also demonstrate the generalization from TKL-divergence to the family of f-divergence. Furthermore, replacing KL-divergence with AKL- and TKL-divergence, we introduce new objective functions for VAEs. Experimental results exhibit lower reconstruction errors and more realistic generated images.

Keywords: KL-divergence, Variational Inference, Variational Autoencoder, Information Theory, Probabilistic Graphical Models, Latent Variable Model, Neural Networks, Representation Learning, Monte-Carlo Estimation

Contents

Acknowledgments **iii**

Abstract **v**

1 Introduction **1**

2 Theoretic Background **3**
2.1 Probabilistic Modeling . . . 3
2.1.1 Intuition . . . 3
2.1.2 Probabilistic Models . . . 4
2.2 Probabilistic Graphical Models . . . 5
2.2.1 Directed Graphical Models . . . 5
2.3 Inference . . . 6
2.3.1 Sampling method . . . 7
2.3.2 Variational Method . . . 9
2.4 Probabilistic Learning . . . 12
2.4.1 Maximum Likelihood . . . 12
2.4.2 Latent Variable . . . 13
2.5 Summary . . . 13

3 Framework **15**
3.1 Deep Generative Models . . . 15
3.2 The Variational Bound . . . 16
3.2.1 Reformulation of Evidence Lower-bound . . . 16
3.2.2 The Naïve Estimator . . . 17
3.3 Encoder and Decoder . . . 17
3.4 The Reparameterization . . . 18
3.5 Auto-Encoding Variational Bayes . . . 20
3.6 The Critical Problem . . . 22
3.7 Summary . . . 23

4 Proposed Estimators **25**
4.1 Reminder: Naïve Estimator . . . 25
4.2 Proposal I: Absolute Estimator . . . 26
4.2.1 Motivation . . . 26
4.2.2 $\hat{\theta}_{\text{Abs}}$: An Alleviation To High Variance . . . 27
4.2.3 Case Study . . . 27

4.2.4 D_{AKL}: Absolute-KL-divergence . . . 28
4.3 Proposal II: Tangent Estimator . . . 29
4.3.1 Motivation . . . 29
4.3.2 $\hat{\theta}_{\text{Tan}}$: Between Convex Function and Tangent Plane . . . 29
4.3.3 Case Study . . . 31
4.3.4 D_{TKL}: Tangent-KL-divergence . . . 32
4.3.5 Analogize To f-divergence . . . 32
4.4 Empirical Result . . . 33
4.5 Summary . . . 35

5 Application **37**
5.1 Lower Reconstruction Errors of Hyperbolic VAEs . . . 37
5.1.1 On Averaged Test Reconstruction Error . . . 38
5.1.2 On Different Curvatures Of Hyperbolic Manifold . . . 39
5.2 Realistic Human Appearance Generation . . . 40
5.3 Summary . . . 46

6 Conclusion **47**
6.1 Brief Summary . . . 47
6.2 Future Directions . . . 47

7 Appendix I **49**
7.1 Software platform . . . 49
7.2 Hardware platform . . . 49
7.3 Artifact . . . 49
7.3.1 Prerequisites . . . 49
7.3.2 Backbone Model . . . 49
7.3.3 Directory structure . . . 50
7.3.4 Trainning . . . 51
7.3.5 Acknowledgement . . . 52
7.3.6 Additional Links . . . 52

Bibliography **55**

List of Figures

2.1 A sample directed graphical model with four variables, visually specifying how random variables depend on each other in Eq.(2.9) 6

2.2 An example graphical notation of latent variable model, powered by directed graphical model. x is the observed data distribution. z is the imposed low-dimensional structure latent factor. θ is the parameter that dominates the variables. $\hat{x}$ is the reconstructed data in generative perspective $p(\hat{x}) = \int_z p(x|z)p(z)$ 13

3.1 Example of the ten handwritten digit images in the MNIST dataset . . . 15

3.2 Encoder model, part of parameters of MLP is represented as ϕ 17

3.3 Decoder model, part of parameters of MLP is represented as θ 18

3.4 Illustration of the reparameterization trick in order to perform back-propagate through random variable. [Kingma's NIPS 2015 workshop slides] 20

3.5 A very basic example of variational auto-encoder architecture (Gaussian case) with four convolutional layers as hidden layers in the encoder and decoder. The style of this illustration is inspired by [David Stutz, 2021], we use such architecture to run the experiment in Chapter.5 on the MNIST dataset and the Market-1501. In this case the convolution includes the batch normalization and we utilize ReLU [Nair and Hinton, 2010] as an example activation function to provide non-linearity. 21

4.1 Comparison of bias among estimators under continuous Bernoulli distribution with different settings of parameter λ where $p = \mathcal{CB}(0)$ and $q = \mathcal{CB}(\lambda)$. 33

4.2 Comparison of variance among estimators under continuous Bernoulli distribution with different settings of parameter λ where $p = \mathcal{CB}(0)$ and $q = \mathcal{CB}(\lambda)$. 33

4.3 Averaged result on variance of different estimators as *Mean-Shift* increase, the shaded area indicates the error for different sample sizes (range from $1e1$ to $1e8$) 34

5.1 Comparison of reconstruction loss of optimizing p-VAEs with KL-, AKL- and TKL-divergence where the weight $\beta = 1$, versus dimensionality of the latent space. 39

5.2 Comparison of reconstruction loss of optimizing p-VAEs with KL-, AKL- and TKL-divergence where the weight $\beta = 2$, versus dimensionality of the latent space. 39
5.7 Examples of human geometrical information y in the Market-1501 dataset [Zheng et al., 2015a] 40
5.3 Comparison of ELBO, reconstruction loss and divergence of optimizing p-VAEs with KL-divergence and AKL-divergence and TKL-divergence, vs. curvature $c = 0.1$. The latent dimension is set to be 40. 41
5.4 Comparison of ELBO, reconstruction loss and divergence of optimizing p-VAEs with KL-divergence and AKL-divergence and TKL-divergence, vs. curvature $c = 0.2$. The latent dimension is set to be 40. 42
5.5 Comparison of ELBO, reconstruction loss and divergence of optimizing p-VAEs with KL-divergence and AKL-divergence and TKL-divergence, vs. curvature $c = 0.7$. The latent dimension is set to be 40. 43
5.6 Comparison of ELBO, reconstruction loss and divergence of optimizing p-VAEs with KL-divergence and AKL-divergence and TKL-divergence, vs. curvature $c = 1.4$. The latent dimension is set to be 40. 44
5.8 Comparison of optimizing vunets with KL- and AKL- and TKL-divergence on the market-1501 dataset [Zheng et al., 2015a]. The first row illustrates the generated images of the vunet [Esser et al., 2018] trained with KL-divergence; the second and third row displays the generated results of the vunet trained with our proposed divergence. The *ite.* stands for iteration. 45

List of Tables

4.1 evaluation results of the bias and variance of different estimators under Beta, Gaussian and Laplace distributions with different parameter settings. The values marked in green are the best of the group and those in red are the worst. 36

7.1 Specification of the package requirements for each experiment. 53
7.2 Specification of the use of hardware for each experiment. 53

Chapter 1

Introduction

Kullback-Leibler divergence, abbreviated as KL-divergence, is a widely-used measure for evaluating the difference between two distributions [Malinin and Gales, 2019]. KL-divergence has many applications, examples including computing mutual information between two variables Belghazi et al. [2018] and devising objective functions of neural networks [Nowozin et al., 2016]. To some pairs of distributions, the exact quantity of their KL-divergence is intractable [Dieng et al., 2017], such as two Gaussian Mixture Models [Goldberger et al., 2003].

We denote KL-divergence as D_{KL}. When KL is hard to compute, Monte Carlo (MC) method is a common approach for approximating [Chen et al., 2008]. To briefly illustrate, we sample N results in order to approximate $D_{\mathrm{KL}}(Q\|P)$ from distribution Q as $\{\mathbf{z}_i\}_{i=1}^{N}$, then to compute $\frac{1}{N}\sum_{i=1}^{N}(\log q(\mathbf{z}_i) - \log p(\mathbf{z}_i))$ as the estimated D_{KL} [Hershey and Olsen, 2007]. However, such estimations can lead to severe noisy results. To illustrate: by definition, D_{KL} is non-negative [MacKay and Mac Kay, 2003], whereas MC estimation can violate the non-negativity of D_{KL} [Nielsen, 2020]. Furthermore, the negative estimated D_{KL} is also irrational since D_{KL} stands for the difference between two distributions, and it is unfounded to say a difference is negative.

To fix the negativity of MC estimation of D_{KL}, we can calculate the absolute value of difference $(\log q(\mathbf{z}_i) - \log p(\mathbf{z}_i))$ *i.e.*, to utilize $|\log q(\mathbf{z}_i) - \log p(\mathbf{z}_i)|$ instead of directly using the log difference. Consequently, MC estimation becomes non-negative due to the non-negativity of the absolute values. We also show in the thesis that the new absolute form is a semi-metric, supporting its discriminability for two distributions. Another approach is to utilize the variance reduction technique to generate the *regression estimator* of KL-divergence used in Monte Carlo estimation. By choosing specific coefficients, we can guarantee the non-negativity of estimation. We give detailed derivation of these two novel estimators, as well as the corresponding new divergences and lower-bound functions used in optimization.

Apart from theoretic advantages of D_{AKL} and D_{TKL} over D_{KL}, in practice, we also demonstrate that one can replace KL-divergence within the objective function of variational autoencoders (VAEs) [Kingma and Welling, 2013] with these two divergence. This introduces new lower bounds of evidence for VAE's evidence of lower bound observation (ELBO). We conduct experiments with two VAEs using different divergence, where it has been shown that the analytical forms of KL-divergence are hard to compute. Our experimental results indicate that MC sampling with new diver-

gence is more stable compared to the vanilla model. Moreover, the generated results using D_{AKL} and D_{TKL} are more realistic than employing D_{KL}, supported by quantitative lower reconstruction error and qualitative more realistic looking of generated sampled images.

In sum, our contributions are three-fold:

1. We propose two novel KL estimators which guarantee the non-negativity, and bring forward new divergence based on these estimators. They are new measures for computing the difference between two distributions. One of them satisfies the axioms of semi-metrics, and the other brings new insights on reducing the variance of estimators and can be connected to the family of f-divergence.

2. Using D_{AKL} and D_{TKL} respectively, we derive new lower bounds of ELBO as new objective functions of VAEs.

3. We show via experimental results that when using our proposed measurement, VAEs have more stable training and produce more realistic results.

Chapter 2

Theoretic Background

2.1 Probabilistic Modeling

Probabilistic modeling is a fascinating scientific field with a solid theory that glue multiple very different branches of mathematics. Among them, the fundamental composition is the probability theory. The probability theory is a basic mathematical framework that formulates the uncertainty of particular process. It also provides the foundation for analyzing the hypothesis and uncertain statements. In the domain of machine learning and artificial intelligence, the probability theory provides intuitions to design complex algorithms and systems. By modeling real-world problems with probability, we can even gain connections to philosophy, particularly for the question of causality. In this section, we briefly recap the basic Bayes' theorem, and then introduce the intuition of what exactly the probabilistic modeling is.

2.1.1 Intuition

The most common definition of a mathematical model for inferring the real-world problem is like:

$$y = Wx, \tag{2.1}$$

where y is some sort of random variable that we want to infer with the input x, and W is the learnt variable defining the weight of transformation. As most real-world problems are consistently stochastic, we usually formulate a particular real-world phenomena with probability distribution

$$p(x, y). \tag{2.2}$$

The probability theory can be viewed as an extension to logical reasoning. The above $p(x, y)$ describe a joint distribution of two random variables. The distributions $p(x)$ and $p(y)$ are the corresponding marginal distributions. The $p(x|y)$ or $p(y|x)$ is the conditional distribution of the former variables given the latter ones. Given these definitions, the Bayes' theorem is defined as:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} \tag{2.3}$$

Empirically, we give names to these terms:

- $p(y) \rightarrow$ Prior: belief before making a particular observation. It encapsulate the subjective prior knowledge of latent variable y before we observe the input data. The prior is usually pre-defined manually but it is important to make sure that they have non-zero density on all y .
- $p(y|x) \rightarrow$ Posterior: belief after making the observation. Posterior is the prior for the next observation. It quantifies the interest in Bayesian since it elaborates what we know about y by observing x.
- $p(x|y)$ or $L(y) \rightarrow$ Likelihood: Bridging the prior and posterior, quantifying how random variables e.g. x and y are correlated. It is a typical function of y rather than a distribution of y. Alternatively, we can call it the probability of x given y.
- $p(x) \rightarrow$ Evidence: the variable we want to observe/infer, to draw conclusions about y.

This equation follows the product rule, in the case of multivariate probability distribution (assuming $\{x\} = [x_n, x_{n-1}, \cdots, x_2, x_1]^T$). We can rewrite the joint probability distribution as:

$$p(x_1, x_2, \cdots, x_n) = p(x_1)p(x_2|x_1)\cdots p(x_n|x_{n-1}, \cdots, x_2, x_1). \tag{2.4}$$

The equation also follows the sum rule. Similarly to above setting, we can rewrite the marginal probability distribution as:

$$p(x_i) = \int_{\{x\}_{\setminus i}} p(x_1, x_2, \cdots, x_n) d\{x\}_{\setminus i} \tag{2.5}$$

The Bayes' theorem allows one to invert the relationship between multiple random variables given the likelihood. It is a fundamental block of Bayesian statistics, contributing to the constitution of statistic machine learning. In the following sections, we will frequently reuse the notion of Bayesian statistics.

2.1.2 Probabilistic Models

Typically, probabilistic models are utilized to describe systems with the use of probability distribution. In the perspective of machine learning, probabilistic models are employed to approximate the true data distribution by a set of samples as input $D = \{x_1, x_2, \cdots, x_n\}$ drawn from data distribution $p_{data}(x)$. What we are interested in is to approximate the true distribution using a model q_θ (parametric family) parameterized by θ, such that $q_{\theta^*}(x) \rightarrow p_{data}(x)$.

A common approach to achieve this in the domain of machine learning is Maximum Likelihood Estimation (MLE), which generates the optimal estimate using the following estimator:

$$\theta^* = \arg_\theta \max q_\theta(D). \tag{2.6}$$

Assuming the samples D are drawn to be independent and identically distributed, we can rewrite the above equation as:

$$\theta^* = \arg_\theta \max \sum_n \{\log p_\theta(x_i)\} = \arg_\theta \max \mathbb{E}_{x\sim D}\{\log p_\theta(x)\}. \tag{2.7}$$

The variational inference in statistics provides a new insight toward MLE, which transfer the maximization to KL-divergence minimization between empirical distribution p^*_{data}, sampled from true data distribution p_{data} and the predefined parameterized distribution q_θ. Intuitively, the KL-divergence is a measure of difference between two distributions. The definition of KL-divergence is as follows:

$$D_{\mathrm{KL}}(p(x)\|q(x)) = \mathbb{E}_{x\sim p}[\log \frac{p(x)}{q(x)}] = \mathbb{E}_{x\sim p^*_{data}}[\log \frac{p^*_{data}}{q_\theta(x)}]. \tag{2.8}$$

In Eq.(2.8), the p^*_{data} is normally a constant for comparison study. Thus, the optimization target is the parameter θ in q_θ. By minimizing the D_{KL}, we can achieve the same goal to Eq.(2.7).

Inspired by the fact that artificial neural networks are universal functional approximators, in the perspective of deep learning, the parametric distribution families such as q_θ is possible to be represented by neural networks (NNs). Thus, the problem of optimizing the parameter θ can be viewed as optimizing the parameter of neural networks. In particular, the process of achieving approximated solution in NN is usually done by gradient descent with backpropagation (BP) algorithm [Rumelhart et al., 1986].

2.2 Probabilistic Graphical Models

2.2.1 Directed Graphical Models

Directed graphical models (a.k.a. Bayesian networks) are a family of probability distributions that admit a compact parametrization that can be naturally described using a directed graph. The above distributions can be naturally interpreted as directed acyclic graph (DAG).

Consider the similar scene mentioned above, suppose we have a arbitrary joint distribution:

$$p(a, b, c, d).$$

By successive application of the product rule, we have:

$$\begin{aligned} p(a, b, c, d) &= p(a|b, c, d)p(b, c, d) \\ &= p(a|b, c, d)p(b|c, d)p(c, d) \\ &= p(a|b, c, d)p(b|c, d)p(c|d)p(d). \end{aligned} \tag{2.9}$$

This decomposition holds for any choice of the joint distribution. We now represent the above described conditional dependencies as follows: we introduce a graph where

the nodes are random variables and associate each node with the corresponding conditional distribution. The conditional distributions are appended with arrowed link specifying the correspondence of random variables on which the distribution is conditioned.

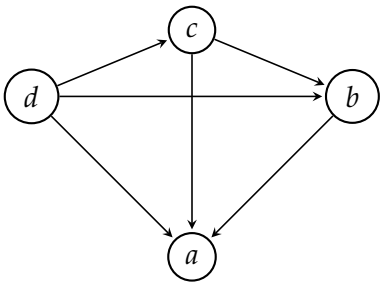


Figure 2.1: A sample directed graphical model with four variables, visually specifying how random variables depend on each other in Eq.(2.9)

Generally, we say that a probability p factorizes over a directed graph G if it can be decomposed into a product of factors, as specified by G. We can interpret the directed graphs in terms of how the data was generated (in a generating perspective, which is important in latter sections). In the above example, to get the distribution of random variable a, we may first sample d, then c is sampled given d. We can then sample b jointly by c and d. Finally, a is generated based on the joint distribution of b, c, and d.

To formally define, a directed graphical model is written as $G = (V, E)$, which is formed by a collection of vertices $V = \{1, 2, ..., m\}$ and a collection of edges $E \in V \times V$. Each edge consists of a pair of vertices $s, t \in E$, in which case we write $(s \to t)$ to indicate the direction. Given such DAG, for each vertex v and its parent set $\pi(v)$, we let $p(x_v|x_{\pi(v)})$ denote a non-negative function over the variables $(x_v, x_{\pi(v)})$. Also, it is normalized such that $\int p(x_v|x_{\pi(v)}) = 1$.

2.3 Inference

Given a joint probability distribution defined by graphical model, we focus on following computational inference tasks:

1. **Marginal inference**: The probability of a specific variable in the model after summing everything out.

$$p(y = k) = \sum_{x_1} \sum_{x_2} \cdots \sum_{x_n} p(y = k, x_1, x_2, \ldots, x_n).$$

2. **Maximum a posteriori inference**: The most possible assignment to the variables in the model.

$$\max_{x_1,\ldots,x_n} p(y = k, x_1, \ldots, x_n)$$

However, doing the above inference problem is challenging in practice. In most cases, performing exact inference is NP-hard since statistically, we need to sum/iterate over all configurations of random variables other than our target in the joint distribution. It is apparent that a brute force approach will rapidly become intractable. For the discrete case, computing a mode entails solving an integer programming problem over an exponential number of configurations. For continuous random vectors, the problems are typically harder, since they require computing a large number of integrals [Wainwright and Jordan, 2008].

In fact, many interesting classes of models may not admit exact polynomial-time solutions. Moreover, the complexity of graph structure may also leads to intractability. Although such difficulties exists, it is still possible to achieve useful result using approximate inference methods. Approximation algorithms like variational inference is the most widely used technique, and people have already attempted concatenating graphical models with neural networks by using algorithms like mean-field inference [Zheng et al., 2015b] and achieved satisfactory results in practice. In the following sections, we will introduce some prevalent approximate inference algorithms that is related to our topic.

2.3.1 Sampling method

Monte Carlo

Historically speaking, Sampling-based inference has been the most prevalent way of performing approximate inference. Such methods can be used to perform both marginal and maximum a posteriori inference queries. In addition, they can compute many interesting quantities. For instance, the expectations of random variables distributed according to the given probabilistic model. Usually, we perform sampling for tasks in the following form:

$$\mathbb{E}_{x\sim p}[f(x)] = \sum_{x} f(x)p(x). \tag{2.10}$$

The above summation/integral is most likely to be impossible to perform analytically. We alternatively approximate it with large number of samples from a given distribution p. Such algorithm is referred to as Monte Carlo (MC). In general the MC generate the approximated result by:

$$\mathbb{E}_{x\sim p}[f(x)] \approx \frac{1}{N}\sum_{n=1}^{N} f(x^n) = I_N, \tag{2.11}$$

where $\{x^1, ..., x^n\}$ are samples drawn from the distribution p. I_N is the estimation value based on N samples, also called the estimator. To evaluate an estimator, we have two basic principles for an estimator: bias and variance, generally a good estimator should be unbiased and have low variance.

We can show the expectation of the estimation:

$$\mathbb{E}_{x^1,...,x^n\sim p}[I_N] = \mathbb{E}_{x\sim p}[f(x)]. \tag{2.12}$$

Similarly the variance of the estimation:

$$\mathrm{Var}_{x^1,\ldots,x^n\sim p}[I_N] = \frac{1}{N}\mathrm{Var}_{x\sim p}[f(x)]. \tag{2.13}$$

The Eq.(2.12) demonstrates that the I_N estimated by MC is unbiased since the expectation will approximate $\mathbb{E}_{x\sim p}[f(x)]$ as $N \to \infty$. Also, in Eq.(2.13), the variance could be arbitrarily small if we have **enough** samples, approximating zero as $N \to \infty$.

Importance Sampling

Importance sampling is a special case of MC integration. The main idea of importance sampling is: suppose we want to approximate $\mathbb{E}_{x\sim p}[f(x)]$, we can rewrite the integral as:

$$\begin{aligned}
\mathbb{E}_{x\sim p}[f(x)] &= \sum_x f(x)p(x) \\
&= \sum_x f(x)\frac{p(x)}{q(x)}q(x) \\
&= \mathbb{E}_{x\sim q}[f(x)g(x)] \\
&\approx \frac{1}{N}\sum_{n=1}^{N} f(x^n)g(x^n),
\end{aligned} \tag{2.14}$$

where $g(x) = \frac{p(x)}{q(x)}q(x)$, and the samples x^n are draw from a predefined distribution q. The expected value of this MC approximation is identical to the original integral, so the estimator is unbiased. We can also evaluate the variance of this estimator by:

$$\mathrm{Var}_{x\sim q}[f(x)g(x)] = \mathbb{E}_{x\sim q}[f^2(x)g^2(x)] - \mathbb{E}_{x\sim q}[f(x)g(x)]^2. \tag{2.15}$$

The value of variance is greater or equal to zero, and only equal to zero when $q^\star(x) = \frac{|f(x)|p(x)}{\int |f(x)|p(x)dx}$. However, we cannot set directly $q = q^\star$ since sampling from $q^\star$ is generally NP-hard.

At this stage, we bring up another important issue. Consider a simple conditional probability:

$$p(X|Y) = \frac{p(X,Y)}{p(Y)}. \tag{2.16}$$

Using the vanilla importance sampling, we approximate the numerator as follows:

$$\begin{aligned}
p(X,Y) &= \sum_x p(x,y)\mathbb{I}(x) \\
&= \sum_z g_y(x)q(x)\mathbb{I}(x) \\
&= \mathbb{E}_{x\sim q}[g_y(x)\mathbb{I}(x)] \\
&\approx \frac{1}{N}\sum_{n=1}^{N} g_y(x^n)\mathbb{I}(x^n),
\end{aligned} \tag{2.17}$$

where $\mathbb{I}(x)$ is control function. If x is consistent with $X_i = x_i$, then the value of $\mathbb{I}$ is 1, otherwise 0. The same as the denominator:

$$p(Y) \approx \frac{1}{N} \sum_{n=1}^{N} g_y(x^n). \tag{2.18}$$

However, if we estimate the numerator and the denominator with different samples of $x \sim q$, then there will be incorrect estimation of the compounded fraction. We can fix this by using the same set of samples $\{x^1, x^2, ..., x^N\} \sim q$ for both the numerator and the denominator, while the problem is such remedy will make the estimator biased:

$$\mathbb{E}_{x\sim q}[p(X|Y)] = \mathbb{E}_{x\sim q}[\frac{\frac{1}{N}\sum_{n=1}^{N} g_y(x^n)\mathbb{I}(x^n)}{\frac{1}{N}\sum_{n=1}^{N} g_y(x^n)}] = \mathbb{E}_{x\sim q}[\mathbb{I}(x)] \neq p(X|Y). \tag{2.19}$$

We can still use such an estimator since it converge to the unbiased value as $N \to \infty$. From the above introductory background toward MC, we can see there is still intractability in many of their operations. although challenges exist, the above approaches provided us at least some direction, and these insights will inspire the development of better estimators for specific tasks.

2.3.2 Variational Method

The variational methods have been gradually becoming popular and more broadly applied to many inference tasks. The main concept of variational methods is to convert inference to an optimization problem. There are many advantages of using variational compared to sampling. For instance, the variational inference has better scalability and more robust to gradient optimization in deep learning. It is also parallelizable, which can be easily accelerated by multi-core GPU. Moreover, we can know exactly if the optimization is converged, and we have analytical formula on the boundary of accuracy.

Generally, assume we are given an intractable distribution p_θ, the variational process will try to solve the optimization problem by using a class of pre-defined, tractable distribution Q, where we find a $q_\phi \in Q$ that is approximately identical to p_θ. We will then query q_ϕ instead of p_θ to get the approximation solution. In the following subsections, we will bring up the KL-divergence, which is the key of variational inference, as well as the variational bound.

Kullback-Leibler Divergence

The Kullback-Leibler divergence a.k.a the relative entropy is an important metric in the context of information theory. In order to explain the pattern of KL-divergence, we start the introduction with the concept of cross-entropy, and bring forward how they are correlated. First, consider the well-informed information entropy of distribution

p:

$$H(P) = -\mathbb{E}_p[\log p] = -\int_{x\sim p} p(x)\log p(x)dx, \tag{2.20}$$

where the entropy is describing the average level of "information", "surprise", or "uncertainty" inherent in the variable's possible outcomes [Wikipedia contributors, 2021a]. This brings us to the definition of cross-entropy:

$$H(P,Q) = -\mathbb{E}_p[\log q] = -\int_{x\sim p} p(x)\log q(x)dx, \tag{2.21}$$

where q, p and two distributions and we only change the estimating probability of Eq.(2.20) to q from p. Intuitively, the cross-entropy is a measure of the difference between two probability distributions for a given random variable. On the ground of information theory, the cross-entropy calculates the expected message transmitted on an average event when a wrong distribution q is assumed while the data actually follows p. The KL-divergence therefore born within the definition of above two entropy, which can be written as:

$$\begin{aligned} D_{\mathrm{KL}}(P\|Q) &= H(P,Q) - H(P) \\ &= -\int_{x\sim p} p(x)\log q(x)dx + \int_{x\sim p} p(x)\log p(x)dx \\ &= \int_{x\sim p} p(x)\log\frac{p(x)}{q(x)}dx \\ &= \mathbb{E}_{p(x)}[\log\frac{p(x)}{q(x)}]. \end{aligned} \tag{2.22}$$

It can be directed interpreted that the KL-divergence is the extra information (information gain) if P would be used instead of Q. In other words, it is the amount of information loss when Q is used to approximate P [Wikipedia contributors, 2021b]. Therefore, KL is always non-negative. Usually, people tend to say that KL-divergence is describing the distance between p and q. However, one thing to note is that KL-divergence is not symmetric (*i.e.*, $D_{\mathrm{KL}}(P\|Q) \neq D_{\mathrm{KL}}(Q\|P)$). Thus, it is not an universal distance unit but an information metric.

Variational Bound

Assume we are given a data distribution p_θ parameterized by unknown parameter θ. We want to approximate p_θ using a tractable distribution q_ϕ parameterized by ϕ. A straightforward idea for such approximation is to optimize the KL-divergence between two distribution, i.e. to minimize $D_{\mathrm{KL}}(q_\phi(x)\|p_\theta(x))$. However, directly evaluating $D_{\mathrm{KL}}(q_\phi(x)\|p_\theta(x))$ is not possible since we have to evaluate the unknown p_θ. Instead, we do the following transformation:

Firstly we have the logarithmic representation of distribution p_θ (for computa-

tional convenience) which we want to approximate:

$$\log p_\theta(x). \tag{2.23}$$

We have the following series of identical deformation:

$$\begin{aligned}
\log p_\theta(x) &= \log p_\theta(x,z) - \log p_\theta(z|x) && \text{(Bayes' theorem)} \\
&= \log\left(\frac{p_\theta(x,z)}{q_\phi(z|x)}\right) - \log\left(\frac{p_\theta(x|z)}{q_\phi(z|x)}\right) \\
&= \log p_\theta(x,z) - \log q_\phi(z|x) - \log\left(\frac{p_\theta(z|x)}{q_\phi(z|x)}\right).
\end{aligned} \tag{2.24}$$

To calculate the expectation of left hand side of Eq.(2.24), we assume $z \sim q_\phi(z|x)$:

$$\mathbb{E}_{z\sim q_\phi}[\log p_\theta(x)] = \int_{z\sim q_\phi} q_\phi(z|x)\log p_\theta(x)dz = \log p_\theta(x)\int_{z\sim q_\phi} q_\phi(z|x)dz = \log p_\theta(x). \tag{2.25}$$

To calculate the expectation of right hand side of Eq.(2.24), we assume $z \sim q_\phi(z|x)$:

$$\begin{aligned}
&\mathbb{E}_{z\sim q_\phi}[\log p_\theta(x,z) - \log q_\phi(z|x) - \log\left(\frac{p_\theta(z|x)}{q_\phi(z|x)}\right)] \\
&= \int_{z\sim q_\phi} \log p_\theta(x,z) q_\phi(z|x)dz - \int_{z\sim q_\phi} \log q_\phi(z|x) q_\phi(z|x)dz - \int_{z\sim q_\phi} \log\left(\frac{p_\theta(z|x)}{q_\phi(z|x)}\right) q_\phi(z|x)dz \\
&= \underbrace{\int_{z\sim q_\phi} \log p_\theta(x,z) q_\phi(z|x)dz - \int_{z\sim q_\phi} \log q_\phi(z|x0) q_\phi(z|x)dz}_{\text{Evidence Lower-bound (ELBO)}} + \underbrace{\int_{z\sim q_\phi} \log\left(\frac{q_\phi(z|x)}{p_\theta(z|x)}\right) q_\phi(z|x)dz}_{D_{\mathrm{KL}}(q_\phi(z|x)\|p_\theta(z|x))}.
\end{aligned} \tag{2.26}$$

The left side of Eq.(2.26) is named as Evidence Lower-bound (ELBO) and the right side is the KL-divergence between $q_\phi(z|x)$ and $p_\theta(z|x)$. We can combine Eq.(2.25) and Eq.(2.26) and get the expression of ELBO (also denoted as $\mathcal{L}$):

$$ELBO(\theta,\phi;x) = -D_{\mathrm{KL}}(q_\phi(z|x)\|p_\theta(z|x)) + \log p_\theta(x). \tag{2.27}$$

Since KL-divergence is non-negative, we can rewrite the Eq.(2.27) as:

$$\log p_\theta(x) \geq ELBO(\theta,\phi;x) = \mathbb{E}_{z\sim q_\phi}[-\log q_\phi(z|x) + \log p_\theta(z|x)], \tag{2.28}$$

which can be also written as [Kingma and Welling, 2013]:

$$ELBO(\theta,\phi;x) = -D_{\mathrm{KL}}(q_\phi(z|x)\|p_\theta(z)) + \mathbb{E}_{z\sim q_\phi(z|x)}[p_\theta(x|z)]. \tag{2.29}$$

The Eq.(2.28) and Eq.(2.29) are derived from the utilization of Jensen's inequality [Jensen, 1906], and we will give a detailed proof in the next chapter. Now we observe that the KL-divergence in Eq.(2.29) can be evaluated since $q_\phi(z|x)$ is a tractable pre-defined distribution (ususally a Gaussian) and $p_\theta(z)$ is a prior, as a prior is the

belief before making a particular observation. We can set arbitrary value to the belief initially thus this term is also tractable. Since the KL now is tractable, we can minimize the KL to get the approximated p_θ using q_ϕ.

In the deep learning fashion, we prefer using gradient descent to optimize the parameter θ and ϕ (e.g. SGD or Adagrad [Kingma and Ba, 2017]). The Monte Carlo gradient descent can be used to approximate an analytical KL and then we calculate the gradient. Alternatively, there is also the reparameterization trick which reparameterize the q_ϕ to a normal multivariate Gaussian which makes the KL analytical.

2.4 Probabilistic Learning

Given the dataset, we are interested in fitting a model that can make prediction on various tasks we care about. In the probabilistic setting, assume we have the dataset D consisting k samples which are sampled from a distribution p, and suppose we have the family of models M, the learning task is to return a model in M that prescribe the distribution p using the knowledge of our sampled k data. However, we cannot get the precise answer since the sampled data is limit so we can only have the approximation of true distribution based on the limited knowledge. Still, we want to select the optimal approximation toward the distribution p. As it is described above, a directed graphical model contains two basic components, respectively the graph structure and the parameter induced by the graph. In the setting of this thesis, we focus more on the parameter learning.

2.4.1 Maximum Likelihood

Recapping the directed models, if we want to answer any probabilistic inference query, we will need the information of full distribution, e.g. Marginal inference, where we need to have the joint distribution, so that we can sum out all subordinate variables. In this setting, we can view the learning process as density estimation, where we want the model to learn p that approximate the original data distribution $p^\star$. We again use the KL-divergence introduced in variational inference above:

$$D_{\mathrm{KL}}(p^\star(x|\theta)\|p(x|\phi)) = \sum_{x\sim p^\star} p^\star(x|\theta)\log\frac{p^\star(x|\theta)}{p(x|\phi)} = -\mathbb{E}_{x\sim p^\star}[\log p(x|\phi)] - H(p^\star(x|\theta)), \tag{2.30}$$

where $p(x|\phi)$ and $p^\star(x|\theta)$ are the same to p_ϕ and $p^\star_\theta$. We minimize the KL-divergence, which is also identical to maximizing the former term of RHS $\mathbb{E}_{x\sim p^\star}[\log p(x|\phi)]$. Under such interpretation, the learning task is about choosing the best model q that induces the most similar expected log-likelihood to the empirical result, where the empirical result can be represented by Monte Carlo estimation. Thus, overall, we can define such maximum likelihood learning as:

$$\arg\max_{\phi\in M}\frac{1}{|D|}\sum_{x\in D}\log p(x|\phi) \tag{2.31}$$

2.4.2 Latent Variable

In the previous section, when we are introducing variational method, we bring up variable z as an auxiliary variable. There is a former name for it, which is *latent variable*. Using a latent variable to help the learning of actual data distribution is extremely useful in many cases. For instance, when some data in the dataset is unobserved or implicit, the learning of latent variable will help us extract useful information from the implicit representation and reconstruct the missing data. Moreover, it enables us to leverage the prior knowledge when defining a model.

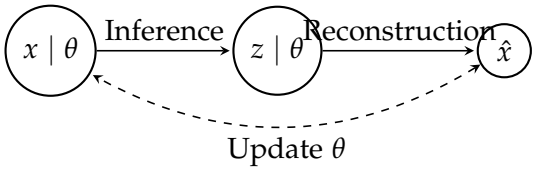


Figure 2.2: An example graphical notation of latent variable model, powered by directed graphical model. x is the observed data distribution. z is the imposed low-dimensional structure latent factor. θ is the parameter that dominates the variables. $\hat{x}$ is the reconstructed data in generative perspective $p(\hat{x}) = \int_z p(x|z)p(z)$

.

To formally define, consider a distribution p parameterized by θ over two sets of variables x and z: $p_\theta(x, z)$, the x are observed variable from the dataset and z are latent unobserved variables. Our goal is to fit the marginal distribution p over observed data. Hence, similar to maximum likelihood learning, we apply KL-divergence by the same argument and optimize the likelihood:

$$\int_{x \in D} \log p_\theta(x) dx = \int_{x \in D} \log \left(\int_z p_\theta(x|z) p_\theta(z) dz \right) dx. \tag{2.32}$$

However, optimizing Eq.(2.32) is difficult since its impossible to divide $p_\theta(x)$ into the product of independent factors because of the participation of z. Moreover, the optimization objective is no longer a convex function (exponential family distribution $p_\theta(x)$) but the conditional density function $p_\theta(x|z)$ by weight $p_\theta(z)$. One specialized algorithm for learning in latent variable model is the Expectation Maximization (EM) algorithm [Dempster et al., 1977]. Another prevalent method for optimizing non-convex objective function is to use neural networks with gradient descent.

2.5 Summary

In this chapter we introduced the basic concepts of the probabilistic model. We describe in detail the notation of graphical model, especially the commonly used directed model. Utilizing the representation of directed graphical model, we further bring up the relationship between inference and learning in the context of probabilistic model. The most important of which are variational inference and Monte Carlo

estimation. Two class of probabilistic operations that will be frequently used in the next chapter – when we are optimizing the neural network. In the next chapter, we will discuss in detail about how to cast latent variable model as generative model, and decompound the sophisticated mechanisms of variational auto-encoders.

Chapter 3

Framework

3.1 Deep Generative Models

In the previous chapter, we introduced latent variable model:

$$p(x, z) = p(x|z)p(z), \tag{3.1}$$

where $x \in D$ is the observed data and z is the latent variable. To give a concrete example, consider the MNIST dataset [LeCun and Cortes, 2010], the setting of x in our scenario refers to the handwritten digit image. z is the latent representation of data which is not seen during the training. Our goal is to encode z with meaningful representation. For instance, one dimension of z can encode the grayscale of image or the brush size of the handwriting, such that z can reconstruct meaningful image after decoding. The random process of forward passing from latent to data distribution

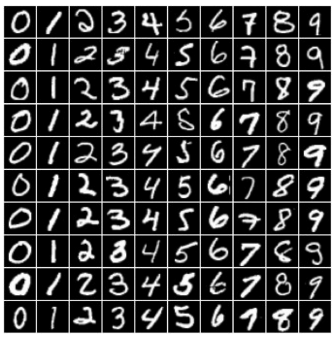


Figure 3.1: Example of the ten handwritten digit images in the MNIST dataset

in a directed graphical model is called generating. A model that functions the above process is called generative model. In deep learning, the data x or the latent z are usually one layer of the neural network. We are also interested in models with multiple layers:

$$p(x|z_1)p(z_1|z_2)\cdots p(z_{n-1}|z_n)p(z_n), \tag{3.2}$$

which are called deep generative model. Suppose now we have $x = \{x^{(i)}\}_{i=1}^{N}$ samples where $x \in D$ are i.i.d. samples from the data distribution. We want the generative model to learn the implicit pattern of x and generative new $\hat{x}$ or reconstruct the

missing part in x. To achieve this, we are interested in the following tasks:

- Learn the parameter of data distribution p.
- Calculate the posterior over z given x, i.e., $p_\theta(z|x)$.
- Calculate the likelihood given z. i.e. $p_\theta(x|z)$.

As θ is an unknown parameter and such distribution family could be extremely complex, computing the posterior $p_\theta(z|x)$ is intractable in practice. However, recall variational inference, it is possible that we define a parameterized tractable q_ϕ and optimize the Evidence lower-bound to approximate p_θ with q_ϕ. If the new distribution family with parameter ϕ is highly analogous to p_θ, it is then possible to operate inference tasks efficiently on the analytical q_ϕ instead of p_θ.

3.2 The Variational Bound

3.2.1 Reformulation of Evidence Lower-bound

In last chapter, our derivation of evidence lower bound landed at Eq.(2.27):

$$ELBO(\theta, \phi; x) = -D_{\mathrm{KL}}(q_\phi(z|x) \| p_\theta(z|x)) + \log p_\theta(x).$$

In order to get the form of Eq.(2.28), we do the following reformulation:

$$\begin{aligned}
ELBO(\theta, \phi; x) &= -D_{\mathrm{KL}}(q_\phi(z|x) \| p_\theta(z|x)) + \log p_\theta(x) \\
&= -\mathbb{E}_{z\sim q_\phi(z|x)}[\log q_\phi(z|x) - \log p_\theta(z|x)] + \mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x)] \\
&= -\mathbb{E}_{z\sim q_\phi(z|x)}[\log q_\phi(z|x) - \log \left(\frac{p_\theta(x|z)p_\theta(z)}{p_\theta(x)} \right) - \log p_\theta(x)] \\
&= -\mathbb{E}_{z\sim q_\phi(z|x)}[\log q_\phi(z|x) - \log p_\theta(x|z) - \log p_\theta(z) + \log p_\theta(x) - \log p_\theta(x)] \\
&= -\mathbb{E}_{z\sim q_\phi(z|x)}[\log q_\phi(z|x) - \log p_\theta(x|z) - \log p_\theta(z)] \\
&= \mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - \mathbb{E}_{z\sim q_\phi(z|x)}[\log q_\phi(z|x) - \log p_\theta(z)] \\
&= \underbrace{\mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)]}_{\textit{Reconstruction error}} - D_{\mathrm{KL}}(q_\phi(z|x) \| p_\theta(z)),
\end{aligned} \tag{3.3}$$

which can be also written as:

$$ELBO(\theta, \phi; x) = \mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z) - \log q_\phi(z|x) + \log p_\theta(z)]. \tag{3.4}$$

Note that Eq.(3.3) is identical to Eq.(2.28) which we discuss in Sec.2.3.2. The intractability of Eq.(2.28) occurs on the true posterior density $p_\theta(z|x) = \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(x)}$ so we cannot directly evaluate the ELBO and we cannot perform EM algorithm. After reformulating using algebra, Eq.(3.3) is evaluatable since the $q_\phi(z|x)$ in KL is derived from analytical distribution family (e.g. Gaussian or Bernoulli). Also, ϕ is a known

parameter so it is tractable. The $p_\theta(z)$ in KL is a prior and it can be any distribution so it is tractable. The $\mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)]$, also called reconstruction in practice can be the output of neural network model. Thus, we can optimize this term with cross-entropy loss.

3.2.2 The Naïve Estimator

We would like to optimize the lower bound we derived in Eq.(3.3). one attempt do this by using gradient descent where we draw N samples of latent $\{z^{(i)}\}_{i=1}^N$ from q_ϕ and differentiate through the Monte Carlo estimation of ELBO:

$$ELBO(\theta, \phi; x) \approx \frac{1}{N}\sum_{i=1}^{N}(\log p_\theta(x|z) - \log q_\phi(z|x) + \log p_\theta(z)), \tag{3.5}$$

where the corresponding terms in ELBO:

$$D_{\mathrm{KL}}(q_\phi(z|x)\|p_\theta(z)) \approx \frac{1}{N}\sum_{i=1}^{N}(\log q_\phi(z|x) - \log p_\theta(z)), \tag{3.6}$$

$$\mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] \approx \frac{1}{N}\sum_{i=1}^{N}(\log p_\theta(x|z)). \tag{3.7}$$

This naïve Monte Carlo estimator is unbiased, which means it will approach the lower bound when $N \to \infty$. However, such estimation exhibits **high variance** and is not practical for the KL term since it will generate negative samples. This critical problem will be discussed in detail in the latter sections. Now, we have an alternative practical estimator where we skip the estimation of KL term and **reparameterize** the latent variable $z \sim q_\phi(z|x)$ using a differentiable transformation, such that the KL can be analytically integrated. In the latter section, we will introduce this reparameterization trick.

3.3 Encoder and Decoder

If we look at Eq.(3.3), this is already a reparameterized ELBO function compared to Eq.(2.27) we derived at the very beginning. The reason we use this new derivation is that this will bring some interesting interpretation with respect to deep learning.

Think of x as an observed data sample, for instance, an digit image from MNIST. Rhe expectation of the KL term and the reconstruction term both invoke sampling from $z \sim q_\phi(z|x)$, imaginably. $q_\phi(z|x)$ takes x as input and z as an output, we therefore call q_ϕ the *Encoder*.

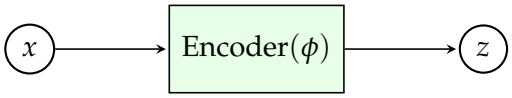


Figure 3.2: Encoder model, part of parameters of MLP is represented as ϕ

The first term in Eq.(3.3) include $\log p_\theta(x|z)$, which is the log likelihood of x given z. One of our optimization object is to maximize this term, mathematically, as log is monotonically increasing. This term is maximized when p_θ assigns high probability, such that imaginably. The $p_\theta(x|z)$ can *reconstruct* a good $\hat{x}$ compared to the origin x based on the input z. For such reason, we call p_θ the *Decoder*.

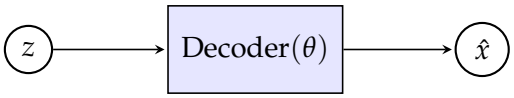


Figure 3.3: Decoder model, part of parameters of MLP is represented as θ

Thus, in this perspective, our optimization goal is to train a $q_\phi(z|x)$ (optimize the parameter ϕ) that maps x into a compact and meaningful latent space z. We then feed z into $p_\theta(x|z)$ to get a reconstructed x. This encoder-decoder architecture is highly analogous to auto-encoder neural networks, which is also the initial design philosophy for variational auto-encoder.

3.4 The Reparameterization

As we discussed earlier, the naïve Monte Carlo gradient estimate exhibits high variance. In this section, we will introduce a low-variance gradient estimator powered by reparametrization trick with Monte Carlo. Consider the condition, we are estimating the gradient using Monte Carlo:

$$\begin{aligned}\nabla_{\theta,\phi} ELBO(\theta,\phi;x) &= \nabla_{\theta,\phi}\mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z) - \log q_\phi(z|x) + \log p_\theta(z)] \\ &= \nabla_{\theta,\phi}\mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - \nabla_{\theta,\phi} D_{\mathrm{KL}}(q_\phi(z|x)\|p_\theta(z)).\end{aligned} \tag{3.8}$$

Notice that we cannot swap the gradient and the expectation. Since the expectation is being taken with respect to the distribution that we are trying to differentiate, this bring high variance when we cannot accurately estimate the ELBO. One thing we can do is to swap the order of gradient and expection for the reconstruction term in Eq.(3.8) $\nabla_{\theta,\phi}\mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] = \mathbb{E}_{z\sim q_\phi(z|x)}[\nabla_\theta \log p_\theta(x|z)]$ as there is no ϕ inside the expectation. At this stage, let's make a brief summary of the existing issues:

- $\nabla_\theta D_{\mathrm{KL}}(\cdot)$ have large variance using naïve estimator.
- $\nabla_\phi D_{\mathrm{KL}}(\cdot)$ have large variance using naïve estimator.
- $\nabla_\phi \mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)]$ have large variance using naïve estimator.
- ($\nabla_\theta \mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)]$ can be computed with low variance by auto differentiation given samples $z \sim q_\phi(z|x)$).

The reparameterization basically does two things:

1. Make KL analytical so we can directly apply auto differentiation and back-propagate.

2. Make back-propagation work for estimating gradient $\nabla_\phi \mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)]$ with low-variance.

Let's first talk about one solution to $D_{\mathrm{KL}}(q_\phi(z|x)\|p_\theta(z))$ instead of direct sampling, in a Gaussian setting, the KL can be analytically integrated. Firstly, we assume the prior p_θ to be a standard univariate Gaussian $N(0, I)$ and the posterior $q_\phi(z|x)$ to be normal univariate Gaussian $N(\mu, \sigma)$, then we have:

$$\begin{aligned}\int_z q_\phi(z|x)\log p_\theta(z)dz &= \int_z N(z;\mu,\sigma^2)\log N(z;0,I)dz \\ &= -\frac{1}{2}N\log 2\pi - \frac{1}{2}\sum_{i=1}^{N}(\mu_i^2+\sigma_i^2),\end{aligned} \tag{3.9}$$

where be the dimensionality of $z \in R^N$ and i is the ith sampled data point. Similarly, we also have:

$$\begin{aligned}\int_z q_\phi(z|x)\log q_\phi(z|x)dz &= \int_z N(z;\mu,\sigma^2)\log N(z;\mu,\sigma^2)dz \\ &= -\frac{1}{2}N\log 2\pi - \frac{1}{2}\sum_{i=1}^{N}(1+\log\sigma_i^2).\end{aligned} \tag{3.10}$$

Thus overall:

$$\begin{aligned}D_{\mathrm{KL}}(q_\phi(z|x)\|p_\theta(z)) &= \int_z(\log q_\phi(z|x) - \log p_\theta(z))q_\phi(z|x)dz \\ &= -\frac{1}{2}N\log 2\pi - \frac{1}{2}\sum_{i=1}^{N}(1+\log\sigma_i^2) + \frac{1}{2}N\log 2\pi + \frac{1}{2}\sum_{i=1}^{N}(\mu_i^2+\sigma_i^2) \\ &= \frac{1}{2}\sum_{i=1}^{N}(\mu_i^2+\sigma_i^2-\log\sigma_i^2-1).\end{aligned} \tag{3.11}$$

Using matrix algebra gives the same result:

$$\begin{aligned}&\int q_\phi(\mathbf{z})(\log q_\phi(\mathbf{z}) - \log p_\theta(\mathbf{z}))d\mathbf{z} \\ &= \int \mathcal{N}(\mathbf{z};\mu,\Sigma)(\log\mathcal{N}(\mathbf{z};\mu,\Sigma) - \log\mathcal{N}(\mathbf{z};0,I))d\mathbf{z} \\ &= \int\left[\frac{1}{2}\log\frac{|\Sigma|}{|I|} - \frac{1}{2}(\mathbf{z})^T(\mathbf{z}) + \frac{1}{2}(\mathbf{z}-\mu)^T\Sigma^{-1}(\mathbf{z}-\mu)\right]\times q(\mathbf{z})d\mathbf{z} \\ &= \frac{1}{2}\log|\Sigma| - \frac{1}{2}tr\left\{\mathbb{E}[\mathbf{z}\mathbf{z}^T]I^{-1}\right\} + \frac{1}{2}\mathbb{E}[(\mathbf{z}-\mu)^T\Sigma^{-1}(\mathbf{z}-\mu)] \\ &= \frac{1}{2}\log|\Sigma| - \frac{1}{2}tr\{I_J\} + \frac{1}{2}\mu^T\Sigma\mu + \frac{1}{2}tr\{\Sigma^{-1}I_J\} \\ &= \frac{1}{2}\left[\log|\Sigma| - J + tr(\Sigma^{-1}I) + \mu^T\Sigma^{-1}\mu\right] \\ &= \frac{1}{2}\sum_{j=1}^{J}(-\log\sigma_j^2 - 1 + \sigma_j^2 + \mu_j^2).\end{aligned}$$

Therefore, Eq.(3.9)∼Eq.(3.11) proves that the KL term can be analytically, such that only the expectation of reconstruction error need to be estimated by sampling. To exhibit a low-variance for calculate the gradient of reconstruction error, we also rely on the reparameterization (to solve the second problem we raised above).

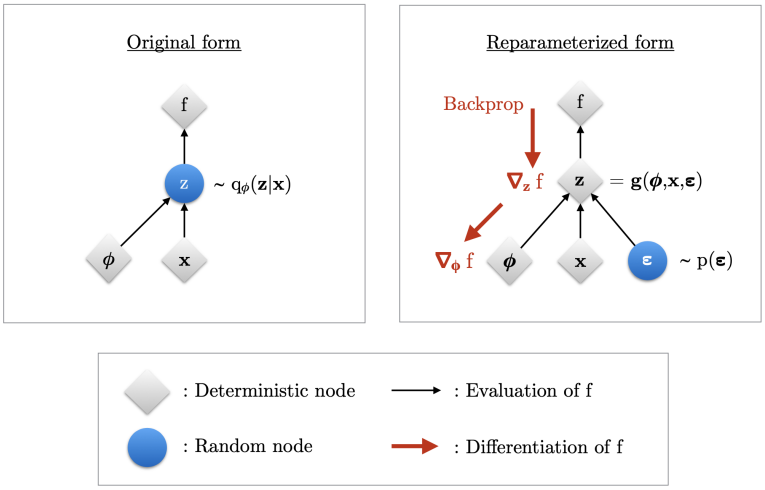


Figure 3.4: Illustration of the reparameterization trick in order to perform backpropagate through random variable. [Kingma's NIPS 2015 workshop slides]

Firstly, we have $z \sim q_\phi(z|x)$ as our sampling space. It is possible to express random variable z as a deterministic variable $z = g_\phi(\epsilon, x)$ and where ϵ is an auxiliary variable with independent marginal $p(\epsilon)$, and $g_\phi(\cdot)$ is some vector-valued function parameterized by ϕ [Kingma and Welling, 2013]. Given such deterministic mapping, the Monte Carlo estimator can be used to compute the gradient as follows:

$$\begin{aligned}
\nabla_\phi ELBO(\theta, \phi; x) &= \nabla_\phi \mathbb{E}_{z\sim q_\phi}[\log p_\theta(x,z) - \log q_\phi(z|x)] \\
&= \nabla_\phi \mathbb{E}_{\epsilon\sim p(\epsilon)}[\log p_\theta(x,z) - \log q_\phi(z|x)] \\
&= \mathbb{E}_{\epsilon\sim p(\epsilon)}[\nabla_\phi\{\log p_\theta(x,z) - \log q_\phi(z|x)\}] \\
&\approx \frac{1}{N}\sum_{i=1}^{N} \nabla_\phi\{\log p_\theta(x,z) - \log q_\phi(z|x)\} \qquad \text{(Monte Carlo)}
\end{aligned} \tag{3.12}$$

where $z = g_\phi(\epsilon, x)$ and $\epsilon \sim p(\epsilon)$. In a multivariate Gaussian setting similar as above, where we assume $q_\phi(z|x) = N(z; \mu, \sigma^2)$, the reparameterization is done as follows:

$$z^{(i)} = \mu^{(i)} + \sigma^{(i)} \odot \epsilon^{(i)} \quad \text{where} \quad \epsilon^{(i)} \sim N(0, I). \tag{3.13}$$

3.5 Auto-Encoding Variational Bayes

Putting together all the content we have discussed before, we get the essence of the variational auto-encoder (VAE), which is the combination of auto-encoder architecture and variational inference. A vanilla VAE can be in the style of multi-layer perceptron

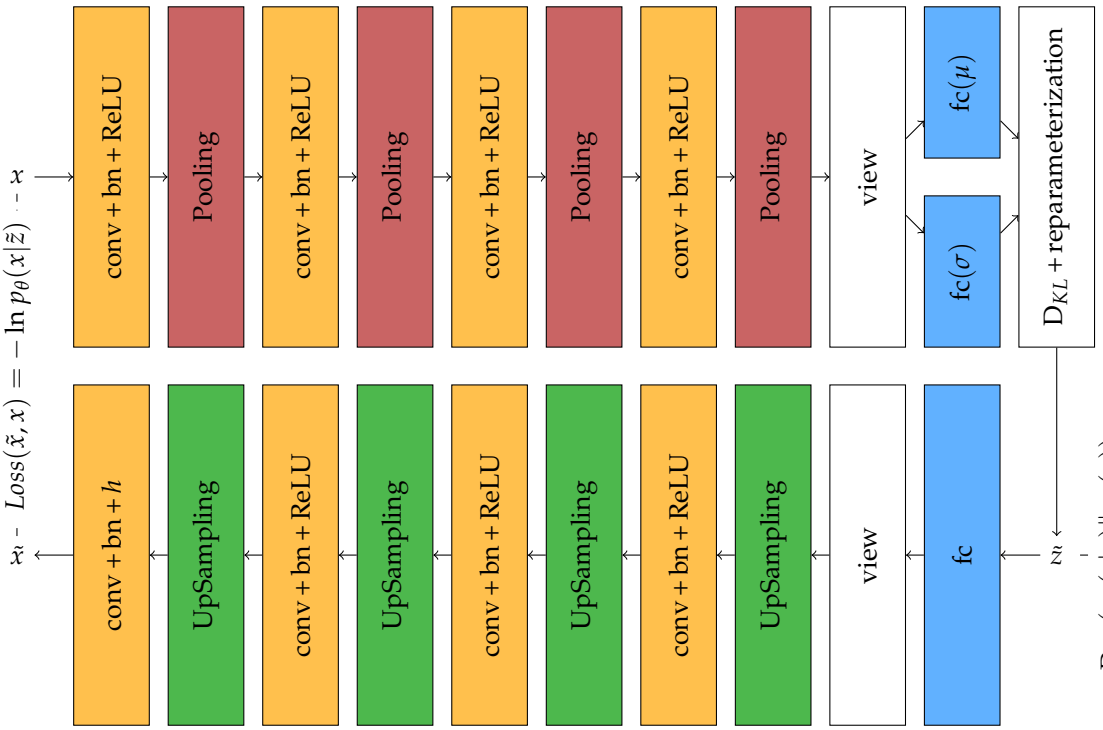


Figure 3.5: A very basic example of variational auto-encoder architecture (Gaussian case) with four convolutional layers as hidden layers in the encoder and decoder. The style of this illustration is inspired by [David Stutz, 2021], we use such architecture to run the experiment in Chapter.5 on the MNIST dataset and the Market-1501. In this case the convolution includes the batch normalization and we utilize ReLU [Nair and Hinton, 2010] as an example activation function to provide non-linearity.

where the STRUCTURE of MLP:

- Encoder: $\log q_\phi(\mathbf{z}|\mathbf{x}) = \log \mathcal{N}(\mathbf{z}; \mu, \Sigma)$
- $\mu = W_1 z + b_1$
- $\log \Sigma = W_2 z + b_2$
- $\mathbf{z} = Activation(W_3 \mathbf{x} + b_3)$

Part of ϕ is used as an encoder. In the decoder where the parameter is θ, $\mathbf{z}$ and $\mathbf{x}$ are swapped. Furthermore, it can also be extended by the convolutional operation. In Fig.3.5, we demonstrate an illustration of convolutional VAE, where we use simple pooling layers in the encoder part and up-sampling in the decoder part. Alternatively, one can also use the transpose convolutional layer (Conv2DTranspose) as the combination of convolution plus up-sampling, which also works perfectly in practice.

What issue does variational auto-encoder addressed compared to vanilla auto-encoder? Let us consider the task of image generation. If we apply the vanilla

auto-encoder (AE), the image x will first pass through an encoder to get the latent representation, and then the latent z pass through the decoder to get the reconstructed image $\tilde{x}$. We train the network such that x and $\tilde{x}$ are as "close" as possible. However, one thing we must be aware of is that it is generally not possible to choose a suitable latent z for AE to generate "meaningful" images. By "meaningful", I mean the reconstruction from latent z weighted by parameter of the decoder generated when z passes through the decoder. In other words, we want $recon(z) \in \{x^{(i)}\}_{i=1}^{N}$.

However, to achieve this is very difficult in practice since we do not put any restrictions/constraints on z. It is entirely up to the neural network to decide how well z is trained, so converged latent may have a value in $[0, 1]$ or in $[1, 100]$, which is quite random and depends on both how the dataset is distributed and the initialization of parameters. More abstractly, it is hard to find the domain of z in the space where it is defined. Therefore, we cannot sample $\tilde{z}$ from that specific domain but only from the whole space $\mathcal{N} \in R^{D(z)}$, as the neural network has no prior knowledge of domains in this high-dimensional space that it has not seen during training, leading to the fact that if the $\tilde{z}$ is sampled from other domains that do not intersect much with the real $dom(z)$ after training. The neural network will have no idea how to reconstruct $\tilde{z}$ and result in the ambiguity of the content in generated images.

The variational manipulation, to some extend, add constraints to the latent variable and force it to be trained in the pattern of predefined distribution, such that we can restrict of $dom(z)$ to follow a gaussian with parameter μ and σ. We can further utilize reparameterization to make $[\mu, \sigma] = [0, 1]$, such that we can sample $\tilde{z}$ from a normal Gaussian to feed in decoder and generate meaningful images. It is apparent that we are free to choose a distribution family other than Gaussian as a sampling distribution. In fact, there is even no need to define a distribution, we can instead assume that it is some complex distribution and approach it with Monte Carlo sampling. Unfortunately, subject to the current computational power and the bias and variance of the Monte Carlo estimator, we actually have difficulty in obtaining an optimal solution during the whole optimization process. However, in the scenario of some tasks', for instance, where the latent is defined in non euclidean manifold *e.g.* Riemannian, there is no generalized Gaussian in high-curvature space so Monte Carlo might be the only choice to compute the KL. Thus, how to enhance our estimator so that our estimates are unbiased and have a relatively low variance has emerged an important issue, which will be addressed in detail in the following sections.

3.6 The Critical Problem

Non-negativity of KL-divergence

The analytical form of KL-divergence between any two probability distribution is non-negative. To start with, it could be easily verified that $\forall \mathbf{x} > 0$: $\mathbf{x} - 1 \geq \log \mathbf{x}$. Further, $f(\mathbf{x}) = \mathbf{x} - 1$ is actually the tangent of $f(\mathbf{x}) = \log \mathbf{x}$. We then have the following proof

of non-negativity:

$$\begin{aligned} D_{\mathrm{KL}}(Q\|P) &= \int_x q(x)\log\frac{q(x)}{p(x)}dx \\ &= -\int_x q(x)\log\frac{p(x)}{q(x)}dx \\ &\geq -\int_x q(x)(\frac{p(x)}{q(x)}-1)dx \quad \text{(log inequality)} \\ &= -\int_x (p(x)-q(x))dx \\ &= \int_x q(x)dx - \int_x p(x)dx \\ &= 0. \end{aligned} \tag{3.14}$$

Approximating KL may generate negative samples: An illustration

To give a concrete example in extreme circumstance, where KL estimation is completely wrong, we firstly assume the dataset is given by $X = \{x^{(i)}\}_{i=1}^N$ and all the data samples are i.i.d. Suppose now we have three samples $\{x^{(1)}, x^{(2)}, x^{(3)}\}$, based on these evidence, we can probably have the following probability generation from distribution q_ϕ: $\{q_\phi(x^{(1)}) = 0.5, q_\phi(x^{(2)}) = 0.1, q_\phi(x^{(3)}) = 0.3\}$, similarly for p_θ: $\{p_\theta(x^{(1)}) = 0.3, p_\theta(x^{(2)}) = 0.3, p_\theta(x^{(3)}) = 0.4\}$, such that the KL-divergence can be estimated as follows:

$$\begin{aligned} D_{\mathrm{KL}}(q_\phi(x)\|p_\theta(x)) &\approx \frac{1}{N}\sum_{i=1}^N \log\frac{q_\phi(x^{(i)})}{p_\theta(x^{(i)}} \\ &= \frac{1}{3}\log\left(\frac{q_\phi(x^{(1)})}{p_\theta(x^{(1)})}\times\frac{q_\phi(x^{(2)})}{p_\theta(x^{(2)})}\times\frac{q_\phi(x^{(3)})}{p_\theta(x^{(3)})}\right) \\ &= \frac{1}{3}\log\left(\frac{0.5}{0.3}\times\frac{0.1}{0.3}\times\frac{0.3}{0.4}\right) = \frac{1}{3}\log\frac{5}{12} < 0. \end{aligned} \tag{3.15}$$

Now the KL estimation contradicts its non-negative property, which is a disastrous consequence for neural network as the negative KL as objective function will mislead the optimization. Compare the hypothetical situation Eq.(3.15) to Eq.(3.14), we can observe that the potential negative problem comes from that $\sum_i q_\phi(x^{(i)}) \neq 1$ and $\sum_i p_\theta(x^{(i)}) \neq 1$. This entails the emergency of the special design of estimator which guarantee the value to be non-negative.

3.7 Summary

In this chapter, we discussed about the concept of deep generative model from a probabilistic perspective. We continue the reformulation of ELBO that we mentioned in previous chapter. We introduced the naïve Monte Carlo estimator and further the potential issue this estimator would bring about e.g. negative estimation of KL

divergence. We also introduced the encoder and decoder architecture in probabilistic setting, and the reparameterization which allows back-propagation for optimizing ELBO in neural network. In the next chapter, we will technically explain two of our proposed estimators which guarantees non-negativity while preserving low-variance of estimation and give working examples toward these two estimators.

Chapter 4

Proposed Estimators

Motivated by the negative MC estimation of KL-divergence using the naïve estimator and the concern mentioned above regarding metrics, in this dissertation, we propose two novel estimators to tackle the non-negativity. We show consistent non-negative MC estimation of our proposed estimators, regardless of sample sizes.

Furthermore, we exhibit the new divergence, one of them is by definition a semi-metric [Wilson, 1931], supporting the rationality of using our estimators to measure the difference between two distributions. In [Abou-Moustafa and Ferrie, 2012], Abou-Moustafa *et al*. have shown divergences of distributions have larger discriminability if the divergences meet the axioms of semi-metrics. Another proposed reformulated estimator can be generalised to the family of f-divergence inspired by the derivation of Bregman divergence [Bauschke and Borwein, 2001] and the idea of control variates [Lemieux, 2017]. This estimator is unbiased and has the lowest variance in most cases.

To start with, we will firstly put forward a reminder of the naïve estimator for KL-divergence, and for the two proposed estimators for KL-divergence. We will in detail describe the motivation, our derivation and proofs of properties.

4.1 Reminder: Naïve Estimator

The KL-divergence that measures the difference between two distributions Q and P, we utilize MC to estimate the value of KL-divergence:

$$D_{\mathrm{KL}}(Q\|P) \approx \frac{1}{N}\sum_{i=1}^{N}\log\frac{q(\mathbf{z}^{(i)})}{p(\mathbf{z}^{(i)})}.$$

Let θ be the parameter needs to be estimated and $\hat{\theta}$ is the estimator of θ. In this case, the naïve estimator $\hat{\theta}$ is defined as

$$\hat{\theta}(\mathbf{z}) = \log\frac{q(\mathbf{z})}{p(\mathbf{z})} \quad \text{or} \quad \hat{\theta}(\mathbf{z}) = \log q(\mathbf{z}) - \log p(\mathbf{z}). \tag{4.1}$$

The naïve estimator is unbiased

The bias describe the distance between the average of the collection of the estimates and the single parameter being estimated. one estimator is unbiased if and only if the bias equal zero. The bias $B(\hat{\theta})$ of estimator $\hat{\theta}$ is by definition:

$$\begin{aligned} B(\hat{\theta}) &= \mathbb{E}(\hat{\theta}) - \theta \\ &= \mathbb{E}[\log \frac{q(\mathbf{x})}{p(\mathbf{x})}] - \mathbb{E}[\log \frac{q(\mathbf{x})}{p(\mathbf{x})}] = 0 \end{aligned} \tag{4.2}$$

The naïve estimator has relatively high variance

Since we can not ensure the relevance of the value between $q(x)$ and $p(x)$, $\hat{\theta} = \log \frac{q(x)}{p(x)}$ can be either positive (if. $\frac{q(x)}{p(x)} > 1$) or negative (if $\frac{q(x)}{p(x)} < 1$). Now we assume half of our samples contribute to negative $\hat{\theta}$ and another half contribute to positive $\hat{\theta}$, then the expectation (mean) of estimator can be zero. Now, we also assume the estimate $\mathbb{E}[\hat{\theta}]$ is zero, according to the definition of variance, we observe that:

$$\begin{aligned} Var(\hat{\theta}) &= \mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}])^2] \\ &= \mathbb{E}[\hat{\theta}^2] = \frac{1}{N}\sum_{i=1}^{N}(\hat{\theta}^{(i)})^2. \end{aligned} \tag{4.3}$$

Note that we previously assumed that $\{\hat{\theta}^{(i)}\}$ is half positive and half negative, the variance can be large due to the power on the estimator. As a consequence, it get rid of negative sign and make the summation larger as the interval between samples increase even it have zero mean.

4.2 Proposal I: Absolute Estimator

4.2.1 Motivation

To fix the negative MC estimation of KL-divergence, one intuitive approach is to calculate the absolute of difference $(\log q(x^{(i)}) - \log p(x^{(i)}))$, *i.e.*, to utilize

$$\left|\log q(x^{(i)}) - \log p(x^{(i)})\right|$$

instead of directly using the log difference. Consequently, MC estimation becomes non-negative due to the non-negativity of absolute. Although the absolute seems similar to the original form of the log difference, it may be debatable whether the new absolute form is an appropriate metric to measure the difference between two distributions.

4.2.2 $\hat{\theta}_{\text{Abs}}$: An Alleviation To High Variance

To fix the negative MC estimation of KL-divergence, one intuitive approach is to calculate the absolute of difference $(\log q(\mathbf{z}^{(i)}) - \log p(\mathbf{z}^{(i)}))$, *i.e.*, to utilize

$$\left|\log q(\mathbf{z}^{(i)}) - \log p(\mathbf{z}^{(i)})\right|$$

instead of directly using the log difference. Consequently, MC estimation becomes non-negative due to the non-negativity of absolute. This yields our first proposed estimator that ensure the non-negativity of MC-based estimation:

$$\hat{\theta}_{\text{Abs}} = \left|\hat{\theta}\right| = \left|\log \frac{q(\mathbf{z})}{p(\mathbf{z})}\right| \tag{4.4}$$

where $\hat{\theta}$ is the naïve estimator. Intuitively, the new estimator is better because each sample can tell how far apart q and p positively. Empirically, $\hat{\theta}_{\text{Abs}}$ indeed has lower variance than $\hat{\theta}$, while unfortunately is a biased estimator. Although the absolute seems similar to the original form of the log difference, it may be debatable whether the new absolute form is an appropriate metric to measure the difference between two distributions.

4.2.3 Case Study

$\hat{\theta}_{\text{Abs}}$ has relatively low variance

In the previous section, we analyzed that the high variance is caused by the half negative samples of $\{\hat{\theta}^{(i)}\}$, in $\hat{\theta}_{\text{Abs}}$. There is no possibility to generate negative sample since the absolute guarantees the non-negativity.

To illustrate, let us assume we have four samples of the estimator $\{\hat{\theta}^{(i)}\} = \{\hat{\theta}^{(1)} = -0.3, \hat{\theta}^{(2)} = -0.1, \hat{\theta}^{(3)} = 0.1, \hat{\theta}^{(4)} = 0.3\}$ which are i.i.d. In the case of naïve estimator $\hat{\theta}$ and $\hat{\theta}_{\text{Abs}}$ respectively, the variance are as follows:

$$\begin{aligned}
Var(\hat{\theta}) &= \mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}])^2] & Var(\hat{\theta}_{\text{Abs}}) &= \mathbb{E}[(\left|\hat{\theta}\right| - \mathbb{E}[\left|\hat{\theta}\right|])^2] \\
&= \mathbb{E}[(\hat{\theta} - \frac{1}{4}\sum_{i}^{4}\hat{\theta}^{(i)})^2] & &= \mathbb{E}[(\left|\hat{\theta}\right| - \frac{1}{4}\sum_{i}^{4}\left|\hat{\theta}^{(i)}\right|)^2] \\
&= \frac{1}{4}\sum_{i=1}^{4}(\hat{\theta}^{(i)})^2 & &= \frac{1}{4}\sum_{i=1}^{4}(\left|\hat{\theta}^{(i)}\right| - 0.2)^2 \\
&= \frac{1}{4} * 0.2 = 0.05 & &= \frac{1}{4} * 0.04 = 0.01 < 0.05 = Var(\hat{\theta}).
\end{aligned}$$

The confidence to do such assumption is because our variational optimization target $\mathbb{E}[\frac{q(x)}{p(x)}]$ should finally converge to 1, *i.e.*, $q_\phi(x) = p_\theta(x)$, such that $\mathbb{E}[\log \frac{q(x)}{p(x)}] = 0$, and thus half of $\hat{\theta}^{(i)}$ are negative and half positive. Similarly, we can assume $\{\hat{\theta}^{(i)}\} = \{\hat{\theta}^{(1)} = -c_2, \hat{\theta}^{(2)} = -c_1, \hat{\theta}^{(3)} = c_1, \hat{\theta}^{(4)} = c_2\}$ i.i.d where $0 \leq c_1 \leq c_2$ are arbitrary positive constants. We can also increase the set size of $\{\hat{\theta}^{(i)}\}$ to $N \geq 4$. Either way, using simple algebra, we can always get the same conclusion that $Var(\hat{\theta}) \leq Var(\hat{\theta}_{\text{Abs}})$.

$\hat{\theta}_{\text{Abs}}$ is biased

By definition, we compute the bias as follows:

$$\begin{aligned} B(\hat{\theta}_{\text{Abs}}) &= \mathbb{E}(\hat{\theta}_{\text{Abs}}) - \hat{\theta} \\ &= \mathbb{E}[\left|\log \frac{q(\mathbf{x})}{p(\mathbf{x})}\right|] - \mathbb{E}[\log \frac{q(\mathbf{x})}{p(\mathbf{x})}] \\ &\approx \sum_{\mathbf{x}} \{\left|\log \frac{q(\mathbf{x})}{p(\mathbf{x})}\right| - \log \frac{q(\mathbf{x})}{p(\mathbf{x})}\}. \end{aligned} \tag{4.5}$$

The former term in the RHS of Eq.(4.5) is guaranteed to be non-negative, where the latter term could be negative depend on the value of q and p, if any $\mathbb{E}[\log \frac{q(x)}{p(x)}]$ has negative samples, *e.g.* if there $\exists i : g(x_i) < p(x_i) \implies \log \frac{q(x_i)}{p(x_i)} < 0$ then $\left|\log \frac{q(x_i)}{p(x_i)}\right| \neq \log \frac{q(x_i)}{p(x_i)}$ and thus $B(\hat{\theta}_{\text{Abs}}) \neq 0$. Therefore, the estimator above is biased which means we will not have the right mean if we plenty of samples are drawn.

4.2.4 D_{AKL}: Absolute-KL-divergence

Since the new distribution metric with estimator $\hat{\theta}_{\text{Abs}}$ is similar to the form of KL-divergence, yet with an additional **absolute** component, we name it as Absolute-KL-divergence, abbreviated as AKL-divergence (D_{AKL}), defined as follows:

$$D_{\text{AKL}}(Q\|P) = \int_{\Omega} \left|\log \frac{q(\mathbf{z})}{p(\mathbf{z})}\right| r(\mathbf{z}) d\mathbf{z} \tag{4.6}$$

where the Ω is the sample space. To avoid asymmetry, we use a new PDF $r(\mathbf{z})$ as the reference distribution on the sample space instead of using $q(\mathbf{z})$. The new PDF $r(\mathbf{z})$ is defined as:

$$r(\mathbf{z}) = \frac{p(\mathbf{z}) + q(\mathbf{z})}{2} \tag{4.7}$$

Despite the similarity between D_{KL} and D_{AKL}, the two are distinct quantities for measuring the difference between two distributions. Therefore, one may have a concern about the plausibility of using D_{AKL} to evaluate distribution differences. In the following, we dispel such a concern by showing D_{AKL} satisfies all the three axioms of a semi-metric Wilson [1931]. Furthermore, as Abou-Moustafa *et al*. have shown in Abou-Moustafa and Ferrie [2012], divergences for measuring the differences between two distributions can have greater discriminability if the divergences meet the semi-metric axioms Ontañón [2020].

D_{AKL} Is A Semi-Metric

The semi-metric axioms consist of four requirements Galas et al. [2017], *i.e.*, for a metric d, it holds: (1) non-negativity, *i.e.*, $d(x; y) \geq 0$; (2) identity of indiscernibles, *i.e.*, $d(x, y) = 0$ iff $x = y$; (3) symmetry, *i.e.*, $d(x, y) = d(y, x)$. The requirements of non-negativity and symmetry are obvious for D_{AKL}.

In the following, we show D_{AKL} also meets conditions of identity of indiscernibles.

Proposition 1. *AKL-Divergence satisfies identity of indiscernible. That is,* $D_{AKL}(Q\|Q) = 0$ *and if* $D_{AKL}(Q\|P) = 0$, *then* $Q = P$.

Proof. It is obvious to observe $D_{\text{AKL}}(Q\|Q) = 0$. We focus demonstrating if $D_{\text{AKL}}(Q\|P) = 0$, then $Q = P$, by contradiction. Suppose there exists Q and P such that $Q \neq P$, yet $D_{\text{AKL}}(Q\|P) = 0$. Then, since $Q \neq P$, $\exists\, \mathbf{z}'$: $Q(\mathbf{z}') \neq P(\mathbf{z}')$, and thus $r(\mathbf{z}') \neq 0$ as well as $\left|\log \frac{q(\mathbf{z}')}{p(\mathbf{z}')}\right| \neq 0$. Since $\forall\ \mathbf{z}$: $r(\mathbf{z})\left|\log \frac{q(\mathbf{z})}{p(\mathbf{z})}\right| \geq 0$, we have $\exists!\ \mathbf{z}''$: $\left|\log \frac{q(\mathbf{z}'')}{p(\mathbf{z}'')}\right| = -\left|\log \frac{q(\mathbf{z}')}{p(\mathbf{z}')}\right|$. Therefore, $\left|\log \frac{q(\mathbf{z}')}{p(\mathbf{z}')}\right|$ cannot be canceled out, and thus D_{AKL} cannot be 0. □

4.3 Proposal II: Tangent Estimator

4.3.1 Motivation

The $\hat{\theta}_{\text{Abs}}$ is a considerably effective remedy for the high variance of naïve MC estimate. Fortunately, the $\hat{\theta}_{\text{Abs}}$ has remarkably low bias since its expectation is an f-divergence [Nielsen, 2020]. However, is there way to construct a estimator that is both unbiased and has low variance? We can find inspiration from the Bregman divergence [Bauschke and Borwein, 2001] where we define a function F for distribution q and p such that the Bregman divergence is the difference between the value of F at p and the value of the first-order Taylor expansion of F around q evaluated at p. This idea is highly analogous to finding a tangent plane (using Taylor expansion on specific point) of a convex function F and looking at the difference between and measure the difference between F and the generated plane.

4.3.2 $\hat{\theta}_{\text{Tan}}$: Between Convex Function and Tangent Plane

The $\hat{\theta}_{\text{Abs}}$ is a considerably effective remedy for the high variance of naïve MC estimate, and fortunately, the $\hat{\theta}_{\text{Abs}}$ has remarkably low bias since its expectation is an f-divergence Nielsen [2020]. However, we still aim to construct a estimator that is both unbiased and has low variance. Inspired by the Bregman divergence Bauschke and Borwein [2001] where we define a function F for distribution q and p such that the Bregman divergence is the difference between the value of F at p and the value of the first-order Taylor expansion of F around q evaluated at p. This idea shares the same spirit with finding a tangent plane (using Taylor expansion on specific point) of a convex function F and looking at the difference between and measure the difference between F and the generated plane.

Let the unknown parameter of interest be μ. Suppose we have a statistic $f(\mathbf{X})$ such that $\mu = \mathbb{E}[f(\mathbf{X})]$. Let $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{X}_i)$, then $\hat{\mu}$ is an unbiased estimator of μ. Suppose we know another statistic $h(\mathbf{X})$ such that $\tau = \mathbb{E}[h(\mathbf{X})]$ is a known value, and let letting $\hat{\tau} = \frac{1}{n}\sum_{i=1}^{n} h(\mathbf{X}_i)$. Then for coefficient $\alpha \in \mathbb{R}$ we can estimate μ by:

$$\hat{\mu}_\alpha = \frac{1}{n}(f(\mathbf{X}_i) + \alpha h(\mathbf{X}_i)) - \alpha\tau = \hat{\mu} + \alpha(\hat{\tau} - \tau),$$

where $\hat{\mu}_\alpha$ is the ***regression estimator*** and the known mean $h(\mathbf{X})$ is the ***control variate*** Lemieux [2017]; Glasserman [2013]; Botev and Ridder [2014]; **?**, it is a variance reduction technique used in Monte Carlo methods. To employ this technique, recall the value we want to estimate $\mathbb{E}_{q(\mathbf{z})} \log \frac{q(\mathbf{z})}{p(\mathbf{z})}$, let

$$\nu(\mathbf{z}) = \frac{p(\mathbf{z})}{q(\mathbf{z})}, \tag{4.8}$$

we have $\mathbb{E}[\nu(\mathbf{z})] = 1$ since the optimization target is to approximate p_θ with q_ϕ and they become identical, *i.e.*, $\frac{q(\mathbf{z}^\star)}{p(\mathbf{z}^\star)} = 1$ when the optimization converge. According to the definition of *regression estimator*, let $\mu(\mathbf{z})$ be the *control variate*, as $\hat{\mu} = \hat{\theta} = \log \frac{q(\mathbf{z})}{p(\mathbf{z})} = \log \nu(\mathbf{z})^{-1} = -\log \nu(\mathbf{z})$ is an unbiased estimator, let the statistic $\hat{\mu} = -\log \nu(\mathbf{z})$, another statistic $\hat{\tau} = \nu(\mathbf{z})$, and $\tau = 1$, we have the regression estimator:

$$\hat{\mu}_\alpha = -\log \nu(\mathbf{z}) + \alpha(\nu(\mathbf{z}) - 1), \tag{4.9}$$

which is by definition also an unbiased estimator of KL-divergence. With respect to the coefficient α, although it is possible calculate the optimal value of α that generate the estimator with minimum variance by:

$$\alpha^\star = -\frac{Cov(\hat{\mu}, \hat{\tau})}{Var(\hat{\mu})}.$$

In most of the scenario the p_θ is considered as not analytical so the whole expression of the estimator is hard to compute. However, according to the log inequality $\forall x > 0$: $f(x) = x - 1$ is the tangent plane to $f(x) = \log x$ with tangent point $x = 1$, such that $x - 1 \geq \log x$. Hence, if $\alpha = 1$ in Eq.(4.9), $\hat{\mu}_\alpha = -\log \nu(\mathbf{z}) + \nu(\mathbf{z}) - 1 \geq 0$ always holds, yielding the definition of our second proposed estimator:

$$\begin{aligned} \hat{\theta}_{\text{Tan}} &= -\log \nu(\mathbf{z}) + \nu(\mathbf{z}) - 1 \\ &= \frac{p(\mathbf{z})}{q(\mathbf{z})} - 1 - \log \frac{p(\mathbf{z})}{q(\mathbf{z})} \\ &= e^{-\hat{\theta}} - 1 + \hat{\theta}, \end{aligned} \tag{4.10}$$

where $\hat{\theta}$ is the naïve estimator. We fix $\alpha = 1$ to guarantee the **non-negativity** of the estimator no matter what the value of $q(\mathbf{z}^{(i)})$ and $p(\mathbf{z}^{(i)})$ are.

4.3.3 Case Study

$\hat{\theta}_{\text{Tan}}$ is unbiased

As $\hat{\theta}$ is unbiased, the generated estimtor $\hat{\theta}_{\text{Tan}}$ is by definition unbiased, i.e.

$$\begin{aligned} B(\hat{\theta}_{\text{Tan}}) &= \mathbb{E}[\hat{\theta}_{\text{Tan}}] - \theta \\ &= \mathbb{E}[-\log k(x) + k(x) - 1] - \mathbb{E}[\log \frac{q(x)}{p(x)}]. \end{aligned} \tag{4.11}$$

By Eq.(4.8) and Eq.(**??**) we have

$$\begin{aligned} &\mathbb{E}[-\log k(x) + k(x) - 1] - \mathbb{E}[\log \frac{q(x)}{p(x)}] \\ &= \mathbb{E}[-\log \frac{p(x)}{q(x)}] + \mathbb{E}[k(x) - 1] - \mathbb{E}[\log \frac{q(x)}{p(x)}] \\ &= \mathbb{E}[\log \frac{q(x)}{p(x)} - \log \frac{q(x)}{p(x)}] + 0 = 0. \end{aligned} \tag{4.12}$$

$\hat{\theta}_{\text{Tan}}$ has relatively low variance

To compare the variance, we still leverage the above example. That is, $\{\hat{\theta}^{(i)}\} = \{\hat{\theta}^{(1)} = -c_2, \hat{\theta}^{(2)} = c_1, \hat{\theta}^{(3)} = c_1, \hat{\theta}^{(4)} = c_2\}$ are i.i.d. where $0 \leq c_1 \leq c_2$ are arbitrary positive constants. This allows us to illustrate an example of the variance of $\hat{\theta}_{\text{Tan}}$

$$\begin{aligned} Var(\hat{\theta}_{\text{Tan}}) &= \mathbb{E}[(\hat{\theta}_{\text{Tan}} - \mathbb{E}[\hat{\theta}_{\text{Tan}}])^2] \\ &= \mathbb{E}[(\hat{\theta}_{\text{Tan}} - \frac{1}{4}\sum_{i}^{4} \hat{\theta}_{\text{Tan}})^2] \\ &= \mathbb{E}[(\hat{\theta}_{\text{Tan}} - \frac{(e^{-c_1} + e^{-c_2} + e^{c_1} + e^{c_2}) - 4 + \mathbb{E}[\hat{\theta}]}{4})^2] \\ &= \mathbb{E}[(\hat{\theta}_{\text{Tan}} - \frac{S}{4} + 1)^2], \end{aligned} \tag{4.13}$$

where $S = (e^{-c_1} + e^{-c_2} + e^{c_1} + e^{c_2})$ and $\hat{\theta}$ is the naïve estimator. In the case of $\{\hat{\theta}^{(i)}\} = \{\hat{\theta}^{(1)} = -0.3, \hat{\theta}^{(2)} = -0.1, \hat{\theta}^{(3)} = 0.1, \hat{\theta}^{(4)} = 0.3\}$, $S \approx 4.10$, and therefore:

$$\begin{aligned} \mathbb{E}[(\hat{\theta}_{\text{Tan}} - \frac{S}{4} + 1)^2] &= \mathbb{E}[(\hat{\theta}_{\text{Tan}} - 1.025 + 1)^2] \\ &= \frac{1}{4}\sum_{i=1}^{4}(e^{-\hat{\theta}^{(i)}} - 1.025 + \hat{\theta}^{(i)})^2 \\ &\approx 0.000417 \ll Var(\hat{\theta}_{\text{Abs}}) < Var(\hat{\theta}) \end{aligned} \tag{4.14}$$

In this example, we get a really small variance that is far less than the variance of $\hat{\theta}_{\text{Abs}}$ and $\hat{\theta}$ that use the same data, which proved the efficacy of control variates technique on variance reduction. Empirically, the variance of $\hat{\theta}_{\text{Tan}}$ is indeed the lowest among

the three proposed estimators. We will show the empirical result in the next section.

4.3.4 D_{TKL}: Tangent-KL-divergence

Similar to D_{AKL}, we propose the Tangent-KL-divergence, abbreviated as TKL-divergence (D_{TKL}), where the estimator $\hat{\theta}$ is reformulated to $\hat{\theta}_{\text{Tan}}$ based on the idea of Bregman divergence.

$$D_{\text{TKL}}(Q\|P) = \int_{\Omega} \left(\frac{p(\mathbf{z})}{q(\mathbf{z})} - 1 - \log \frac{p(\mathbf{z})}{q(\mathbf{z})} \right) q(\mathbf{z}) d\mathbf{z} \tag{4.15}$$

Unfortunately, the D_{TKL} is not symmetric regardless of the reference distribution of space Ω, even if we leverage the PDF $r(\mathbf{z})$ in Eq.(4.7), this metric is not symmetric. Thus the D_{TKL} cannot be a semi-metric. However, such reformulation of the divergence still have one good property, which is the analogism to the family of f-divergence Rényi et al. [1961].

4.3.5 Analogize To f-divergence

The f-divergence is a function $D_f(Q\|P)$ that measures the difference between two probability distributions Q and P

$$D_f(Q\|P) = \int_{\Omega} f\left(\frac{q(\mathbf{z})}{p(\mathbf{z})} \right) p(\mathbf{z}) d\mathbf{z} \tag{4.16}$$

where the function f is a convex function such that $f'(1) = 0$, which could be interpreted as a weighted average of odds ratio given by Q and P. The KL-divergence is a special case of the f-divergence (where $f(t) = t \log t$), and some of the familiar divergences *e.g.* total variation ($f(t) = \frac{|t-1|}{2}$), Jensen-Shannon divergence ($f(t) = (t+1)\log(\frac{2}{t+1} + t \log t)$) also belong to the family of f-divergence.

Under the definition of $\nu(\mathbf{z})$ in Eq.(4.8), we have the conclusion that $\mathbb{E}[\nu(\mathbf{z})] = \mathbb{E}[\nu(\mathbf{z})^{-1}] = 1$, thus we always have an estimator of f-divergence according to the principle of control variates:

$$\hat{\theta}_f^{\star} = f\left(\nu(\mathbf{z})^{-1}\right) - c\left(\nu(\mathbf{z})^{-1} - 1\right) \tag{4.17}$$

Since f is convex function, we let $c = f'(1)$ to get the tangent plane at $\nu(x) = 1$ in order to guarantee the non-negativity of the estimator, which gives us a general estimator for any f-divergence $D_f(Q\|P)$:

$$\begin{aligned} \hat{\theta}_f &= f\left(\nu(\mathbf{z})^{-1}\right) - f'(1)\left(\nu(\mathbf{z})^{-1} - 1\right) \\ &= f\left(\frac{q(\mathbf{z})}{p(\mathbf{z})}\right) - f'(1)\left(\frac{q(\mathbf{z})}{p(\mathbf{z})} - 1\right) \end{aligned} \tag{4.18}$$

To give an illustration in the case of D_{KL} where $f(t) = t \log t$, $f'(t=1) = \log t + 1 = 1$,

replacing t with $\frac{q(\mathbf{z})}{p(\mathbf{z})}$, we have one possible estimator for the D_{KL}:

$$\hat{\theta}_{f_{\mathrm{KL}}} = \frac{q(\mathbf{z})}{p(\mathbf{z})} \log\left(\frac{q(\mathbf{z})}{p(\mathbf{z})}\right) - \frac{q(\mathbf{z})}{p(\mathbf{z})} + 1 \tag{4.19}$$

In this case, we assume the tangent point is at $\nu(\mathbf{z}) = 1$ since its the most straightforward and intuitive, however, we can actually leverage any possible value on the domain of $\nu(\mathbf{z})$ as the tangent point as long as it gives us a feasible tangent plane that avoids negativity.

4.4 Empirical Result

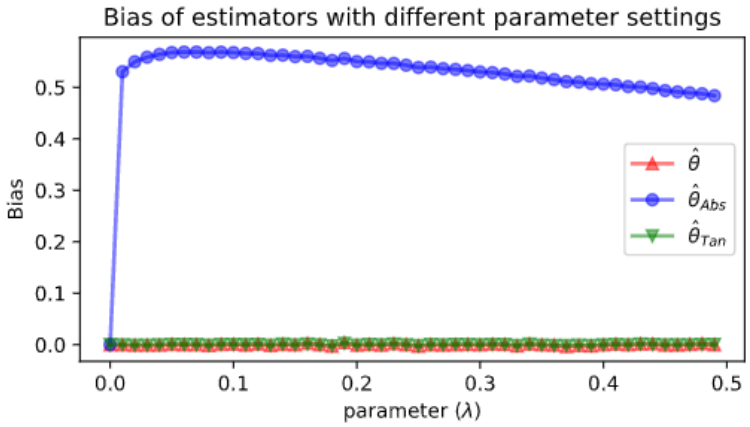


Figure 4.1: Comparison of bias among estimators under continuous Bernoulli distribution with different settings of parameter λ where $p = \mathcal{CB}(0)$ and $q = \mathcal{CB}(\lambda)$.

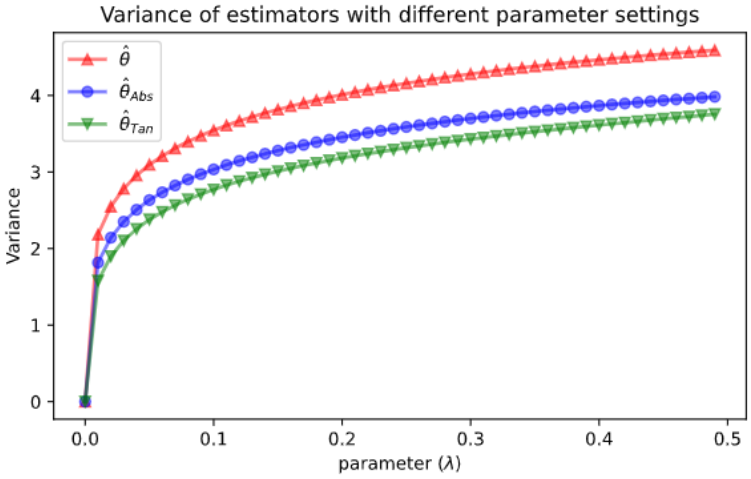


Figure 4.2: Comparison of variance among estimators under continuous Bernoulli distribution with different settings of parameter λ where $p = \mathcal{CB}(0)$ and $q = \mathcal{CB}(\lambda)$.

We firstly demonstrate the bias and variance of three estimators in the setting of continuous Bernoulli distribution [Loaiza-Ganem and Cunningham, 2019] since this is a single parameter distribution and have meaningful interpretations. The continuous Bernoulli is defined on $[0, 1]$ and parameterized by $\lambda \in (0, 1)$, which is defined by:

$$\mathbf{X} \sim \mathcal{CB}(\lambda) \iff p(\mathbf{x}|\lambda) \propto \lambda^{\mathbf{x}}(1-\lambda)^{1-\mathbf{x}} \tag{4.20}$$

we assume the distribution p in the estimator which represents the data distribution always have zero mean after data normalization, hence we assume $p = \mathcal{CB}(0)$. We observe in the empirical result Fig.4.1 that the bias of the tangent estimator is tightly the same to the naïve estimator, since naïve estimator is strictly unbiased, we have enough confident to claim the tangent estimator is strongly unbiased. In Fig.4.2, the tangent estimator achieved the minimum variance among three estimator which demonstrated the effectiveness of our previous variance reduction technique, the absolute estimator has a very close performance to the tangent one, which also demonstrated its respectable efficacy in controlling variance.

These empirical results yields one possible cause of the high variance, which is the influence of mean-shift. To illustrate, lets consider the Gaussian example, where p and q both subject to Gaussian distribution, for the sake of demonstration, we assume they have the same sigma. Define *Mean-Shift* as the absolute value of mean difference between two Gaussians, *i.e.*, *Mean-Shift* $= |\mu_2 - \mu_1|$, in our evaluation, we let $\mu_2 = -\mu_1$, $\sigma = 1$ and increase value of *Mean-Shift* between two Gaussians, Fig.4.3 demonstrated the result

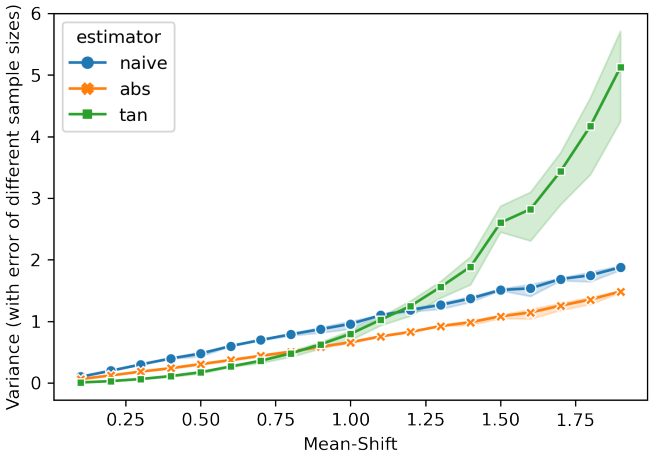


Figure 4.3: Averaged result on variance of different estimators as *Mean-Shift* increase, the shaded area indicates the error for different sample sizes (range from $1e1$ to $1e8$)

where we observe that $\hat{\theta}_{\text{Tan}}$ performs best when the *Mean-Shift* is less than about 0.85 to 0.9, exhibiting the minimum variance, while as the *Mean-Shift* increase, the $\hat{\theta}_{\text{Abs}}$ consistently outperform two other estimators and the $\hat{\theta}_{\text{Tan}}$ becomes sharply unstable, this proves to some extent that $\hat{\theta}_{\text{Tan}}$ is not optimal in all cases, and that $\hat{\theta}_{\text{Abs}}$ is less

sensitive to the data and performs better in some cases, such as the special case of the large *Mean-Shift* we mentioned above.

In Table.4.1, we further demonstrate some empirical results from other distribution family with featured parameter settings. We see approximately half of the variance of $\hat{\theta}_{\text{Tan}}$ outperforms $\hat{\theta}_{\text{Abs}}$ and half on the contrary, thus there is no one best between the two, it mostly depends on the application scenario. In practice, we recommend trying out both estimators on experimental models to find the one that performs best.

4.5 Summary

We introduce in this chapter three of the estimators, respectively:

- Naïve estimator: $\hat{\theta} = \log \frac{q(\mathbf{z})}{p(\mathbf{z})}$;
- Absolute estimator: $\hat{\theta}_{\text{Abs}} = \left| \log \frac{q(\mathbf{z})}{p(\mathbf{z})} \right|$;
- Tangent estimator: $\hat{\theta}_{\text{Tan}} = \frac{p(\mathbf{z})}{q(\mathbf{z})} - 1 - \log \frac{p(\mathbf{z})}{q(\mathbf{z})}$;

for Monte Carlo estimation of KL-divergence. We in detail give derivations, evaluations on bias and variance. Our empirical result shows our hypothesis that the $\hat{\theta}_{\text{Abs}}$ and $\hat{\theta}_{\text{Tan}}$ estimator both have the capability to reduce variance compare against the naïve one, while the $\hat{\theta}_{\text{Tan}}$ estimator can be relatively unbiased while the $\hat{\theta}_{\text{Abs}}$ is biased in half of the situation. In the next chapter, we will bring on some real world applications and evaluate the performance of our introduced estimators on deep learning tasks that utilize variational inference.

distribution family	probability density function	data distribution $p(\cdot;\theta)$	variational distribution $q(\cdot;\phi)$	Bias $B(\hat{\theta})$	Bias $B(\hat{\theta}_{\text{Abs}})$	Bias $B(\hat{\theta}_{\text{Tan}})$	Variance $Var(\hat{\theta})$	Variance $Var(\hat{\theta}_{\text{Abs}})$	Variance $Var(\hat{\theta}_{\text{Tan}})$
Beta	$f(\cdot;\alpha,\beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)}$	$Beta(0.5,0.5)$	$Beta(5,1)$	1.5e-4	1.4e-1	-1.2e-1	6.0e-1	3.6e-1	9.8e1
		$Beta(0.5,0.5)$	$Beta(1,3)$	-7.3e-5	1.7e-1	5.2e-3	5.7e-1	3.4e-1	1e2
		$Beta(0.5,0.5)$	$Beta(2,2)$	4.2e-4	2.1e-1	1e-2	5.9e-1	3.5e-1	6.2e1
		$Beta(0.5,0.5)$	$Beta(2,5)$	-1.2e-4	1.6e-1	-1.1e-1	6.4e-1	3.8e-1	5.4e1
Gaussian	$f(\cdot;\mu,\sigma) = \frac{e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}}{\sigma\sqrt{2\pi}}$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,0.2)$	-1.3e-4	1.1e-1	-3.1e-1	6.7e-1	4.5e-1	3.6e1
		$\mathcal{N}(0,1)$	$\mathcal{N}(0,5)$	-5.3e-3	6.1e-1	-5.6e-3	1.8e1	1.7e1	1.6e1
		$\mathcal{N}(0,1)$	$\mathcal{N}(0.1,1)$	6.2e-6	7.4e-2	7.7e-7	1e-1	6.0e-2	7.1e-3
		$\mathcal{N}(0,1)$	$\mathcal{N}(1,1)$	2.5e-4	3.9e-1	9.4e-5	1.0	6.6e-1	8.4e-1
Laplace	$f(\cdot;\mu,b) = \frac{e^{-\frac{\lvert x-\mu\rvert}{b}}}{2b}$	$Laplace(0,1)$	$Laplace(0,2)$	3.3e-4	3.8e-1	1.4e-4	1.0	7.8e-1	5.7e-1
		$Laplace(0,1)$	$Laplace(0,4)$	-1.0e-3	5.5e-1	-1.1e-3	2.9	2.6	2.4
		$Laplace(0,1)$	$Laplace(0.1,1)$	-4.1e-6	9.2e-2	1.6e-7	9.8e-2	1.2e-2	7.9e-4
		$Laplace(0,1)$	$Laplace(-5,4)$	-1.5e-3	4.9e-1	-6.2e-5	3.8	3.3	3.2

Table 4.1: evaluation results of the bias and variance of different estimators under Beta, Gaussian and Laplace distributions with different parameter settings. The values marked in green are the best of the group and those in red are the worst.

Chapter 5

Application

5.1 Lower Reconstruction Errors of Hyperbolic VAEs

One application of our proposed estimators is to derive new evidence of lower bound estimation (ELBO) regarding variational autoencoders (VAEs). The new ELBO functions are more robust when using MC to estimate D_{AKL} and D_{TKL}.

The original VAEs and the subsequent proposed modifications consider latent representation spaces as Euclidean spaces. However, Euclidean spaces cannot support embedding data in a hierarchical fashion and have limitation in distance metrics, so is not a good representation for many kinds of data structure, *e.g.* graph-like data. To facilitate embedding hierarchies in hyperbolic space, *Poincaré*-VAEs ($\mathcal{P}^c$-VAEs) have recently been introduced [Mathieu et al., 2019], which learn the projection from the latent space to hyperbolic ball using an encoder-decoder architecture. The $\mathcal{P}^c$-VAEs can be viewed as a generalisation of vanilla VAEs, *i.e.*, $\mathcal{P}^c$-VAE $\longrightarrow$ $\mathcal{N}$-VAE when $c = 0$. Nevertheless, when switching from Euclidean spaces to hyperbolic spaces, the analytical form of D_{KL} becomes hard to compute. Thus, again MC techniques are utilized for estimating D_{KL}.

In our experiments, we optimize $\mathcal{P}^c$-VAEs with the new lower bounds (Eq.(5.1)) constructed by our proposed divergence. We investigate different latent dimensions and curvatures of poincaré spaces using the MNIST [LeCun and Cortes, 2010] dataset. We compare the quality of reconstructed images by comparing the reconstruction loss values, and compare the optimization efficiency and correctness by looking at the convergence of D_{KL}.

$$\log p(\mathbf{x}) \geq \underbrace{\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \beta D_{\star\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}))}_{\text{our new lower bound of ELBO}}, \tag{5.1}$$

which can be extended for Riemannian latent spaces by

$$\begin{aligned}
\log p(\mathbf{x}) &= \log \int_{\mathcal{M}} p_\theta(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) d\mathcal{M}(\mathbf{z}) \\
&= \log \int_{\mathcal{M}} \frac{p_\theta(\mathbf{x}|\mathbf{z}) p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} q_\phi(\mathbf{z}|\mathbf{x}) d\mathcal{M}(\mathbf{z}) \\
&\geq \int_{\mathcal{M}} \log \frac{p_\theta(\mathbf{x}|\mathbf{z}) p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} q_\phi(\mathbf{z}|\mathbf{x}) d\mathcal{M}(\mathbf{z}) \\
&= \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x}) \mathcal{M}(\mathbf{z})} [\log p_\theta(\mathbf{x}|\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x}) + \log p(\mathbf{z})] \\
&\geq \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x}) \mathcal{M}(\mathbf{z})} [\log p_\theta(\mathbf{x}|\mathbf{z})] - \beta D_{\star \mathrm{KL}}(q_\theta(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z})).
\end{aligned} \tag{5.2}$$

The $\mathcal{M}$ is a smooth manifold which is a set of points $\mathbf{z}$. The $D_{\star \mathrm{KL}}$ can be replaced by D_{KL}, D_{AKL} and D_{TKL}. For each, we evaluated the performance of two βs, *i.e.*, the trade-off coefficient of the KL-divergence term against the reconstruction error term in ELBO i.e. $\beta = 1$ and $\beta = 2$. More specifically, the β in Eq.(5.1) stands for the KKT [Mangasarian, 1994; Gordon and Tibshirani, 2012] multiplier for the Lagrangian of the original ELBO function, which can be intepreted as a regularisation coefficient that constrains the capacity of the latent information channel $\mathbf{z}$ and puts implicit independence pressure on the learnt posterior [Higgins et al., 2016]. We change this parameter to adjust the degree of applied learning pressure during training so that we can test the ability of our proposed divergence in encouraging the latent representation learning, in comparison with the vanilla KL-divergence.

5.1.1 On Averaged Test Reconstruction Error

We conduct an ablation study on the usefulness of different divergence on higher dimensional latent space. To do so, we estimate the reconstruction loss for poíncare models achieved by different latent dimension where the curvature c of hyperbolic spaces is a fixed negative constant to get a warped manifold.

As Fig.5.1 and Fig.5.2 illustrated, although using D_{AKL} and D_{TKL} as new lower bounds of ELBO seems unable to decrease reconstruction loss with lower dimensional latent space, as the latent spaces dimension become larger, the reconstruction loss of $\mathcal{P}^c$-VAE-KL starts to rise slowly while those with D_{AKL} and D_{TKL} drop significantly on latent dimension approximately equal to 30, suggesting that optimizing the ELBO with D_{AKL} and D_{TKL} obtains increasing advantages over using D_{KL}.

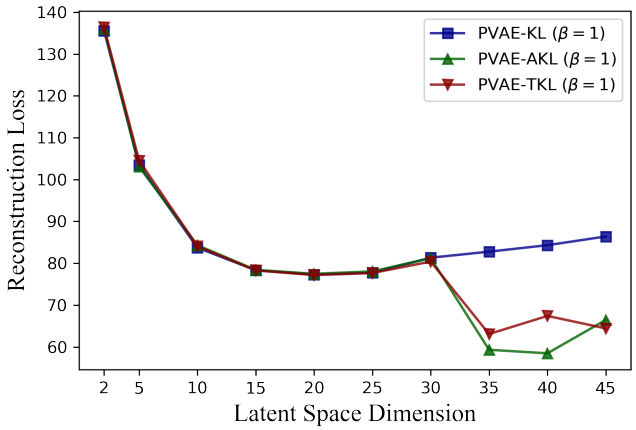


Figure 5.1: Comparison of reconstruction loss of optimizing p-VAEs with KL-, AKL- and TKL-divergence where the weight $\beta = 1$, versus dimensionality of the latent space.

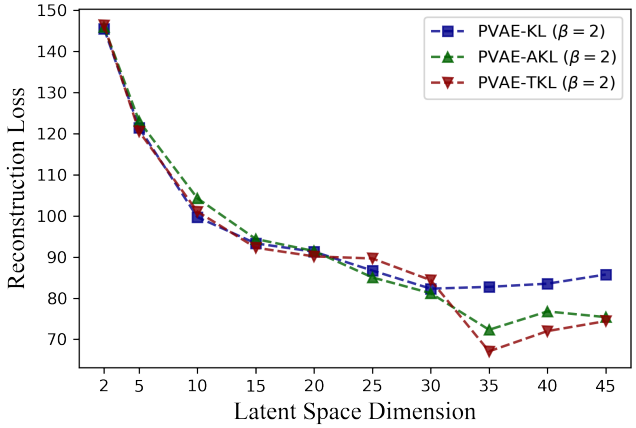


Figure 5.2: Comparison of reconstruction loss of optimizing p-VAEs with KL-, AKL- and TKL-divergence where the weight $\beta = 2$, versus dimensionality of the latent space.

5.1.2 On Different Curvatures Of Hyperbolic Manifold

We also investigated the performance as curvature c increases. We fix the latent dimension to be 40. From Fig.5.3 to Fig.5.6, we can observe that as curvature increases, MC estimation of KL-divergence reflects a downward trend toward negative approximated values, resulting in more noise and growing reconstruction loss. In contrast, greater curvature values lead to lower reconstruction loss for optimization with AKL-divergence and TKL-divergence.

The inaccuracy for KL estimation is specifically demonstrated by the negative KL estimation values Fig.5.5(b) and Fig.5.6(b), where we observe that the blue curve that represents the estimation KL-divergence by naïve estimator returned a negative

value although the KL is by definition non-negative (the estimations below the pink dashed line are considered as errors), such results suggest serious problems with those estimators suffering from the uncertainty of non-negativity, especially for high-dimensional dependent variables and high curvature models.

With our proposed estimators, such miscalculation has improved, particularly in Fig.5.5 and Fig.5.6, compared to Fig.5.3 and Fig.5.4, the red and dark-red curve indicating the D_{AKL} estimation, the green and dark-green curve indicating the D_{TKL} estimation are much better than the D_{KL} estimation (blue and dark-blue curve) by having a non-negative convergence in the divergence term and the overall ELBO loss term.

On average, with parameter $\beta = 1$, the model performs slightly better than those with $\beta = 2$, reflecting in a lower reconstruction loss, however, the divergence loss of $\beta = 2$ outperforms those with $\beta = 1$. This is due to the loss of high frequency details when passing through a constrained latent bottleneck controlled by β, when $\beta > 1$, then it will put a stronger constraint on the latent bottleneck than in the original VAE and thus force the model to learn better disentangled representations from input data, resulting in more informative $q_\phi(z|x)$ and thus approximate better to the ground truth.

5.2 Realistic Human Appearance Generation

Let x be an image of an object from a dataset. A standard variational auto-encoder architecture with two latent variables is not suitable for learning disentangled representations of y and z where in this case y indicates the shape of human body (*i.e.*, geometrical information) and z represents the appearance (*i.e.*, intrinsic appearance characteristics).

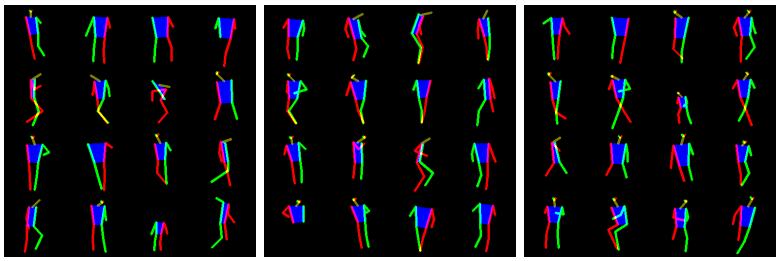
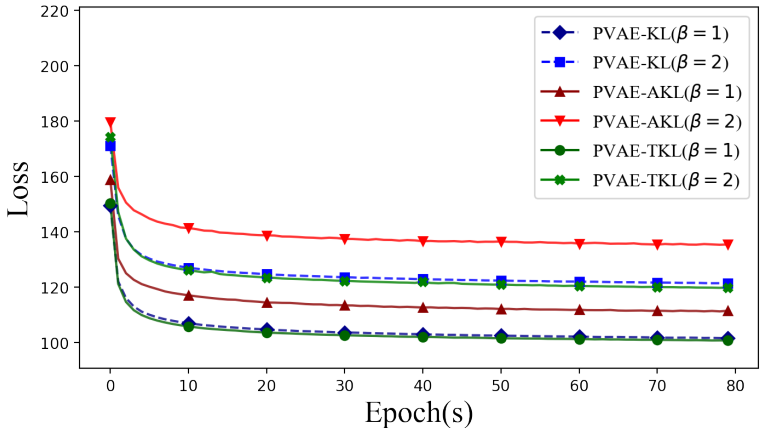
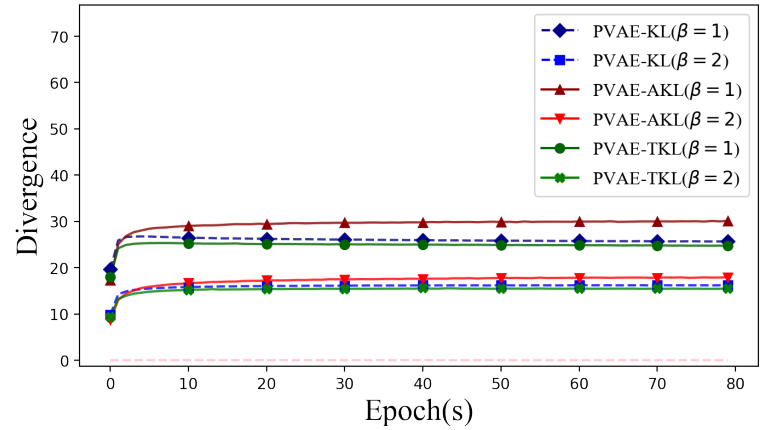


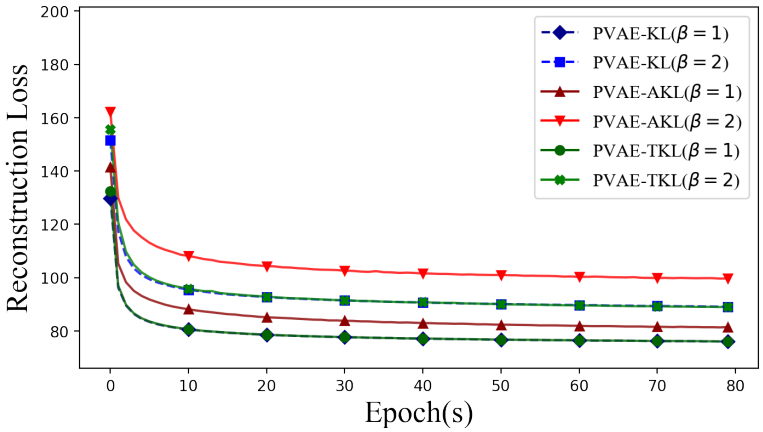
Figure 5.7: Examples of human geometrical information y in the Market-1501 dataset [Zheng et al., 2015a]



(a) The evidence lower-bound, identical to KL + Reconstruction

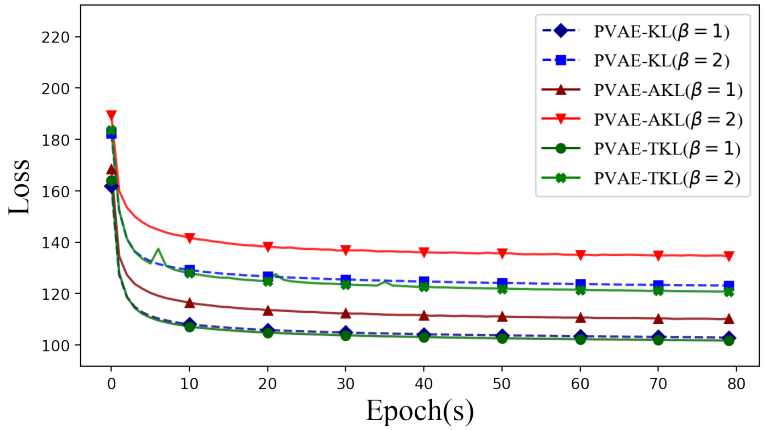


(b) Convergence of the KL-divergence term

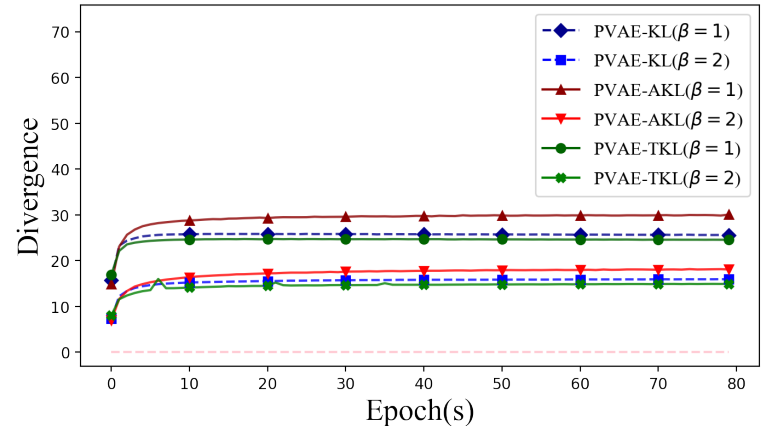


(c) Convergence of the reconstruction term

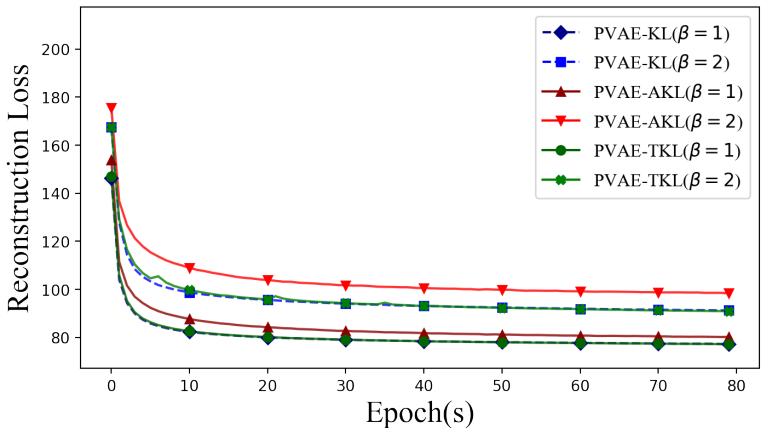
Figure 5.3: Comparison of ELBO, reconstruction loss and divergence of optimizing p-VAEs with KL-divergence and AKL-divergence and TKL-divergence, vs. curvature $c = 0.1$. The latent dimension is set to be 40.



(a) The evidence lower-bound, identical to KL + Reconstruction

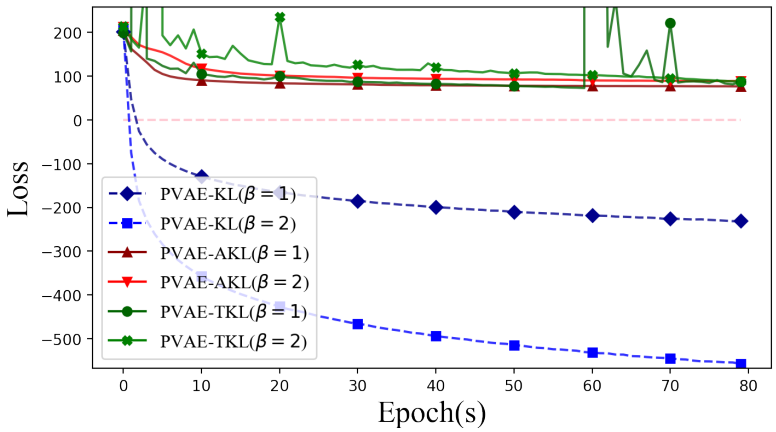


(b) Convergence of the KL-divergence term

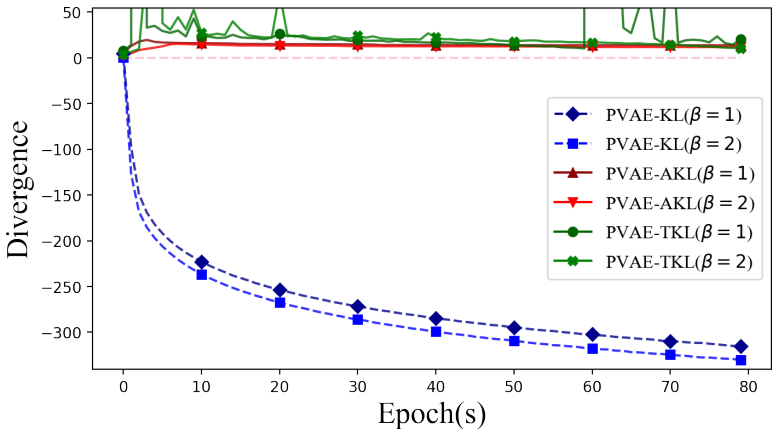


(c) Convergence of the reconstruction term

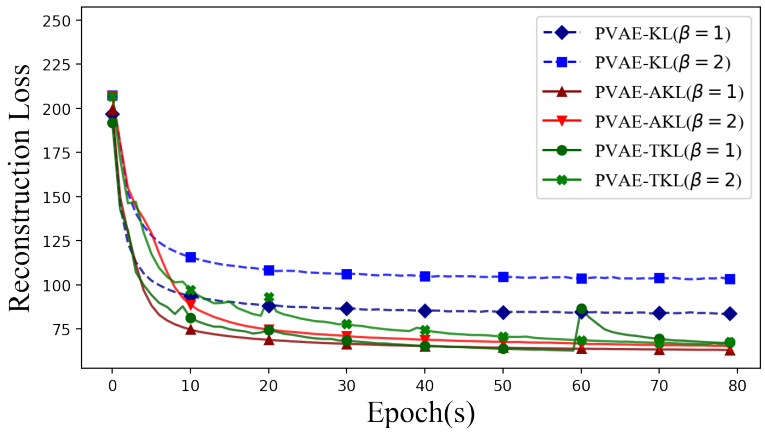
Figure 5.4: Comparison of ELBO, reconstruction loss and divergence of optimizing p-VAEs with KL-divergence and AKL-divergence and TKL-divergence, vs. curvature $c = 0.2$. The latent dimension is set to be 40.



(a) The evidence lower-bound, identical to KL + Reconstruction

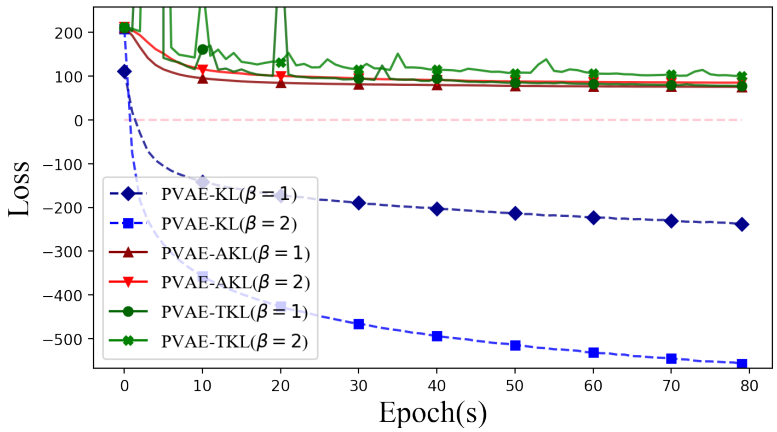


(b) Convergence of the KL-divergence term

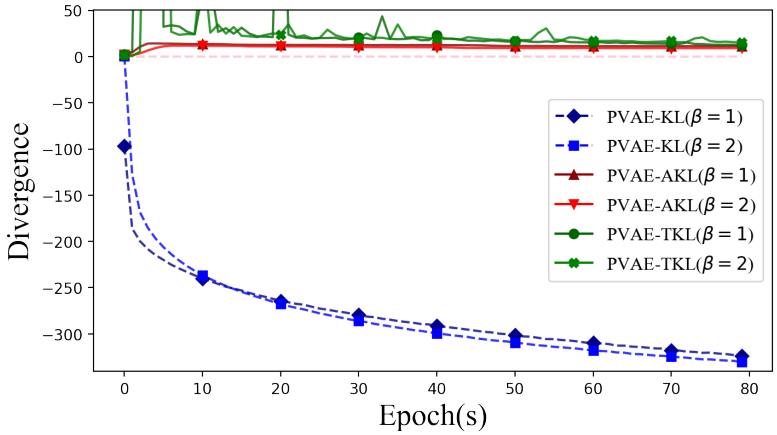


(c) Convergence of the reconstruction term

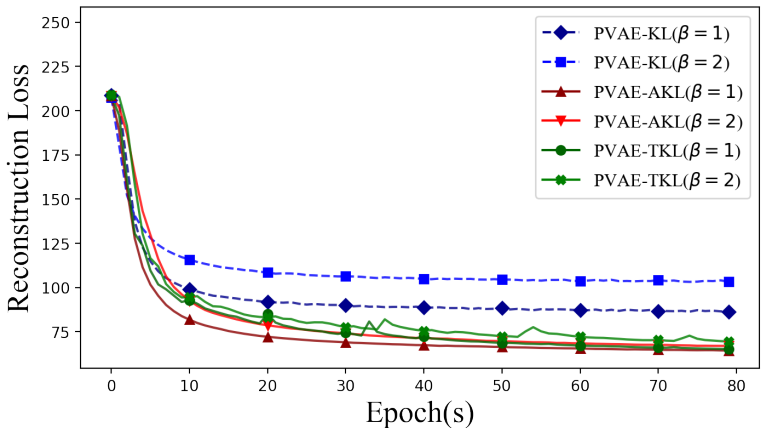
Figure 5.5: Comparison of ELBO, reconstruction loss and divergence of optimizing p-VAEs with KL-divergence and AKL-divergence and TKL-divergence, vs. curvature $c = 0.7$. The latent dimension is set to be 40.



(a) The evidence lower-bound, identical to KL + Reconstruction



(b) Convergence of the KL-divergence term



(c) Convergence of the reconstruction term

Figure 5.6: Comparison of ELBO, reconstruction loss and divergence of optimizing p-VAEs with KL-divergence and AKL-divergence and TKL-divergence, vs. curvature $c = 1.4$. The latent dimension is set to be 40.

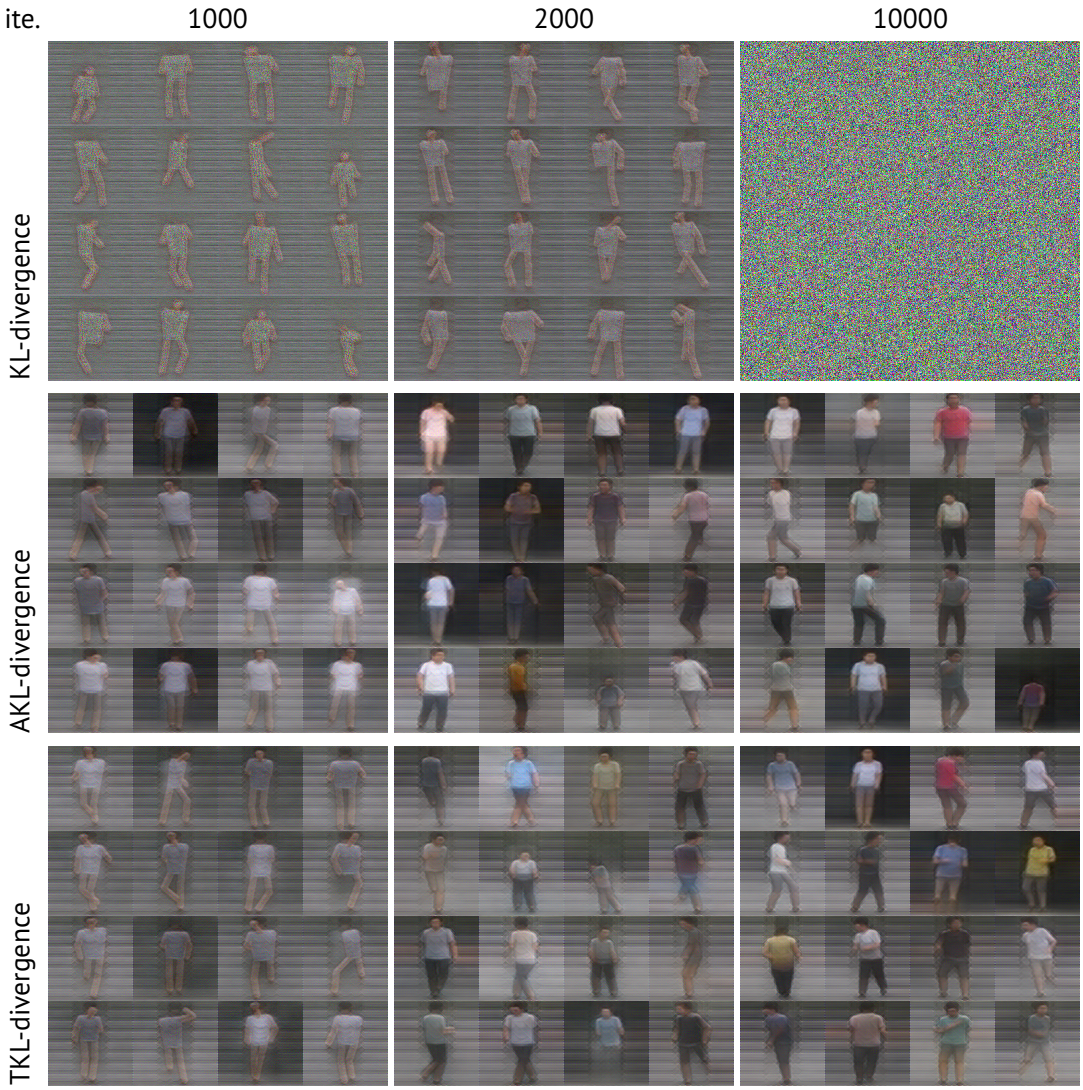


Figure 5.8: Comparison of optimizing vunets with KL- and AKL- and TKL-divergence on the market-1501 dataset [Zheng et al., 2015a]. The first row illustrates the generated images of the vunet [Esser et al., 2018] trained with KL-divergence; the second and third row displays the generated results of the vunet trained with our proposed divergence. The *ite.* stands for iteration.

The variational u-nets (vunet) [Esser et al., 2018] utilizes a conditional variational auto-encoder to model appearance information y with the help of an estimator e, *i.e.*, $\hat{y} = e(x)$. The task is therefore converted to inferring the latent z from the image and the estimate $\hat{y}$, we achieve this by maximizing their conditional log-likelihood (Eq.(5.3)).

$$\log p(\mathbf{x}|\hat{\mathbf{y}}) = \log \int_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}|\hat{\mathbf{y}}) d\mathbf{z} \geq \mathbb{E}_{q(\mathbf{z}|\mathbf{x},\hat{\mathbf{y}})} \log \frac{p(\mathbf{x}, \mathbf{z}|\hat{\mathbf{y}})}{q(z|\mathbf{x}, \hat{\mathbf{y}})} \tag{5.3}$$

Thus the parameter of Eq.(5.3) can be optimized by variational inference with the following form of the ELBO loss function:

$$ELBO(\theta, \phi; \mathbf{x}) = \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{x}|\mathbf{z}, \hat{\mathbf{y}})] - D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}, \hat{\mathbf{y}}) \| p_\theta(\mathbf{z}|\hat{\mathbf{y}})) \tag{5.4}$$

We performed experiments utilizing vunet on the Market-1501 dataset [Zheng et al., 2015a]. The aim here is to generate different appearances given a human pose via Vunet [Esser et al., 2018]. The implemented loss function of vunet includes a MC estimation of KL-divergence to measure the difference between the prior and posterior conditional distributions. Market-1501 is a dataset consisting of a large array of human pose images. The images are collected in a market without artificial designs. We presented generated appearances of vunets trained with KL-, AKL- and TKL-divergence respectively in Fig.5.8. The results illustrate that replacing KL-divergence in Eq.(5.4) with AKL- and TKL-divergence can lead to more realistic generated appearances.

5.3 Summary

This chapter demonstrated two applications of the proposed estimator, respectively, for hyperbolic variational auto-encoder and variational u-net for human appearance generation. Our first experiment on p-VAE [Mathieu et al., 2019] induces the efficacy of newly formulated divergence by our estimators compared to the vanilla estimator. The remedy effect for the negative estimation was particularly exceptional when it comes to high dimensional and highly curved latent space. In our second experiment, we demonstrate the comparison of optimizing vunets with different divergence on the market-1501 dataset, which illustrates the generated images of the vunet trained with KL-divergence and compared them to the generated results of the vunet trained with our proposed divergence. With the iteration increase, the separation of the final effect becomes more and more pronounced, to the point that at very high iterations, the result with normal KL-divergence has large noise. In contrast, our augmented models stay stable enough to generate clearer images. With these two experiments, we have demonstrated the effectiveness of our estimators, and we will be adding more experiments in the future.

Chapter 6

Conclusion

6.1 Brief Summary

Although estimating D_{KL} with MC methods is a commonly applied technique, such estimation can lead to unrealistic negative values of D_{KL}, whereas by definition, D_{KL} is non-negative. To fix such severe noise in MC, we propose Absolute-KL-divergence (D_{AKL}) and Tangent-KL-divergence (D_{TKL}), respectively, based on the absolute estimator (θ_{Abs}) and the tangent estimator (θ_{Tan}). The MC estimations of D_{AKL} and D_{TKL} are intrinsically non-negative, using MC for our proposed divergence is free from the previous unrealistic estimated results. We further show D_{AKL} satisfies the axioms of a semi-metric therefore D_{AKL} is a plausible quantity to measure the difference between two distributions, and demonstrate the analogy of the concept of D_{TKL} toward the family of f-divergence. In practice, our derived divergence can help design VAEs. We replace KL-divergence with AKL- and TKL-divergence to derive a new lower bound of the traditional VAE ELBO. Via optimizing the new lower ELBO, we show that two VAEs can have lower reconstruction loss values and generate more visually-sharp images. We believe the newly introduced NKL-divergence can inspire research in variational inference and signal processing.

6.2 Future Directions

Though MC techniques are not widely used in variational deep learning models since in a bayesian view, we can choose conjugate prior (*e.g.* the Gaussian distribution) and a corresponding posterior distribution family to make the log-likelihood maximization analytical in most scenarios, such that both p and q distributions have to be differentiable with respect to their parametrization. However in some situations, for instance, the latent space is assumed to be non-Euclidean, where the KL does not have a closed-form solution independent on the latent space, we can only utilize MC techniques to estimate the KL. [Skopek et al., 2020] has conclusively discussed the generalizations of probability distributions to hyperbolic manifolds where all variants of those generalized distribution are reparametrizable, differentiable, and the KL can be computed using Monte Carlo estimation. Since there are different generalizations of probability distributions (*e.g.* Wrapped, Maximizing entropy, etc.),

the problem we discussed in this dissertation may still exist (a special case p-VAE was demonstrated in our experiments), hence it still requires more verification if our method would contribute significantly positively or not to the final performance of generalized models.

Chapter 7

Appendix I

7.1 Software platform

In general, our experiments are conducted on the python platform and we use anaconda to manage the python packages. For the two different experiments, we have different package dependencies. For detail please refer to the Tab.7.1.

7.2 Hardware platform

The Poincaré and the distribution evaluation experiments are less demanding for computing power, so I use my local machine for these experiments. Running the the variational u-net model is very demanding on both graphic card memory and RAM, special thanks to Mr.Zhenyue and Mrs.Yang for offering me the cloud computing server, the specification of these hardware are listed in Tab.7.2.

7.3 Artifact

GitHub repository [Here]

The code is succeed to the Poincare-VAE official implementation. This repo is code for a PyTorch implementation of Poincare Variational Auto-Encoder. KL-divergence inside Evidence Lower-Bound is replaced by Tangent/Absolute-KL-divergence.

7.3.1 Prerequisites

- Install Prerequisite packages: `pip install -r -U requirements.txt`

7.3.2 Backbone Model

The Poincare-VAE (Mathieu et al (2019) [Paper], [Repo]):

- Curvature (`--c`): 1.0
- Prior distribution (`--prior`): `WrappedNormal` or `RiemannianNormal`

- Posterior distribution (`--posterior`): `WrappedNormal` or `RiemannianNormal`
- Decoder architecture (`--dec`):
  - `Linear` (MLP)
  - `Wrapped` (logarithm map followed by MLP),
  - `Geo` (first layer is based on geodesic distance to hyperplanes, followed by MLP)
  - `Mob` (based on Hyperbolic feed-forward layers from Ganea et al (2018))
- Encoder architecture (`--enc`): `Wrapped` or `Mob`
- Estimator (`--est`): `tan`, `abs` or `naive` (default).

7.3.3 Directory structure

```
README.md
data
   |-- .gitkeep
experiments
   |-- .gitkeep
pvae
   |-- __init__.py
   |-- datasets
   |   |-- __init__.py
   |   |-- datasets.py
   |-- distributions
   |   |-- __init__.py
   |   |-- ars.py
   |   |-- hyperbolic_radius.py
   |   |-- hyperspherical_uniform.py
   |   |-- riemannian_normal.py
   |   |-- wrapped_normal.py
   |-- main.py
   |-- manifolds
   |   |-- __init__.py
   |   |-- euclidean.py
   |   |-- poincareball.py
   |-- models
   |   |-- __init__.py
   |   |-- architectures.py
   |   |-- mnist.py
   |   |-- tabular.py
   |   |-- vae.py
```

```
   |-- objectives.py
   |-- ops
   |   |-- __init__.py
   |   |-- manifold_layers.py
   |-- utils.py
   |-- vis.py
requirements.txt
run_357.sh
run_all.sh
run_vae_40_1.sh
run_vae_40_2.sh
run_vae_40_3.sh
run_vae_40_4.sh
run_vae_60_1.sh
run_vae_60_2.sh
run_vae_60_3.sh
run_vae_60_4.sh
run_vae_80_1.sh
run_vae_80_2.sh
run_vae_80_3.sh
run_vae_80_4.sh
tests
   |-- __init__.py
   |-- test_hyperbolic_radius.py
   |-- test_hyperspherical_uniform.py
```

7.3.4 Trainning

MNIST dataset

Command for different experiment presets:

- curvature=0.1, latent_dim=40: `./run_vae_40_1.sh`
- curvature=0.2, latent_dim=40: `./run_vae_40_2.sh`
- curvature=0.7, latent_dim=40: `./run_vae_40_3.sh`
- curvature=1.4, latent_dim=40: `./run_vae_40_4.sh`
- curvature=0.1, latent_dim=60: `./run_vae_60_1.sh`
- curvature=0.2, latent_dim=60: `./run_vae_60_2.sh`
- curvature=0.7, latent_dim=60: `./run_vae_60_3.sh`
- curvature=1.4, latent_dim=60: `./run_vae_60_4.sh`
- curvature=0.1, latent_dim=80: `./run_vae_80_1.sh`
- curvature=0.2, latent_dim=80: `./run_vae_80_2.sh`
- curvature=0.7, latent_dim=80: `./run_vae_80_3.sh`
- curvature=1.4, latent_dim=80: `./run_vae_80_4.sh`

- Custom dataset via csv file (placed in `/data`, no header, integer labels on last column): `python3 pvae/main.py --model csv --data-param CSV_NAME --data-size NB_FEATURES`

7.3.5 Acknowledgement

Special thanks to Mr. Zhenyue Qin in Australian National University; Mrs. Yang Liu and Dr. Saeed Anwar in Data61 CSIRO; Dr. Pan Ji in OPPO US Research Center.

7.3.6 Additional Links

- Geoopt: Riemannian Optimization in PyTorch: [Link]
- Monte Carlo theory, methods and examples: [Link]

Experiment	Package	Version	Note
Poincaré VAE Sec.5.1	numpy	≥1.17.4	
	scikit-learn	≥0.21.3	
	scipy	≥1.3.2	
	seaborn	≥0.9.0	
	torch	≥1.4.0	
	torchvision	=0.4.2	
	geoopt	NA	This package is the unofficial implementation of [Kochurov et al., 2020]. Please refer to the [latest release]
Variational U-Net (Apperance Generation) Sec.5.2	tensorflow-gpu	=1.10.1	
	numpy	≥1.14.5	
	opencv-python	≥3.4.3.18	
	Pillow	≥5.2.0	
	tqdm	≥4.26.0	This package provides the progress bar, its optional but we suggest installing it.
	PyYAM	≥3.13	
	h5py	≥2.8.0	
Distribution Family Evaluation Sec.4.4	numpy	=1.20.1	
	matplotlib	=3.3.4	
	torch	=1.7.1	

Table 7.1: Specification of the package requirements for each experiment.

Experiment	Type	Model	Pieces	Technology
Sec.5.1 Poincaré & Sec.4.4 Evaluation	CPU	i7-7700HQ	1	3.90GHz Kaby Lake
	GPU	GTX1060	1	6GB×1
	RAM	∼	2	8GB×2 1600MHz DDR3
Variational U-Net Sec.5.2	CPU	i7-9900k	1	Maximum 5.00 GHz Coffee Lake
	GPU	Tesla P100	4	16GB×4
	RAM	∼	∼	64GB 2666MHz DDR4

Table 7.2: Specification of the use of hardware for each experiment.

Bibliography

Abou-Moustafa, K. T. and Ferrie, F. P., 2012. A note on metric properties for some divergence measures: The gaussian case. In *Asian Conference on Machine Learning*, 1–15. (cited on pages 25 and 28)

Bauschke, H. H. and Borwein, J. M., 2001. Joint and separate convexity of the bregman distance. In *Inherently Parallel Algorithms in Feasibility and Optimization and their Applications* (Eds. D. Butnariu; Y. Censor; and S. Reich), vol. 8 of *Studies in Computational Mathematics*, 23–36. Elsevier. doi:https://doi.org/10.1016/S1570-579X(01)80004-5. https://www.sciencedirect.com/science/article/pii/S1570579X01800045. (cited on pages 25 and 29)

Belghazi, M. I.; Baratin, A.; Rajeshwar, S.; Ozair, S.; Bengio, Y.; Courville, A.; and Hjelm, D., 2018. Mutual information neural estimation. In *International Conference on Machine Learning*, 531–540. (cited on page 1)

Botev, Z. and Ridder, A., 2014. Variance reduction. *Wiley StatsRef: Statistics Reference Online*, (2014), 1–6. (cited on page 30)

Chen, J.-Y.; Hershey, J. R.; Olsen, P. A.; and Yashchin, E., 2008. Accelerated monte carlo for kullback-leibler divergence between gaussian mixture models. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, 4553–4556. IEEE. (cited on page 1)

David Stutz, 2021. Collection of latex resources and examples — github repository. https://github.com/davidstutz/latex-resources. [Online; accessed 11-March-2021]. (cited on pages ix and 21)

Dempster, A. P.; Laird, N. M.; and Rubin, D. B., 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39, 1 (1977), 1–22. (cited on page 13)

Dieng, A. B.; Tran, D.; Ranganath, R.; Paisley, J.; and Blei, D., 2017. Variational inference via chi upper bound minimization. In *Advances in Neural Information Processing Systems*, 2732–2741. (cited on page 1)

Esser, P.; Sutter, E.; and Ommer, B., 2018. A variational u-net for conditional appearance and shape generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8857–8866. (cited on pages x, 45, and 46)

Galas, D. J.; Dewey, G.; Kunert-Graf, J.; and Sakhanenko, N. A., 2017. Expansion of the kullback-leibler divergence, and a new class of information metrics. *Axioms*, 6, 2 (2017), 8. (cited on page 28)

Glasserman, P., 2013. *Monte Carlo methods in financial engineering*, vol. 53. Springer Science & Business Media. (cited on page 30)

Goldberger, J.; Gordon, S.; and Greenspan, H., 2003. An efficient image similarity measure based on approximations of kl-divergence between two gaussian mixtures. In *ICCV*, 487. IEEE. (cited on page 1)

Gordon, G. and Tibshirani, R., 2012. Karush-kuhn-tucker conditions. *Optimization*, 10, 725/36 (2012), 725. (cited on page 38)

Hershey, J. R. and Olsen, P. A., 2007. Approximating the kullback leibler divergence between gaussian mixture models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, vol. 4, IV–317. IEEE. (cited on page 1)

Higgins, I.; Matthey, L.; Pal, A.; Burgess, C.; Glorot, X.; Botvinick, M.; Mohamed, S.; and Lerchner, A., 2016. beta-vae: Learning basic visual concepts with a constrained variational framework. (2016). (cited on page 38)

Jensen, J. L. W. V., 1906. Sur les fonctions convexes et les inégualités entre les valeurs Moyennes. doi:10.1007/bf02418571. https://doi.org/10.1007/bf02418571. (cited on page 11)

Kingma, D. P. and Ba, J., 2017. Adam: A method for stochastic optimization. (cited on page 12)

Kingma, D. P. and Welling, M., 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, (2013). (cited on pages 1, 11, and 20)

Kochurov, M.; Karimov, R.; and Kozlukov, S., 2020. Geoopt: Riemannian optimization in pytorch. (cited on page 53)

LeCun, Y. and Cortes, C., 2010. MNIST handwritten digit database. (2010). http://yann.lecun.com/exdb/mnist/. (cited on pages 15 and 37)

Lemieux, C., 2017. *Control Variates*, 1–8. American Cancer Society. ISBN 9781118445112. doi:https://doi.org/10.1002/9781118445112.stat07947. https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118445112.stat07947. (cited on pages 25 and 30)

Loaiza-Ganem, G. and Cunningham, J. P., 2019. The continuous bernoulli: fixing a pervasive error in variational autoencoders. *arXiv preprint arXiv:1907.06845*, (2019). (cited on page 34)

MacKay, D. J. and Mac Kay, D. J., 2003. *Information theory, inference and learning algorithms*. Cambridge university press. (cited on page 1)

Malinin, A. and Gales, M., 2019. Reverse kl-divergence training of prior networks: Improved uncertainty and adversarial robustness. In *Advances in Neural Information Processing Systems*, 14547–14558. (cited on page 1)

Mangasarian, O. L., 1994. *Nonlinear programming*. SIAM. (cited on page 38)

Mathieu, E.; Le Lan, C.; Maddison, C. J.; Tomioka, R.; and Teh, Y. W., 2019. Continuous hierarchical representations with poincaré variational auto-encoders. In *Advances in neural information processing systems*, 12565–12576. (cited on pages 37 and 46)

Nair, V. and Hinton, G. E., 2010. Rectified linear units improve restricted boltzmann machines. In *Icml*. (cited on pages ix and 21)

Nielsen, F., 2020. Non-negative monte carlo estimation of f-divergences. (2020). (cited on pages 1 and 29)

Nowozin, S.; Cseke, B.; and Tomioka, R., 2016. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems*, 271–279. (cited on page 1)

Ontañón, S., 2020. An overview of distance and similarity functions for structured data. *Artificial Intelligence Review*, (2020), 1–43. (cited on page 28)

Rényi, A. et al., 1961. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. The Regents of the University of California. (cited on page 32)

Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J., 1986. Learning representations by back-propagating errors. *nature*, 323, 6088 (1986), 533–536. (cited on page 5)

Skopek, O.; Ganea, O.-E.; and Bécigneul, G., 2020. Mixed-curvature variational autoencoders. In *International Conference on Learning Representations*. https://openreview.net/forum?id=S1g6xeSKDS. (cited on page 47)

Wainwright, M. J. and Jordan, M. I., 2008. *Graphical models, exponential families, and variational inference*. Now Publishers Inc. (cited on page 7)

Wikipedia contributors, 2021a. Entropy (information theory) — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Entropy_(information_theory)&oldid=1004670944. [Online; accessed 18-February-2021]. (cited on page 10)

Wikipedia contributors, 2021b. Kullback–leibler divergence — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Kullback\T1\textendashLeibler_divergence&oldid=1005801944. [Online; accessed 18-February-2021]. (cited on page 10)

Wilson, W. A., 1931. On semi-metric spaces. *American Journal of Mathematics*, 53, 2 (1931), 361–373. (cited on pages 25 and 28)

Zheng, L.; Shen, L.; Tian, L.; Wang, S.; Wang, J.; and Tian, Q., 2015a. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision*, 1116–1124. (cited on pages x, 40, 45, and 46)

Zheng, S.; Jayasumana, S.; Romera-Paredes, B.; Vineet, V.; Su, Z.; Du, D.; Huang, C.; and Torr, P. H., 2015b. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision*, 1529–1537. (cited on page 7)

Australian National University

INDEPENDENT STUDY CONTRACT

Note: Enrolment is subject to approval by the Honours/projects co-ordinator

SECTION A (Students and Supervisors)

UniID: u6920122

FAMILY NAME: Jiaxu PERSONAL NAME(S): Liu

PROJECT SUPERVISOR (*may be external/HDR*): Zhenyue Qin

FORMAL SUPERVISOR (*a RSCS academic*): Tom Gedeon

COURSE CODE, TITLE AND UNIT: COMP4560 Advanced Computing Project 12 units

SEMESTER(S) ☒ S2 YEAR: _2020_ ☒ S1 YEAR: _2021_

PROJECT TITLE:

Fixing the Unreliable Monte Carlo Estimation on KL-divergence within Variational Autoencoders

LEARNING OBJECTIVES:

- A high-level understanding of variational inference and variational autoencoders.
- Strong experience in implementing neural networks.
- Paper writing experience and logic thinking.

PROJECT DESCRIPTION:

- Write a literature survey
- **Background**: The objective function of variational autoencoders (VAE) consists of two components: the reconstruction loss and the KL-divergence between a posterior and a prior distribution. Despite the popularity of applying Monte-Carlo estimation in estimating the latter term, we argue such an estimation can be astray. This project aims to propose an alternative method for estimating KL divergence, intending to fix the unreliability of Monte-Carlo estimation on the KL term.
- Expectation: We expect the student to:
  1. Scrutinise our arguments claiming the Monte-Carlo estimation on KL-divergence is unreliable.
  2. Examine the correctness of our new VAE objective in its convergence to the true likelihood.
  3. Conduct at least three experiments to investigate the alternative VAE objective.
  4. Participate in writing a paper.
  5. If time allows, re-propose the theory in an information-theoretic framework.
- Write report

ASSESSMENT (as per course's project rules web page, with the differences noted below):

☐ Honours (24 credit)	(fixed)	☒ Projects (6-12 credit)	…. / (fixed)
Assessed project components:	**% of mark**	**Assessed project components:**	**% of mark**
Thesis	(85%)	**Thesis** (reviewer mark)	45 45-60%
Presentation	(10%)	**Artefact** (supervisor project mark)	45 30-45%
Critical Feedback	(5%)	**Presentation**	(10%)

MEETING DATES (IF KNOWN):

Weekly

STUDENT DECLARATION: I agree to fulfil the above defined contract:

刘嘉伟 T.D. Gedeon 26/Jul/2020.

Signature Date

SECTION B (Supervisor):

I am willing to supervise and support this project. I have checked the student's academic record and believe this student can complete the project.

秦震岳 26/Jul/2020

Signature Date

Reviewer:

Name: Fatemeh Saleh Signature: Agreed email 30 Jul 20

Reviewer 2: (for Honours only)

Name: ..... Signature: .....

REQUIRED DEPARTMENT RESOURCES:

SECTION C (Honours / Projects coordinator approval)

…….. ……

Signature Date