Enhancing Generative Modeling with Multivariate g-k Distribution in Normalizing Flows

Yuyao Wang

May 2020

1 Introduction to Normalizing Flows

Normalizing Flows (NFs) are a powerful framework for constructing flexible and tractable probability distributions. They are particularly useful in scenarios where we need to approximate complex distributions, such as in generative modeling, variational inference, and density estimation. At their core, Normalizing Flows transform a simple base distribution into a more complex one through a sequence of invertible and differentiable mappings. This process allows us to retain the ability to compute both the density and sample from the transformed distribution efficiently.

1.1 Mathematical Foundation

Given a random variable $\mathbf{z}_0$ with a simple base distribution $p_{Z_0}(\mathbf{z}_0)$, we aim to construct a more complex distribution $p_{Z_K}(\mathbf{z}_K)$ through a series of transformations. Each transformation f_k in the flow modifies the current variable $\mathbf{z}_k$ to produce the next variable $\mathbf{z}_{k+1}$:

$$\mathbf{z}_{k+1} = f_k(\mathbf{z}_k), \quad k = 0, \ldots, K-1$$

The overall transformation from $\mathbf{z}_0$ to $\mathbf{z}_K$ is then:

$$\mathbf{z}_K = f_K \circ f_{K-1} \circ \cdots \circ f_1(\mathbf{z}_0)$$

To compute the probability density function of $\mathbf{z}_K$, we use the change of variables formula:

$$p_{Z_K}(\mathbf{z}_K) = p_{Z_0}(\mathbf{z}_0) \left| \det \left(\frac{\partial f^{-1}}{\partial \mathbf{z}_K} \right) \right|$$

Or equivalently:

$$p_{Z_K}(\mathbf{z}_K) = p_{Z_0}(\mathbf{z}_0) \prod_{k=1}^{K} \left| \det \left(\frac{\partial \mathbf{z}_k}{\partial \mathbf{z}_{k-1}} \right) \right|$$

The Jacobian determinant $\det\left(\frac{\partial \mathbf{z}_k}{\partial \mathbf{z}_{k-1}}\right)$ quantifies how the volume of the distribution changes under the transformation f_k.

1.2 Why Use Normalizing Flows?

In many practical applications, the distributions we wish to model are highly complex and multimodal. Traditional approaches, like Gaussian distributions in VAEs, are often insufficient to capture such complexities. Normalizing Flows provide a solution by enabling the construction of highly flexible distributions that can approximate a wide variety of data distributions, including those that are multimodal or have intricate structures.

By stacking simple invertible transformations, Normalizing Flows can transform a simple base distribution into one that closely matches the target distribution. This flexibility makes them particularly well-suited for tasks in generative modeling, where accurate representation of the data distribution is crucial.

1.3 Integration of Multivariate g-k Distribution and Levenberg-Marquardt Optimization

In our approach, we enhanced the flexibility of Normalizing Flows by integrating the multivariate g-k distribution as the base distribution. The g-k distribution is a flexible family of distributions that can model skewness and kurtosis, providing a richer initial representation of the data. The parameters of this distribution, however, are non-trivial to estimate due to their non-linear nature.

To address this, we applied the Levenberg-Marquardt (LM) optimization algorithm, a technique well-suited for solving non-linear least squares problems. The LM algorithm allowed us to efficiently and precisely estimate the parameters of the multivariate g-k distribution, ensuring that the base distribution is well-tuned to the underlying data. This, in turn, improved the overall expressiveness of the Normalizing Flow, leading to more accurate generative models, particularly in the context of generating high-quality MNIST digits.

1.4 Conclusion

Normalizing Flows provide a powerful framework for building flexible probabilistic models by transforming simple base distributions into complex ones. The integration of the multivariate g-k distribution and the use of LM optimization further enhance this framework, allowing for precise control over the base distribution and improving the model's ability to capture complex data distributions. These advances are crucial in tasks such as generative modeling, where the quality of the generated samples is directly tied to the flexibility and accuracy of the underlying distribution.

2 Overview of Multivariate g-k Distribution and Levenberg-Marquardt Optimization

2.1 Multivariate g-k Distribution

The g-k distribution is a flexible family of probability distributions that allows for the modeling of skewness and kurtosis. This distribution is particularly useful in situations where the data exhibits non-Gaussian features such as asymmetry or heavy tails.

The univariate g-k distribution is typically defined by the following quantile function:

$$Q(u; a, b, g, k) = a + b \left[1 + c \cdot \frac{1 - \exp(-g \cdot u)}{1 + \exp(-g \cdot u)} \right] \cdot \left(1 + u^2\right)^k \cdot u$$

where:

- u is the quantile (ranging from 0 to 1),
- a is the location parameter,
- b is the scale parameter,
- g controls the skewness,
- k controls the kurtosis,
- c is typically set to 0.8 to stabilize the skewness transformation.

The multivariate extension of the g-k distribution involves applying this quantile function to each dimension of the data independently. For a d-dimensional random vector $\mathbf{U}$ with independent components U_i, the multivariate g-k distribution is given by:

$$\mathbf{Z} = \mathbf{a} + \mathbf{b} \odot \left[1 + c \cdot \frac{1 - \exp(-\mathbf{g} \odot \mathbf{U})}{1 + \exp(-\mathbf{g} \odot \mathbf{U})} \right] \odot \left(1 + \mathbf{U}^2\right)^{\mathbf{k}} \odot \mathbf{U}$$

where $\mathbf{a}, \mathbf{b}, \mathbf{g}, \mathbf{k} \in R^d$ are vectors of the parameters for each dimension, and $\odot$ denotes the element-wise multiplication.

The g-k distribution is not typically used in its probability density form because it is often difficult to obtain analytically. Instead, the distribution is defined through its quantile function, making it easier to sample from, which is particularly useful in the context of Normalizing Flows.

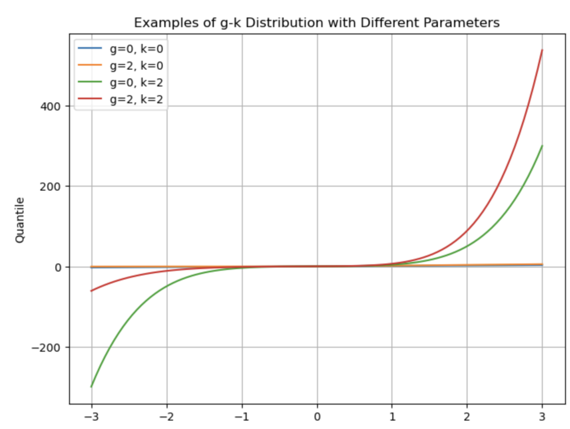

Figure 1: Illustration of the flexibility of the g-k distribution in capturing different shapes of data distributions.

2.2 Levenberg-Marquardt Optimization

Levenberg-Marquardt (LM) optimization is a popular algorithm for solving non-linear least squares problems, which is particularly useful when fitting complex models such as the g-k distribution.

The goal of LM optimization is to minimize the sum of squared residuals:

$$\text{minimize} \sum_{i=1}^{n} [f_i(\mathbf{x}) - y_i]^2$$

where $f_i(\mathbf{x})$ represents the model function dependent on the parameters $\mathbf{x}$, and y_i are the observed data points.

The LM algorithm combines the gradient descent method and the Gauss-Newton method. It adjusts the parameter vector $\mathbf{x}$ iteratively by solving:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \left[\mathbf{J}_k^\top \mathbf{J}_k + \lambda_k \mathbf{I}\right]^{-1} \mathbf{J}_k^\top \mathbf{r}_k$$

where:

- $\mathbf{J}_k$ is the Jacobian matrix of the residuals with respect to the parameters at the k-th iteration,
- $\mathbf{r}_k$ is the vector of residuals,
- λ_k is a damping parameter that controls the balance between the Gauss-Newton method (when λ_k is small) and gradient descent (when λ_k is large),
- $\mathbf{I}$ is the identity matrix.

The damping parameter λ_k is adjusted dynamically during the optimization process. If a step results in a reduction of the residual, λ_k is decreased (favoring Gauss-Newton), otherwise, it is increased (favoring gradient descent). This adaptive strategy allows the LM algorithm to handle the complexities of non-linear problems effectively.

In the context of fitting a multivariate g-k distribution, LM optimization is employed to adjust the parameters $\mathbf{a}, \mathbf{b}, \mathbf{g}, \mathbf{k}$ so that the distribution generated by the quantile function closely matches the target data distribution. This fitting process involves minimizing the difference between the generated samples and the observed data.

2.3 Why Use Multivariate g-k and LM Optimization?

The multivariate g-k distribution provides a highly flexible model for capturing the skewness and kurtosis often observed in real-world data. However, due to the non-linear nature of its parameters, fitting the distribution to data is challenging. Levenberg-Marquardt optimization is particularly well-suited for this task, as it can efficiently handle the non-linear least squares problem posed by the parameter estimation of the g-k distribution.

By integrating the multivariate g-k distribution into the base of a Normalizing Flow, we enhance the model's ability to represent complex, multimodal distributions. The LM optimization ensures that the g-k parameters are finely tuned, leading to more accurate and expressive generative models, as demonstrated in tasks such as digit generation in the MNIST dataset.

This combination of advanced statistical modeling (g-k distribution) with robust optimization techniques (Levenberg-Marquardt) exemplifies the state-of-the-art approaches in modern generative modeling, pushing the boundaries of what is achievable in terms of model flexibility and data fidelity.

3 Why MNIST Dataset?

The MNIST dataset is a widely recognized benchmark in the field of machine learning and computer vision, particularly for evaluating generative models. Here are the key reasons why MNIST was chosen for testing our algorithm:

1. **Simplicity and Accessibility**: MNIST consists of grayscale images of handwritten digits (0-9), which are simple yet sufficiently complex for testing various machine learning models. The dataset's accessibility and standardized format make it ideal for comparing different models and algorithms.

2. **Benchmarking and Comparability**: MNIST is one of the most commonly used datasets in the field, with a vast amount of prior research. This allows for easy benchmarking and comparison of our algorithm's performance against established methods.

3. **Data Characteristics**: Despite its simplicity, MNIST exhibits variability in the style of handwriting, which introduces challenges in learning the underlying distribution. This makes it an excellent test case for evaluating how well our model captures and generates diverse patterns.

4. **Evaluation of Generalization**: The variability in handwriting styles also serves as a test for the generalization capabilities of our model. A model that performs well on MNIST is likely to be robust in terms of generating or recognizing variations in more complex datasets.

4 Why Normalizing Flows in VAE?

Normalizing Flows are integrated into the Variational Autoencoder (VAE) framework to address the limitations of the standard VAE, particularly in representing complex data distributions. Here's why Normalizing Flows are beneficial in the context of VAEs:

1. **Enhanced Latent Space Flexibility**: In a standard VAE, the latent space is typically modeled by a simple Gaussian distribution, which may not be flexible enough to capture the intricacies of real-world data distributions. Normalizing Flows introduce a series of invertible transformations to the latent space, allowing it to better model complex, multimodal distributions.

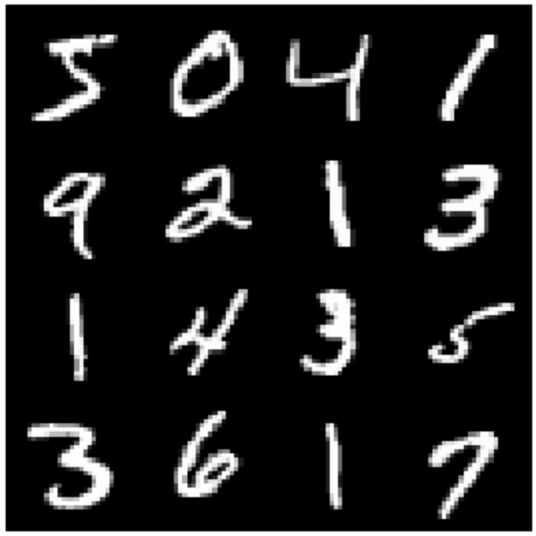

Figure 2: Sample images from the MNIST dataset.

2. **Improved Generative Performance**: By using Normalizing Flows, the VAE can transform a simple latent distribution into a more expressive one, leading to higher quality generated samples. This is crucial for tasks like image generation, where the goal is to produce realistic and diverse outputs.

3. **Tractability in Inference and Sampling**: Normalizing Flows maintain the ability to efficiently compute the probability density and to sample from the distribution. This makes them particularly suitable for VAEs, where both inference and generative tasks are performed.

4. **Better Representation of Data**: The integration of Normalizing Flows into VAEs allows for a more accurate representation of the underlying data distribution, leading to improved reconstructions and more reliable generation of new samples. This is particularly important in generative modeling tasks, such as those involving the MNIST dataset, where capturing the subtle variations in handwritten digits is key to the model's success.