Summer internship report

Alexander Stepikin

July - August 2020

1 Introduction: variational inference

Approximation of difficult-to-compute densities of probability distributions is an important problem of Bayesian statistics. First, we have discussed the article [1] which provides an observation of Variational inference approach to solve this problem.

Consider a statistical model $p(z, x) = p(z)p(x|z)$, where $x \in \mathbb{R}^n$ is a vector of observed variables and $z \in \mathbb{R}^m$ is a vector of latent (unobserved) variables. In a Bayesian model the latent variables are drawn from a prior distribution $p(z)$ and the observations come from a conditional distribution $p(x|z)$. The goal of inference in such models is to compute the density of posterior distribution $p(z|x)$.

It can be written:

$$p(z|x) = \frac{p(z, x)}{p(x)} = \frac{p(z, x)}{\int p(z, x)dz}. \tag{1}$$

The problem occurs when the integral in equation (1) is not available in a closed form or its computation requires exponential time. In this case an efficient approximation of $p(z|x)$ is needed.

Variational inference (or VI) offers us a method to solve this problem with optimization: we can specify a family of distributions $\mathfrak{F}$ and find its member $q^*(z)$ which is the closest to the posterior $p(z|x)$. The «closeness» of distributions is measured with Kullback-Leibler divergence.

Thus, the problem can be formulated in a following way:

$$q^*(z) = \underset{q(z)\in\mathfrak{F}}{\operatorname{argmin}} KL(q(z)||p(z|x)), \tag{2}$$

where $KL(q(z)||p(z|x)) = \mathbb{E}_{q(z)}[\log q(z)] - \mathbb{E}_{q(z)}[\log p(z|x)]$.

In general, $KL(q(z)||p(z|x))$ is not computable, because it includes intractable posterior $p(z|x)$. That's why we can equivalently maximize the evidence lower bound (ELBO) function which is equal to the negative considered KL divergence up to an added constant:

$$ELBO(q) = \mathbb{E}_{q(z)}[\log p(z, x)] - \mathbb{E}_{q(z)}[\log q(z)] = \log p(x) - KL(q(z)||p(z|x)), \tag{3}$$

where $\log p(x)$ is a constant w.r.t. $q(z)$.

In addition, we can observe that $\log p(x) \geq ELBO(q)$. Consequently, maximizing the objective function ($ELBO(q)$) results in maximizing the log evidence $\log p(x)$ which is our general goal.

In [2] $ELBO(q)$ is reformulated in different equivalent ways.

The complexity of the family $\mathfrak{F}$ determines the difficulty of the problem (2). In [1] the authors are focused on the mean-field variational family:

$$\mathfrak{F} = \left\{ q(z) : q(z) = \prod_{j=1}^{m} q_j(z_j) \right\}. \tag{4}$$

For this case, the coordinate ascent variational inference (CAVI) algorithm is proposed and derived in [1].

Finally, for conditionally conjugate models (a special class of exponential family models) where CAVI becomes computationally unaffordable there is an alternative algorithm – stochastic variational inference (SVI) that scales to massive data.

2 Variational auto-encoder

Second, we have read and discussed the article [3] where Stochastic Gradient Variational Bayes (SGVB) estimator of ELBO and Auto-Encoding Variational Bayes (AEVB) algorithm were proposed. Variational auto-encoder (VAE) was given as an example.

Consider the same problem of posterior approximation in Bayesian model $p(x, z)$. Let us assume that the prior distribution $p_{\lambda^*}(z)$ and conditional distribution $p_{\theta^*}(x|z)$ come from parametric distribution families $p_\lambda(z)$ and $p_\theta(x|z)$ parameterized by the vectors of variational parameters λ and θ. The true parameters λ^*, θ^* and latent variables z that take part in the generation of data x are unknown.

The authors introduce a recognition model $q_\varphi(z|x)$ that approximates the intractable true posterior $p_\theta(z|x)$ and is parameterized by φ.

Assuming that the datapoints x are i.i.d., we have: $\log p_{\theta,\lambda}(x) = \sum_{i=1}^n \log p_{\theta,\lambda}(x^{(i)})$.

Applying VI approach, we can introduce $ELBO(q_\varphi(z|x^{(i)})) = ELBO(\theta, \varphi, \lambda; x^{(i)})$ as we have done it before:

$$\log p_{\theta,\lambda}(x^{(i)}) = KL(q_\varphi(z|x^{(i)})||p_\theta(z|x^{(i)})) + ELBO(\theta, \varphi, \lambda; x^{(i)}) \geq ELBO(\theta, \varphi, \lambda; x^{(i)}). \tag{5}$$

According to [2],

$$ELBO(\theta, \varphi, \lambda; x^{(i)}) = \mathbb{E}_{q_\varphi(z|x^{(i)})}[\log p_{\theta,\lambda}(x^{(i)}, z) - \log q_\varphi(z|x^{(i)})] = \mathbb{E}_{q_\varphi(z|x^{(i)})}[\log p_\theta(x^{(i)}|z)] - KL(q_\varphi(z|x^{(i)})||p_\lambda(z)). \tag{6}$$

The idea of SGVB estimator of ELBO is to apply reparametrization trick. In particular, for a chosen approximate posterior $q_\varphi(z|x)$ we can reparametrize random variable $\tilde{z} \sim q_\varphi(z|x)$ using a differentiable transformation:

$$\tilde{z} = g_\varphi(\varepsilon, x) \text{ where } \varepsilon \sim p(\varepsilon) \tag{7}$$

and $p(\varepsilon)$ is a marginal distribution of auxiliary variable ε.

After that, we can use Monte-Carlo estimates of expectation (the first term) in equation (6). KL divergence (the second term) often can be integrated analytically. If it is not possible, it can be approximated in the same way.

For example, $q_\varphi(z|x)$ is often chosen to be a normal distribution $N(z|\mu, \sigma^2 I)$. In this case, we can sample $z^{(i)} \sim N(z|\mu^{(i)}, (\sigma^2)^{(i)})$ using the suitable reparametrization: $z^{(i)} = g_\varphi(x^{(i)}, \varepsilon) = \mu^{(i)} + \sigma^{(i)} \cdot \varepsilon$ where $\varepsilon \sim N(0, 1)$.

AEVB algorithm includes optimization of the objective (ELBO) w.r.t. all the parameters computing it with SGVB estimator.

Finally, we can interpret our model in a following way. Given the datapoint x the recognition model $q_\varphi(z|x)$ produces the distribution over the possible values of latent variables z from which x could have been generated. This part of the model is referred to as probabilistic «encoder». At the same time $p_\theta(x|z)$ is called a probabilistic «decoder» since given a latent variable z it produces the distribution of the corresponding values of x.

If both probabilistic encoder and decoder are parameterized by Neural Networks (MLPs), we come to VAE architecture.

To sum up this section, VAE is a Neural Network that consists of 2 parts: encoder and decoder. Encoder takes data x and produces the distribution $q_\varphi(z|x)$ (e.g. normal) – its latent representation. Decoder takes a sample from $q_\varphi(z|x)$ and constructs the distribution $p_\theta(x|z)$ that should approximate the initial data x. The loss function equals to negative $ELBO(\theta, \varphi, \lambda; x)$ given by equation (6).

3 Prior choice problem

As we can see in the previous section, the choice of the prior distribution $p_\lambda(z)$ in VAE is not specified. Usually, a standard normal prior $p_\lambda(z) = N(z|0, I)$ with no parameters is chosen.

On the one hand, a simple prior is easy to work with (e.g. $KL(q_\varphi(z|x^{(i)})||p_\lambda(z))$ can be computed analytically). On the other hand, according to equation (6), maximizing ELBO could force the encoder to produce the distribution $q_\varphi(z|x)$ which is close to the simple prior (e.g. $p_\lambda(z) = N(z|0, I)$). It may cause over-regularization and not make the latent representation $q_\varphi(z|x)$ expressive enough.

This issue is discussed in the article [4]. The idea is to chose a prior which is more expressive than a standard normal one.

For example, we can think of mixture of Gaussians (MoG) prior $p_\lambda(z) = \frac{1}{K}\sum_{k=1}^{K} N(\mu_k, \sigma_k^2)$ with trainable parameters $\lambda = \{\mu_k, \sigma_k^2\}_{k=1}^K$ (K is chosen in advance). This may be the simplest alternative to the standard normal prior.

The authors of [4] propose the Variational Mixture of Posteriors Prior (VampPrior) that outperforms MoG and standard normal priors.

According to [2], we can rewrite ELBO function one more time:

$$ELBO(\theta, \varphi, \lambda, x) = \mathbb{E}_{q_\varphi(z|x)}[\log p_{\theta,\lambda}(z, x)] + \mathbb{H}[q_\varphi(z|x)], \tag{8}$$

where $\mathbb{H}[q_\varphi(z|x)] = -\mathbb{E}_{q_\varphi(z|x)}[\log q_\varphi(z|x)]$ is called «entropy».

Or,

$$ELBO(\theta, \varphi, \lambda) = \mathbb{E}_{x\sim q(x)}[\mathbb{E}_{q_\varphi(z|x)}[\log p_\theta(x|z)]] + \mathbb{E}_{x\sim q(x)}[\mathbb{H}[q_\varphi(z|x)]] + \mathbb{E}_{z\sim q(z)}[\log p_\lambda(z)], \tag{9}$$

where $q(x)$ is the empirical distribution.

The first term in equation (9) is the negative reconstruction error, the second one is the expectation of the entropy and the third one is cross-entropy between the aggregated posterior $q_z = \frac{1}{n}\sum_{l=1}^{n} q_\varphi(z|x_l)$ and the prior $p_\lambda(z)$. It can be shown that this third term is maximized when

$$p_\lambda^*(z) = q(z) = \frac{1}{n}\sum_{l=1}^{n} q_\varphi(z|x_l). \tag{10}$$

However computing the aggregated posterior may be too expensive as we need to go through all the n datapoints. Moreover, choosing such a prior might lead to overfitting.

Considering all the problems, the authors suggest using the VampPrior that approximates the aggregated posterior. The idea is to use $K \ll n$ trainable pseudo-inputs u_k instead of the real ones:

$$p_\lambda(z) = \frac{1}{K}\sum_{k=1}^{K} q_\varphi(z|u_k). \tag{11}$$

In this case, the prior parameters are $\lambda = (u_1, \ldots, u_K, \varphi)$ and they are trained with backpropagation.

On the one hand, the VampPrior is expected to be a good approximation of the aggregated posterior and, thus, it maximizes the third term of the objective (equation (9)). On the other hand, choosing $K \ll n$ prevents us from overfitting and makes the prior easier to compute. There are also some other advantages of the VampPrior mentioned in the paper.

The experiments conducted by the authors of [4] have proven that the VampPrior (applied with a special two-layered VAE architecture) shows state-of-the-art results on 6 different datasets.

4 The task and experiments

The practical part of our work was to try and implement VAE with different priors and compare the results[1].

First, we have generated the following dataset X. We created a mixture of 2 Gaussian distributions with fixed parameters (p_{data} – the probability of the first peak, μ_1, μ_2 and $covar = I$ – means and covariance matrix of 2 multivariate normal distributions) and got $N = 10000$ samples from it.

Data X is visualized at the Figure 1.

[1]The code and the results are available at GitHub: `https://github.com/stalex2902/VAE`

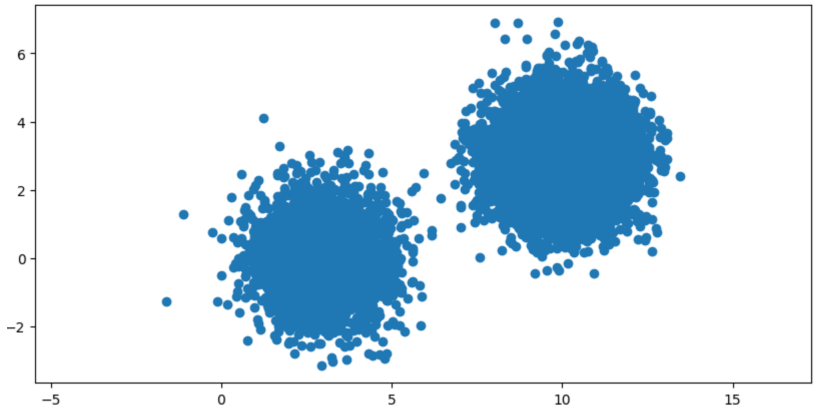

Figure 1: Data X

We trained VAE consisting of 3-layered encoder and 3-layered decoder with the dimension of latent space $z_{\text{dim}} = 2$ and different priors. The distribution $q_\varphi(z|x)$ was chosen to be a normal one parameterized by its mean μ and variance σ^2 (outputs of the encoder). The distribution $p_\theta(x|z)$ was also multivariate normal with trainable mean μ_0 (output of the decoder) and fixed covariance matrix $covar = I$. We compared the results in data generation and reconstruction ($X_{\text{rec}} = \text{decoder}(\text{encoder}(X))$). Data generation is done in the following way. We get $pr\_samples$ – 1000 samples from the prior – and process them through the decoder ($X_{\text{gen}} = \text{decoder}(pr\_samples)$).

1. VAE with Standard Normal prior.

   We started with the easiest and typical prior choice – a standard normal one $p(z) = N(0, 1)$. We trained our VAE for $num\_epoches = 250$ epoches. The loss function is defined by negative equation (6). The first term $\mathbb{E}_{q_\varphi(z|x)}[\log p_\theta(x|z)]$ was computed using 1-MC approximation, the second term $KL(q_\phi(z|x)||p_\lambda(z)) = KL(N(z; \mu, \sigma^2)||N(z; 0, 1))$ was integrated analytically as it was shown in [3] (appendix B).

   The visual results of data reconstruction and generation are depicted at the Figure 2.

   In addition, we decided to visualize the latent space: latent representation of datapoints that have come from different Gaussians in initial distribution. The corresponding means and log-variances of $q_\varphi(z|x)$ distributions are at the Figure 2 as well.

   As we can see, there is a clear border between the representations of the datapoints from 2 different peaks. Taking this into account, we decided to try another type of prior – mixture of 2 Gaussians prior with trainable parameters.

2. VAE with MoG prior

   In this case, the prior has the following form: $p_\lambda(z) = p_{\text{prior}} \cdot N(m_1, \sigma_1^2) + (1 - p_{\text{prior}}) \cdot N(m_2, \sigma_2^2)$, i.e. $\lambda = (p_{\text{prior}}, m_1, m_2, \sigma_1, \sigma_2)$.

   We trained VAE with the same architecture. The loss function was computed using 1-MC estimates of $\mathbb{E}_{q_\varphi(z|x)}[\log p_\theta(x|z)]$ and $\mathbb{E}_{q_\varphi(z|x)}[p_\lambda(z)]$.

   (a) First, we trained VAE with MoG prior with fixed parameter $p_{\text{prior}} = 0.5$. Another parameters of the prior were randomly initialized and trained.

   The visual results of data reconstruction and generation in this instance and the trained prior visualization are at the Figure 3.

   (b) Second, we made the parameter p_{prior} trainable and got the results depicted at the Figure 4.

   It occurred that after training $p_{\text{prior}} \approx p_{\text{data}}$ (or $p_{\text{prior}} \approx 1 - p_{\text{data}}$). Moreover, 2 modes of the trained prior are separated from each other in both cases. It means that the trained prior has the similar structure as the initial data distribution.

Comparing figures 2-4, we can observe that VAE with MoG prior (both variants) manages to generate and reconstruct data X more precisely because the peaks are more clearly separated. At the same time, we cannot see any significant difference between the results in data generation and reconstruction by VAEs with MoG priors with fixed and trainable parameter p_{prior} (figures 3 and 4).

Finally, we have made the estimation of average log-likelihood $\log p_{\theta,\lambda}(x)$ using importance sampling.

As we have already mentioned, $\log p_{\theta,\lambda}(X) = \sum_{i=1}^{n} \log p_{\theta,\lambda}(x^{(i)})$. For a fixed $x \in X$ we have:

$$\log p_{\theta,\lambda}(x) = \log \int_Z p_{\theta,\lambda}(x,z)dz = \log \int_Z q_\varphi(z|x)\frac{p_{\theta,\lambda}(x,z)}{q_\varphi(z|x)}dz \approx \log \frac{1}{N_{IS}} \sum_{i=1}^{N_{IS}} \frac{p_\theta(x|z_i)p_\lambda(z_i)}{q_\varphi(z_i|x)}, \tag{12}$$

where $\{z_i\}_{i=1}^{N_{IS}}$ – samples from $q_\varphi(z|x)$.

We have generated a test dataset X_{test} consisting of $N_{\text{test}} = 5000$ datapoints and estimated mean and std of $\log p_{\theta,\lambda}(x)$ on this dataset using $N_{IS} = 2000$ samples.

The results are at the Table 1.

$\log p_{\theta,\lambda}(x)$	Standart Normal prior	MoG prior with fixed $p_{\text{prior}} = 0.5$	MoG prior with trainable p_{prior}
mean	-0.76	-0.82	-1.00
std	0.14	0.18	0.29

Table 1: Results of $\log p_{\theta,\lambda}(X)$ estimation

Log-likelihood $\log p_{\theta,\lambda}(X)$ is the initial objective to maximize when training a generative model. That is why we expected that this parameter would be higher for VAEs with MoG priors. However, according to our estimations, it occurred to be false.

This issue is covered in the article [5]. The authors show that the main parameters used to compare different trained generative models (e.g. average log-likelihood and visual fidelity of samples) are largely independent from each other. There are examples when one model produces better visual results while having lower average log-likelihood than the other one and vice versa.

We assume that such thing could happen in our case: it is obvious that VAE with MoG prior shows better (visual) results in both data generation and reconstruction but, according to the estimations, has lower average $\log p_{\theta,\lambda}(x)$.

5 Conclusion

To conclude, we have discussed the issue of posterior approximation and VI approach to solve this problem, the architecture of VAE and how it is used (e.g. for the tasks of data generation and reconstruction). We paid attention the problem of prior choice and discussed strategies of using complex priors instead of the standard one that have already been proposed. The theoretical part of the work was based on three articles: [1], [3] and [4].

The practical part included the implementation of VAEs with different prior types: the standard normal one and the Mixture of 2 Gaussians prior with trainable parameters (in 2 variants). We have compared the visual results in data generation and reconstruction and made sure that VAE with MoG prior works better. We visualized trained MoG prior and saw that it tends to simulate the structure of the initial data distribution. Finally, we tried to estimate average log-likelihood of the trained models using importance sampling. The results were quite contradictory: visually, trained VAE with MoG prior generate and reconstruct the initial data better that the one with standard normal prior, however, the average log-likelihood in this case appeared to be lower. This discrepancy can probably be explained as it was done in the article [5] which covers the same issue. It means that, when choosing the criteria of comparing different models, first of all, we should take into account what is more important from the practical point of view.

The work can be further developed. For example, it is necessary to estimate average log-likelihood more precisely: make sure that the models were trained to convergence. It would be also interesting to implement VampPrior proposed in [4] and think of new ones.

6 Visual results

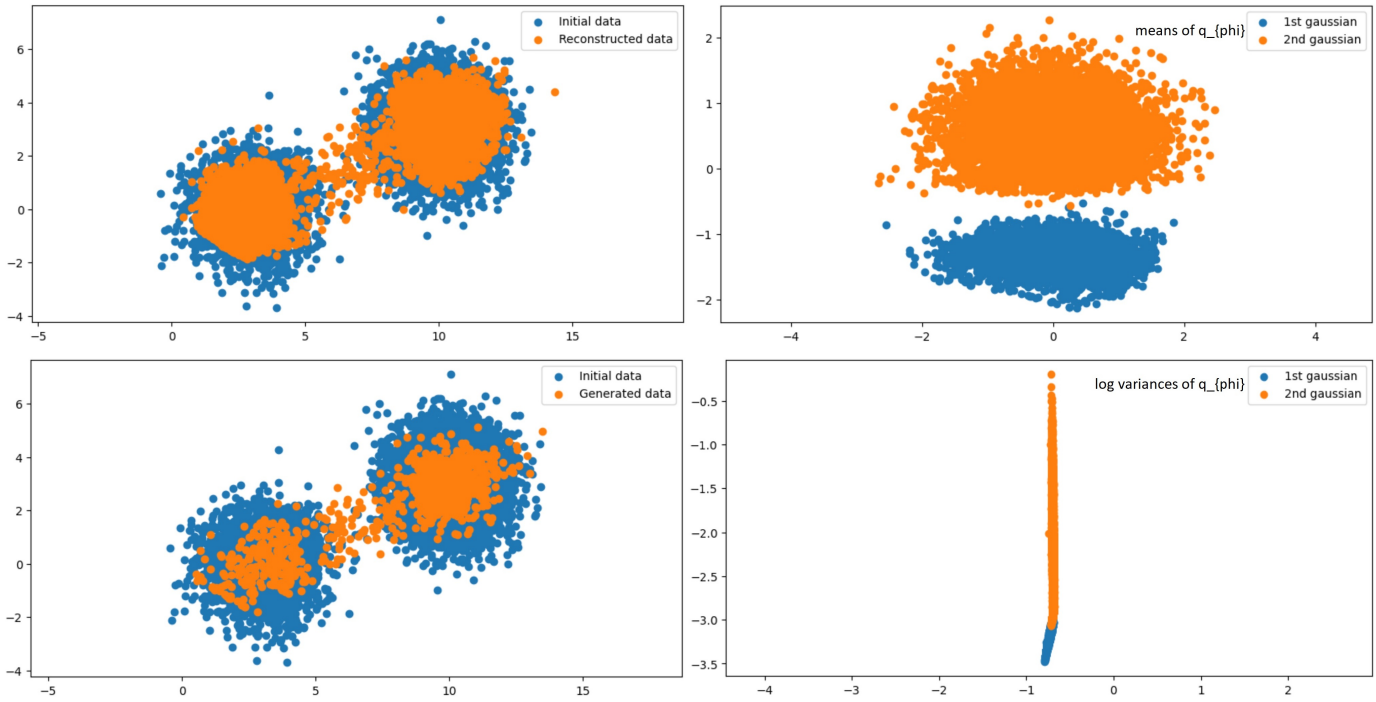

Figure 2: VAE with standard normal prior: left – visual results in data reconstruction and generation, right – latent space visualization.

References

[1] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[2] M. D. Hoffman and M. J. Johnson, "Elbo surgery: yet another way to carve up the variational evidence lower bound."

[3] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[4] J. Tomczak and M. Welling, "Vae with a vampprior," in *International Conference on Artificial Intelligence and Statistics*, 2018, pp. 1214–1223.

[5] L. Theis, A. v. d. Oord, and M. Bethge, "A note on the evaluation of generative models," *arXiv preprint arXiv:1511.01844*, 2015.

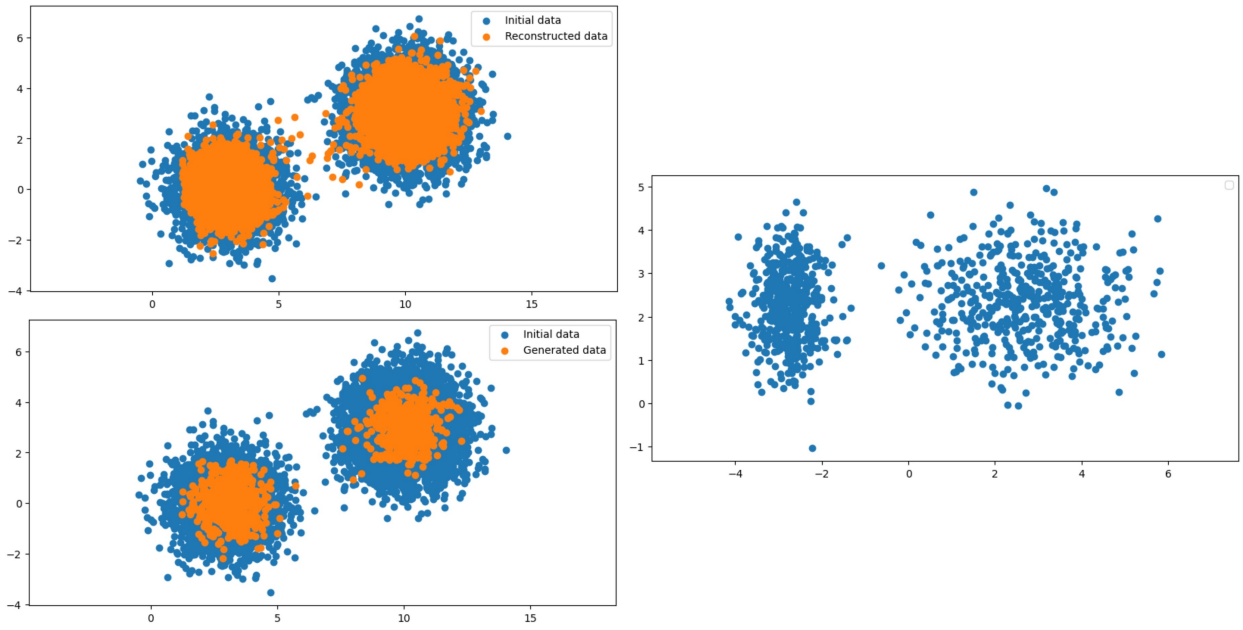

Figure 3: VAE with MoG prior (fixed $p_{\text{prior}} = 0.5$): left – visual results in data reconstruction and generation, right – trained prior visualization.

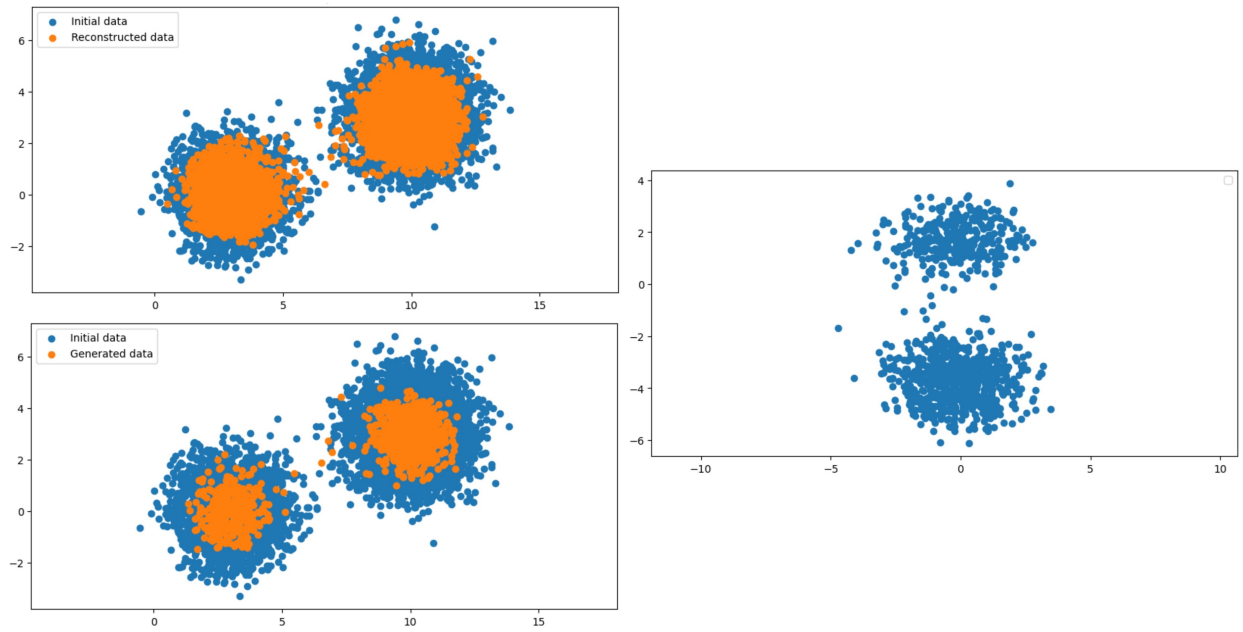

Figure 4: VAE with MoG prior: left – visual results in data reconstruction and generation, right – trained prior visualization.