Risk-Aware Variational Autoencoder

Sida (Star) Li[1], *Hao Zhu*[1]

[1]The University of Chicago

listar2000@uchicago.edu, haozhu@uchicago.edu

Abstract

Variational Autoencoders (VAEs) [1] have garnered considerable attention as powerful generative models leveraging deep learning for effective data compression and synthesis. Drawing inspiration from the robust optimization technique Conditional Value at Risk (CVaR) [2], which is witnessing escalating adoption in deep learning model training paradigms [3] [4], this paper introduces a novel class of Risk-Aware Variational Autoencoders (RA-VAEs). The aim is to optimize these RA-VAEs for either risk-seeking (best-case) or risk-averse (worst-case) scenarios. To accomplish this, we propose batch risk-awareness and subsampling risk-awareness, two innovative strategies designed to bias the reconstruction loss component, $-E_{q(z|x)}[\log p(x \mid z)]$, in the VAE's training objective towards the worst or best-case losses. Through experiments on the MNIST [5] and Fashion-MNIST [6] datasets, batch risk-awareness not only succeeds in establishing a lower bound for the worst-case performance but also enhances overall model performance by serving as a regularization mechanism. Although some model specifications did not yield as promising results, they exhibited expected tail behaviors, thereby validating the concept and laying the groundwork for future research.

Index Terms: unsupervised learning, variational autoencoder, robust optimization

1. Background

Variational Autoencoder (VAE) is a type of generative model that uses deep learning techniques to learn a compressed representation of data, while also being able to generate new data that resembles the original input. The objective function of a VAE consists of two components: the reconstruction loss and the Kullback-Leibler (KL) loss. While the KL loss acts as a regularizer that shapes the latent distribution towards a standard Gaussian, the reconstruction loss measures how well the decoder can recreate the original input from the encoded latent variable and is of main interest. In this paper, we consider evaluating a VAE model on a test mini-batch $x_1, ..., x_N \in \mathbb{R}^d$, and obtaining corresponding losses [1] $l_1, ..., l_N \in \mathbb{R}$. The VAE's best/worst-case performances on this mini-batch, parameterized by a parameter ϵ, refer to the top/bottom ϵ fraction of losses in terms of magnitude. We can define the sets

$$L_{best}(x_1, ..., x_N; \epsilon) = \{l_i \mid i = 1, ..., N,\ l_i < T_\epsilon\} \quad (1)$$

$$L_{worst}(x_1, ..., x_N; \epsilon) = \{l_i \mid i = 1, ..., N,\ l_i > T_{1-\epsilon}\} \quad (2)$$

where the threshold value T_ϵ is the ϵ quantile of all losses. Obviously, taking the average of L_{best} and L_{worst} gives a measure of "how good/bad in general can the best/worst-cases be" for the subset of best/worst performing data points. Finally, if the above evaluations are applied to every minibatch of a test dataset, we obtain an empirical distribution (mean and std) of the VAE's tail behaviors (overall best/worst-case performances).

The above definition is closely related to the concept of Conditional Value at Risk (CVaR). Also known as Expected Shortfall, CVaR is a risk measure first introduced in quantitative finance to handle risk-averse decision-making. CVaR measures the expected loss in the worst-case scenarios. Specifically, it is defined as the expected value of the loss given that the loss is beyond a certain threshold, which can be approximated through the Monte Carlo estimator, i.e. the average of elements in $L_{worst}(x_1, ..., x_N; \epsilon)$. We can similarly define a risk-seeking metric as the expected value of the loss for loss **below** a certain threshold. Thus, in the following parts, CVaR refers to both the risk-seeking and risk-averse measures if not specified otherwise.

VAE has been applied to various scenarios where the best/worst-case performances are highly valued. For instance, when cherry-picking the best images out of an array of generated outputs, we might hope for optimizing the best results we can get; in situations where the consequence of a very bad reconstruction/generation is expensive, a safety net lower bounding the loss is desired. The vanilla VAE would suffer from the "expectation problem": the mismatch between its training objective of optimizing expectations (average reconstruction loss) and the real objective of maximizing best/worst-case performances. In section 2, we propose two mechanisms to mitigate such mismatch by introducing risk-awareness into the training reconstruction loss.

1.1. Vanilla VAE

The vanilla VAE assumes the following model:

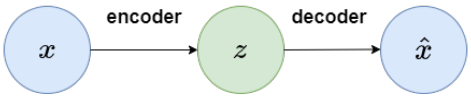

Figure 1: *Variational Autoencoder*

The weights of the networks are fitted by maximizing the evidence lower bound, defined as

$$-\mathrm{KL}(q(z \mid x) \,||\, p(z)) + \mathbb{E}_{q(z|x)}[\log p(x \mid z)] \quad (3)$$

2. Risk-Aware VAE

2.1. Batch Risk-awareness

VAE is usually trained using mini-batch gradient descent of batch size N; we also assume that for each data point x_i, only $B = 1$ subsampling is done. The reconstruction loss of x_i is then approximated by a stochastic estimator

$$-E_{q(z|x_i)}[\log p(x_i \mid z))] \approx -\log p(x_i \mid z_i) \quad (4)$$

[1]The loss metric we used in evaluating the model is consistent with the metric in reconstruction loss at training time, which in turn depends on assumptions on the form of $p(x \mid z)$.

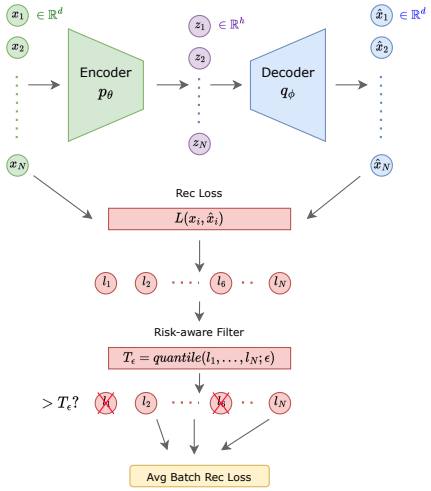

Figure 2: *Model pipeline for RA-VAE with batch risk-awareness (risk-averse). It is identical with a vanilla VAE until filtering the individual losses $l_1, ..., l_n$ based on threshold T_ϵ, and finally calculate the risk-averse average batch rec loss based on the filtered values.*

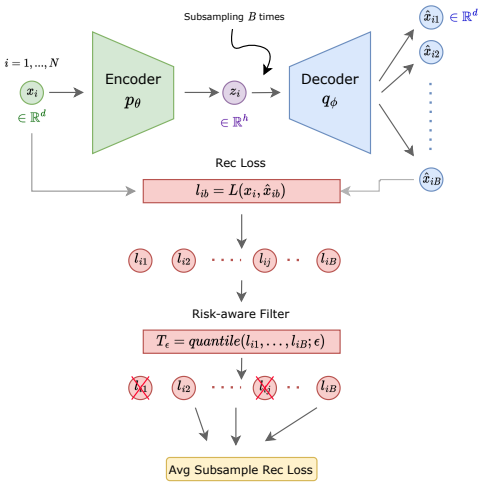

Figure 3: *Model pipeline for RA-VAE with subsampling risk-awareness (risk-averse). When subsampling B times, we have loss l_{ij} between each data point x_i and its subsample $\hat{x}_{ij}$. We then filter these losses based on threshold T_ϵ and bias the mean subsample reconstruction loss towards the filtered values.*

and the RHS expression can be written as $l_i = -\log p(x_i \mid z_i) = L(x_i, \hat{x}_i)$ for some loss L based on $p(x \mid z)$'s assumption (e.g. a Gaussian assumption corresponds to L being MSE loss). In vanilla mini-batch GD, the expected (average) reconstruction loss for the mini-batch is calculated as

$$l(x_1, ..., x_N) = \frac{1}{N} \sum_{i=1}^{N} l_i \tag{5}$$

Applying CVaR's idea of considering a conditional expectation, we can define the risk-averse and risk-seeking average reconstruction loss

$$l_{averse}(x_1, ..., x_N; \epsilon) = \frac{1}{N_a} \sum_{i=1}^{N} l_i \mathbf{1}\{l_i > T_{1-\epsilon}\} \tag{6}$$

$$l_{seeking}(x_1, ..., x_N; \epsilon) = \frac{1}{N_s} \sum_{i=1}^{N} l_i \mathbf{1}\{l_i < T_\epsilon\} \tag{7}$$

respectively, where $N_a = \sum_i \mathbf{1}\{l_i > T_\epsilon\}$, $N_s = \sum_i \mathbf{1}\{l_i < T_\epsilon\}$, and the threshold T_ϵ is the ϵ quantile of losses $l_1, ..., l_N$.

The risk-averse loss only averages over the worst-case individual losses above the threshold T_ϵ, thus it is aligned with the real objective of optimizing the worst-case performance in risk-averse problems. Similarly, the risk-seeking loss is designed to focus on evaluating the best-performing samples.

It is also worth noting that T_ϵ's dependency on the losses introduces an extra source of randomness. In practice, we can stabilize and smooth T_ϵ by running an exponential moving average with values calculated in previous minibatches [7].

2.2. Subsampling Risk-Awareness

An alternative way to introduce risk-awareness is through subsampling. That is, instead of doing only one subsample per data point x_i as usual, $B > 1$ subsamples are drawn. Now equation (4) can be more accurately approximated by

$$-E_{q(z|x_i)}[\log p(x_i \mid z))] \approx -\frac{1}{B} \sum_{j=1}^{B} \log p(x_i \mid z_{ij}) \tag{8}$$

we then define $l_{ij} = -\log p(x_i \mid z_{ij}) = L(x_i, \hat{x}_{ij})$, which is a loss function between data x_i and its j$^{\text{th}}$ subsample. For this expectation, we can again apply the idea of risk-awareness to construct the risk-averse and risk-seeking reconstruction loss for subsampling

$$l_{averse}(x_1, ..., x_N; \epsilon) = \frac{1}{N_{i,a}} \sum_{j=1}^{N} l_{ij} \mathbf{1}\{l_{ij} > T_{1-\epsilon}\} \tag{9}$$

$$l_{seeking}(x_1, ..., x_N; \epsilon) = \frac{1}{N_{i,s}} \sum_{j=1}^{N} l_{ij} \mathbf{1}\{l_{ij} < T_\epsilon\} \tag{10}$$

where $N_{i,a} = \sum_j \mathbf{1}\{l_{ij} > T_\epsilon\}$, $N_{i,s} = \sum_j \mathbf{1}\{l_{ij} < T_\epsilon\}$, and the threshold T_ϵ is the ϵ quantile of losses. The above process is repeated for every data point x_i in a minibatch.

3. Experiments

In this project, we conducted experiments for both risk-seeking and risk-averse VAE with different hyperparameters including: batch sizes, batch or subsampling risk-awareness and T_ϵ. To test for the effectivenss of risk-awareness, comparisons were made on a before and after basis: a vanilla VAE is compared with a RA-VAE where the only modification is the implementation of reconstruction loss. The data sets used are MNIST and Fashion-MNIST. Experiments were run on Google Colab and our personal computers.

3.1. Data and implementation

For MNIST, the training set consists of 48,000 samples, the validation set consists of 12,000 samples and the test set consists of 10,000 samples. Similarly for Fashion-MNIST, the sizes for the

training set, validation set and test set are 48,000, 12,000 and 10,000 respectively. Batch size is set to 256 across all reported experiments. The entire project was implemented in PyTorch.

3.2. Choice of evaluation metric

The Inception Score (IS) [8] and Fréchet inception distance (FID) [9] are popular metrics for evaluating the quality of generated images. Even though these qualities might correlate well with human judgement empirically, as Barratt and Sharma pointed out, using these metrics on data sets that are not ImageNet is a common pitfall [10]. Both IS and FID make use of the Inception model, which was trained on the ImageNet data set, to extract features from the generated images. So, using these metrics for our analysis requires some fine tuning of the Inception model, which may not be appropriate given the available computation resources and time constraint.

Xu et al. have shown that the 1-Nearest Neighbor classifier and Kernel Maximum Mean Discrepancy are good sample-based metrics for evaluating generated image qualities [11]. However, our project set out to show that RA-VAEs could achieve better extreme case performances. For example, a risk-seeking model might allow us to generate a few high quality images while sacrificing the average-case quality. Since we are interested in comparing the extreme case performances instead of batch-wise average performances, these metrics are not well-suited for our goals.

Gu et al. pointed out that a possible way to evaluate the quality of a single image is to train a Gaussian mixture model (GMM) on the real images and assess the probability of observing the generated images given the trained GMM,

$$p(\mathbf{x} \mid \lambda) = \sum_{i=1}^{M} w^i g(\mathbf{X} \mid \mu^i, \Sigma^i), \tag{11}$$

where w^i is the mixture weights and $g(\mathbf{X} \mid \mu^i, \Sigma^i)$ is the component Gaussian densities [12]. We notice that the form of this evaluation metric is in fact similar to VAE's reconstruction loss.

As a result, we decided on using the reconstruction loss and human evaluation, which are both easy to implement and well-defined for image by image comparisons.

3.3. Reconstruction loss

With an evaluation metric in mind, we begin by analyzing the reconstruction loss. In this section, we would like to highlight some interesting findings with risk-averse models. With these models, the worst case performances between the vanilla model and the risk-averse model are compared. That is, we measure the reconstruction loss by considering the worst performing losses on each batch. Figure 4 displays the effects of batch risk-averseness on the MNIST data set. We note that the results we got from subsampling risk-averseness were qualitatively similar. Each plot in Figure 4 corresponds to a specific loss threshold T_ϵ that was used when risk-averse VAE was trained. On the x-axis, we have the percentages of worst performing losses considered at evaluation time, while the y-axis represents the mean squared validation error. The blue dots represent the averages of the worst losses for the vanilla model. And the orange dots represent the averages for the risk-averse model. The error bars indicate one standard deviation of the losses.

Overall, risk-averse VAE consistently outperforms the vanilla VAE in terms of worst-case performance, except in scenarios where only a small fraction of the data was used during training. As we increase the threshold parameter ϵ to 0.5,

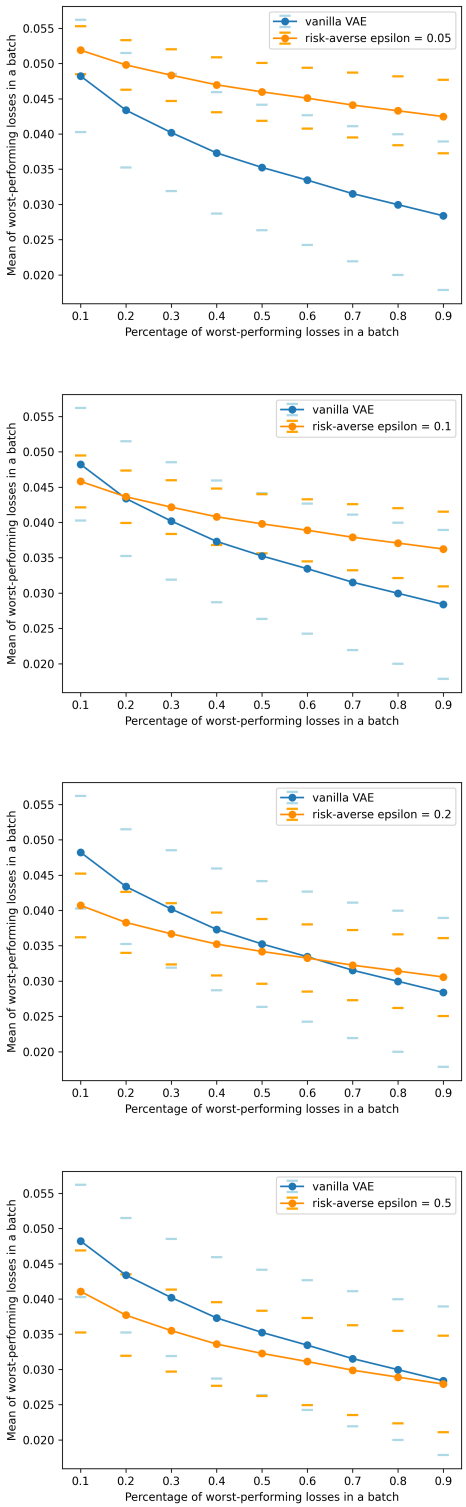

Figure 4: *Reconstruction loss as measured by the worst losses. The models compared are the vanilla VAE (in blue) and the risk-averse VAE (in orange) with batch risk-awareness. Each plot above corresponds to a specific loss threshold T_ϵ used during training. The x-axis's are the percentages of worst performing losses used at evaluation time. The y-axis's are mean squared validation error on the MNIST data set.*

the risk averse VAE seems to demonstrate dominance over the vanilla model. On the other hand, for risk-seeking VAE, although it also exhibits tail behavior where the model does best on select samples, it did not get as impressive results as risk-averse VAE did. A possible explanation for this observation is that the risk-averse VAE was able to better learn by focusing on the most difficult images while risk-seeking VAE could not do the same by focusing on the easiest images.

3.4. Reconstructing distorted images

The previous experiment appears to show that batch risk-averse VAE could outperform the vanilla VAE with an appropriate T_ϵ. Since reconstruction loss may not be perfectly correlated with the quality of reconstructed images, we further tested risk-averse VAE's ability in reconstructing images by visualization. Figure 5 reports some results. The images are taken from the Fashion-MNIST data set and distorted with an elastic transformation. The resulting images still look realistic enough to be something that the model might encounter in practice. We show the results on these images since they are more challenging than MNIST. When the model was tested on MNIST, we observed more subtle differences. On Figure 5, the first two images on each row are the original and distorted images. The third image is the reconstruction by vanilla VAE and the rest are reconstruction by the risk averse VAE with different T_ϵ. Again, we observe that except when ϵ is too small, risk averse VAE seems to demonstrate an ability to produce sharper reconstructions.

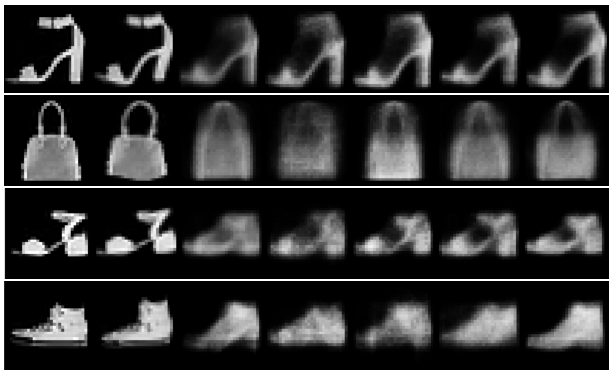

Figure 5: ***First row:*** *reconstructed images of a high-heeled shoe.* ***Second row:*** *reconstructed images of a bag.* ***Third row:*** *reconstructed images of a sandal.* ***Fourth row:*** *reconstructed images of a boot.* ***From left to right:*** *the input image, the distorted input image, reconstruction of the distorted image by the vanilla model, reconstruction of the distorted image by the risk averse VAE with* $\epsilon = 0.05, 0.1, 0.2, 0.5$.

3.5. Downstream classification

Given the results of the previous experiments, a natural question to ask is why is the risk-averse model showing better generalization performance. One hypothesis is that the risk-averse model was able to better learn an embedding of the images that allows it to do well even on distorted images. To test this hypothesis, we tested the models' performance on a downstream classification task. The MNIST data set was used for this experiment. A logistic regression classifier was trained on the models' extracted embeddings of the training images and evaluated on the extracted embeddings of the validation images. The RA-VAE's were trained with subsampling risk-awareness and $\epsilon_1 = 0.05, \epsilon_2 = 0.1$ and $\epsilon_3 = 0.2$. Table 1 reports the results of the experiment. Comparing the performance of risk-averse VAE with the vanilla model, we observe that the risk-averse model could not achieve as good performance while the risk-seeking models achieved similar performance. These numbers seem to suggest that the encoder part of risk-averse models may not be the suitable for classification tasks. Rather, it's the whole model pipeline that is leading to the promising results observed in previous experiments.

Model	**Error Rate**
Vanilla VAE	0.146
Risk Averse VAE ($\epsilon = 0.05$)	0.515
Risk Averse VAE ($\epsilon = 0.1$)	0.494
Risk Averse VAE ($\epsilon = 0.2$)	0.451
Risk Seeking VAE ($\epsilon = 0.05$)	0.149
Risk Seeking VAE ($\epsilon = 0.1$)	0.146
Risk Seeking VAE ($\epsilon = 0.2$)	0.148

Table 1: *Downstream classification performance on the MNIST data set using different RA-VAE models.*

4. Conclusions

We have successfully demonstrated the effectiveness of a batch risk-averse VAE. This success suggests that with fine-tuning of ϵ and other training configurations, the risk-averse VAE can function as a regularization mechanism.

While some models, such as the batch risk-seeking VAE, have not generated particularly remarkable results, they have still displayed anticipated tail behaviors. Given that the risk-seeking VAE has performance comparable to the vanilla VAE in downstream classification tasks, further exploration of areas where risk-seeking VAE excels would be a productive avenue for future research.

Some promising next steps of this research topic include:

- Simultaneously incorporate both batch risk-awareness and subsampling risk-awareness into VAE. This endeavor will likely necessitate meticulous hyperparameter tuning across an extensive range, allowing for more comprehensive model optimization.
- Using alternative thresholds other than T_ϵ for the conditional expectation, or utilizing different smoothing methods to stabilize the existing threshold.
- Evaluating performance in more downstream tasks such as clustering and image generation.

5. Division of labor

Both authors contributed equally to the project. Sida mainly worked on the theory and implementation of the risk-aware VAE models. Hao mainly worked on the evaluation strategies.

6. Acknowledgements

We sincerely thank Professor Karen Livescu and TA David Yunis for organizing this amazing course and providing valuable feedback for our project proposal and update. We also appreciate our classmates Marziyeh Movahedi, Richard Xu, Qizhong Zhang, Naren Manoj and Younghun Lee for offering their insights and questions to our work in progress. Finally we thank TTIC for giving us the resource and space for discussing the project.

7. References

[1] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2022.

[2] R. T. Rockafellar, S. Uryasev *et al.*, "Optimization of conditional value-at-risk," *Journal of risk*, vol. 2, pp. 21–42, 2000.

[3] B. K. Petersen, M. Landajuela, T. N. Mundhenk, C. P. Santiago, S. K. Kim, and J. T. Kim, "Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients," *arXiv preprint arXiv:1912.04871*, 2019.

[4] A. Rajeswaran, S. Ghotra, B. Ravindran, and S. Levine, "Epopt: Learning robust neural network policies using model ensembles," *arXiv preprint arXiv:1610.01283*, 2016.

[5] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[6] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," 2017.

[7] R. D. Harris and C. Guermat, "Robust conditional variance estimation and value-at-risk," *Available at SSRN 254569*, 2000.

[8] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," 2016.

[9] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[10] S. Barratt and R. Sharma, "A note on the inception score," 2018.

[11] Q. Xu, G. Huang, Y. Yuan, C. Guo, Y. Sun, F. Wu, and K. Weinberger, "An empirical study on evaluation metrics of generative adversarial networks," 2018.

[12] S. Gu, J. Bao, D. Chen, and F. Wen, "Giqa: Generated image quality assessment," 2020.