Generating Realistic Financial Losses

Danilo Marinho Fernandes, Federico Testa

Contents

1 Introduction

In the past twenty years, the rise in unexpected shocks and financial crises has posed a significant concern for financial risk management teams. Stress tests have become of primary importance for regulators to gauge the banking system's ability to withstand diverse risks (like market fluctuations, credit uncertainties, operational issues, climate changes, and more). A key challenge they face is simulating adverse yet plausible negative returns to anticipate and hedge against potential financial losses. This challenge, which represents an unsupervised learning problem, revolves around two main aspects: firstly, determining a suitable method to generate extreme values from given input noise, and secondly, constructing a generative model using a dataset that contains limited instances of extreme data and is yet able to replicate them.

1.1 Objective

Given an input noise $Z \in \mathbb{R}^N$, our goal is to build a generative model G_θ (parametrized by θ) that can simulate samples

$$G_\theta(Z) = \tilde{X} \in \mathbb{R}^d$$

that accurately reproduce the distribution of the real (negative) financial log-returns $X \in \mathbb{R}^d$.
We will discuss the choice of the distribution of the noise and of the size of the latent space $\mathbb{R}^N$ below.

1.2 The Dataset

We consider $\left\{X^{(i)} = \left(X_1^{(i)}, X_2^{(i)}, X_3^{(i)}, X_4^{(i)}\right) : i \in \{1, 2, ..., n\}\right\}$, $n = 746$, i.i.d. observations of 4-dimensional vectors representing simultaneous negative financial log-returns of $d = 4$ financial assets. Data are actually transformed to positive values, for manipulation purposes.

2 Evaluation Metrics

To evaluate the ability of the model to reproduce effectively the extreme values of the target distribution, we rely on two criteria:

- the Anderson-Darling distance, which focuses on measuring the accuracy of the generated marginals compared to the target ones;

- the Absolute Kendall Error, which instead focuses on evaluating the similarity of the dependence structure of the generated random vector compared to the observed one.

Intuitively, the idea is that we can evaluate the performance of a generative model based on both its ability to mimic accurately the marginals of the observed data - choosing a measure that gives particular attention to a good approximation of the heavy tails - and its ability to produce a similar dependence structure to that of the original data between those marginals.

2.1 Anderson-Darling Distance

Let us recall that given a sample $X_1, ..., X_n$ its empirical cumulative distribution function (or c.d.f.) is defined as

$$\widehat{F}_n(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left\{X_i \leq x\right\}$$

The Anderson-Darling distance (A.D. distance) has the following expression:

$$W_n = n\int_{-\infty}^{\infty} \frac{\left(\widehat{F}_n(x) - F(x)\right)^2}{F(x)(1-F(x))} dF(x)$$

which consists essentially in a weighted square difference between the theoretical cumulative distribution function of the considered distribution and the empirical c.d.f. of an i.i.d. sample with that same distribution. This quantity measures the distance between the two distributions but emphasizing, thanks to the choice of the weights, the similarity of the tails. It is therefore well suited for our purpose.
In our setting, we refer to $\widehat{F}^\tau_n$ as the empirical distribution function associated to $X^\tau_1, \ldots, X^\tau_n$, and to $X^\tau_{1,n} \leq \cdots \leq X^\tau_{n,n}$ as the order statistics for each financial ticker $\tau = 1, \ldots, d$. For a generated variable $\tilde{X} = G(Z)$, for a specific ticker τ, we define the following

$$\tilde{u}^\tau_{i,n} = \frac{1}{n+2}\left(\sum_{j=1}^{n} \mathbb{1}\left\{X^\tau_j \leq \tilde{X}^\tau_{i,n}\right\} + 1\right) \approx \widehat{F}^X_n(\tilde{X}^\tau_{i,n}) \approx \mathbb{P}_{X\sim\mu}\left(X \leq \tilde{X}^\tau_{i,n}\right)$$

which represents the model probability of a generated variable (with small corrections to avoid having $\tilde{u}^\tau_{i,n} = 0, 1$).
The way we compute the Anderson-Darling distance for each ticker τ is then

$$W^\tau_n = -n - \frac{1}{n}\sum_{i=1}^{n}(2i-1)\left(\log\left(\widetilde{u}^\tau_{i,n}\right) + \log\left(1 - \widetilde{u}^\tau_{n-i+1,n}\right)\right).$$

and the global metric on the marginals is taken as the average of the distances for all tickers

$$\mathcal{L}_M = \frac{1}{d}\sum_{\tau=1}^{d} W^\tau_n.$$

It's clear that this quantity measures only the similarity between the marginals of the generated data and of the observed ones, without accounting for the dependence structure of the distributions. For further details on this metric and the reformulation of W^τ_n above, see [1] and [2].

2.2 Absolute Kendall Error

Contrary to the A.D. distance, the absolute Kendall error (A.K.E.) can be seen as a measure of similarity of the dependence structure of the marginals of a multivariate distribution. The formulation used below stems from the Kendall's dependence function (see [3]), i.e. a univariate c.d.f. characterizing the dependence structure of a copula C with uniform marginals $U^{(i)}$ on $[0, 1]$ (while being able to freely choose the latent distribution):

$$K_C(t) = \mathbb{P}\left[C\left(U^{(1)}, U^{(2)}, ..., U^{(d)}\right) \leq t\right]$$

We implemented an estimation of Kendall's dependence function based on the following pseudo-observations on the observed data, in order to comply with the requirement of working with uniform margins in the unit interval:

$$Z_i = \frac{1}{n-1}\sum_{j\neq i}^{n} \mathbb{1}\left\{X^1_j < X^1_i,\, X^2_j < X^2_i,\, X^3_j < X^3_i,\, X^4_j < X^4_i\right\}$$

and the analogous on the simulated data

$$\tilde{Z}_i = \frac{1}{n-1} \sum_{j \neq i}^{n} \mathbb{1} \left\{ \tilde{X}_j^1 < \tilde{X}_i^1, \, \tilde{X}_j^2 < X_i^2, \, \tilde{X}_j^3 < \tilde{X}_i^3, \, \tilde{X}_j^4 < \tilde{X}_i^4 \right\}$$

To compare them, we compute the (normalized) L^1 norm of the ordered vectors of pseudo observations:

$$\mathcal{L}_D = \frac{1}{n} \sum_{i=1}^{n} \left| Z_{i,n} - \tilde{Z}_{i,n} \right|$$

where $Z_{1,n} \leq Z_{2,n} \leq ... \leq Z_{n,n}$ are the order statistics, and likewise for $\tilde{Z}_{j,n}$.

3 Models

The first model we used to tackle the problem is a Generative Adversarial Network (GAN), which is known to be very effective at replicating distribution in different tasks, but also very unstable in the training and difficult to tune. The result was not completely satisfying, so we opted for a more stable version known as Wasserstein GAN (WGAN), detailed below. We also looked for implementations of GAN that could be particularly suitable for extreme value estimation, similar to the EV-GAN (see [4]), but with unstable results. Finally, we implemented a simple yet very effective model based on the minimization of an "empirical energy distance".

3.1 GAN

GANs, introduced in [5], are powerful generative models made of two "opposing components": a generator and a discriminator. The generator is a neural network that maps the latent space into the space of the observations, and aims turning random noise into samples that follow the same distribution of the training data. The discriminator instead is neural network that performs a binary classification and is tasked with learning how to recognize the "fake" samples of the generator from the "true" ones of the training data. The idea is to tune the networks to create a virtuous cycle: the discriminator should get progressively better at discerning the two kind of observations while the generator should improve its "deceptive" abilities and create samples that are closer and closer to the target distribution.

Due the instability in the optimization of GANs and the risk of mode-collapse, we opted for a different formulation that can be interpreted in a similar framework but that was showing more promising results and proved to be more stable: Wasserstein GAN.

3.2 Wasserstein GAN

Wasserstein GAN, introduced in [6], operates in a similar way to GAN, although the problem is tackled a slightly different perspective. The idea is simply to generate data from a latent space and train a generator (a neural network that takes inputs from the latent space and outputs values in the space of the observed sample) with the goal of minimizing the Wasserstein-1 distance between the distribution of the generated data and that of the real one.

Wasserstein-1 distance between two distributions μ and ν on $\mathbb{R}^d$ is defined as

$$W(\mu, \nu) = \inf_{\gamma \in \Pi(\mu, \nu)} \mathbb{E}_{(x,y) \sim \gamma}[\|x - y\|]$$

where $\Pi(\mu, \nu)$ denotes the set of all joint distributions $\gamma(x, y)$ whose marginals are respectively μ and ν. Relying on this definition, the distance is intractable. However the same article [6] provides a dual formulation that allows us to implement a two-step procedure for approximating it and minimizing it effectively. Indeed, the distance can be rewritten as

$$W(\mu, \nu) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{X \sim \mu}[f(X)] - \mathbb{E}_{X \sim \nu}[f(X)]$$

where the supremum is over all the real-valued 1-Lipschitz functions. Note that if we replace $\|f\|_L \leq 1$ for $\|f\|_L \leq K$ (consider K-Lipschitz for some constant K), then we end up with $K \cdot W(\mu, \nu)$ (i.e.: it's equivalent up to a multiplicative constant, so we should only care that the Lipschitz constants of the considered functions are in a bounded set if we are interested in minimization).

In the context of our problem, denoting by $\mathbb{P}_X$ the distribution of the real data, by $\mathbb{P}_Z$ the distribution of the latent data and by θ the parameters of the generator G_θ, we would need to find the values of θ minimizing

$$W(\theta) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{X \sim \mathbb{P}_X}[f(X)] - \mathbb{E}_{\mathbb{P}_Z}[f(G_\theta(Z)]$$

Under some technical assumptions (i.e. assumption 1 in [6]), the supremum above is actually a maximum, and the gradient w.r.t. commutes with the second expectation. This would allow us to implement a gradient-based optimization method, assuming we know the Lipschitz function that maximizes that difference. WGAN approximates said function with the output of a neural network called the ”critic”, which represents the parallel of the discriminator. This procedure is sound, as long as some clipping is performed on the weights of the critic (to enforce the Lipschitz condition, whose constant is unimportant for the maximization of the distance using gradient-based methods). The pseudo-code for this model is the following:

```
Algorithm: Wasserstein Generative Adversarial Network
Input: $\alpha$: learning rate, $c$: clipping parameter, $m$: batch size, $n_{\text{critic}}$: iterations of critic per
       generator iteration, $w_0$: initial critic parameters, $\theta_0$: initial generator parameters
for $k=1,\ldots,n_{epochs}$ do
    for $t = 1, \ldots, n_{critic}$ do
        Sample $x_r = \left\{x_r^{(i)}\right\}_{i=1+rm}^{(r+1)m} \sim \mathbb{P}_X$ (batch from real data);
        Sample $\left\{z^{(i)}\right\}_{i=1}^{m} \sim \mathbb{P}_Z$ (batch of latent samples);
        $g_w \leftarrow \nabla_w \left[\frac{1}{m}\sum_{i=1}^{m} f_w\left(x^{(i)}\right) - \frac{1}{m}\sum_{i=1}^{m} f_w\left(G_\theta\left(z^{(i)}\right)\right)\right]$;
        $w \leftarrow w + \alpha \cdot \text{RMSProp}\left(w, g_w\right)$;
        $w \leftarrow \text{clip}(w, -c, c)$;
    end
    Sample $\left\{z^{(i)}\right\}_{i=1}^{m} \sim \mathbb{P}_Z$ (batch of latent samples);
    $G_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{i=1}^{m} f_w\left(G_\theta\left(z^{(i)}\right)\right)$;
    $\theta^k \leftarrow \theta^{k-1} - \alpha \cdot \text{RMSProp}\left(\theta, g_\theta\right)$;
end
```

where f_w is the function describing the critic output, w are the parameters of the critic network. The implementation is very similar to that of a ”standard” GAN, the key difference is the output of the critic, which is a probability instead of a binary label.

3.2.1 Attempt with Extreme Value Theory Modelling

Along with WGAN and the standard GAN, we tried implementing a simplified version of the EV-GAN, presented in [4], which focuses on accurate estimation of extreme values. Along with that, we tried to use both EV-GAN and WGAN and make the model learn a combination of the outcomes. However, the training turned out to be really unstable and not as promising as that of WGAN (in terms of the two evaluation metrics introduced before) in the first case, whereas in the second it learnt to only use the outcomes of WGAN. This has probably to do with an improvable implementation of the regularity techniques presented in [4].

3.3 Energy Distance Model

In an attempt to create a simple benchmark model, but also out of genuine curiosity, we implemented a generative method that is based on the minimization of the ”empirical” energy distance.
Given two distributions μ and ν on $\mathbb{R}^d$ (with probability moments of finite order α) and their characteristic functions ϕ_μ, ϕ_ν, the (α-)energy distance is defined as:

$$\mathcal{E}^\alpha(\mu, \nu) = \frac{1}{C(d, \alpha)} \int_{\mathbb{R}^d} \frac{|\phi_\mu(t) - \phi_\nu(t)|^2}{|t|^{d+\alpha}} dt$$

with $C(d, \alpha) = 2\pi^{d/2} \frac{\Gamma(1+\alpha/2}{\alpha 2^\alpha \Gamma((d+\alpha)/2)}$.

We have however a simpler formulation for a metric version of the energy distance on $\mathbb{R}^d$ (endowed with the metric induced by the euclidean norm):

$$\mathcal{E}(\mu, \nu) = \mathbb{E}\left[2\|X - Y\| - \|X - X'\| - \|Y - Y'\|\right]$$

with $X, X' \sim \mu$ and $Y, Y' \sim \nu$ (independent).
Therefore, given two independent samples $X_1, ..., X_n$ and $\tilde{X}_1, ..., \tilde{X}_{\tilde{n}}$, we can compute an empirical version of the energy distance between their distributions as

$$\hat{\mathcal{E}}(X, \tilde{X}) = \frac{2}{n\tilde{n}} \sum_{i,j} X_i \tilde{X}_j - \frac{1}{n^2} \sum_{i,j} X_i X_j - \frac{1}{\tilde{n}^2} \sum_{i,j} \tilde{X}_i \tilde{X}_j$$

with a statistical error of the order $\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{\tilde{n}}}$.
Given this formulation, we can implement a generative procedure that samples data from a latent space, transforms them into generated data using a neural network and then trains the network to minimize the distance $\hat{\mathcal{E}}$ between the generated data and the "true" ones.
Notice that in this procedure the assumption of independence of the considered samples is actually violated after the first epoch. Indeed, during the first epoch at each iteration over a batch the generated data, output of the network trained on the previous batches, will be independent of the current batch of "real" data. However, in the following epochs the network will have been updated using the whole training dataset and will therefore be a function of the training data, and thus the condition of independence of the generate data from the training sample will not properly hold. Despite this issue (and despite its simplicity) the model has proven to be extremely effective and has provided results that compare with those of WGAN across all the metrics and analysis we have performed.

```
Algorithm: Energy Distance Model (EDM)
Input: $x = (x^{(1)}, ..., x^{(n)})$: training data, $\alpha$: learning rate, $m$: batch size, $n$: train data size,
       $\theta_0$: initial network parameters
for $k$=0,...,$n_{epochs} - 1$ do
    for $r$=0,...,$\frac{n}{n_{epochs}} - 1$ do
        Sample $x_r = \left\{ x_r^{(i)} \right\}_{i=1+rm}^{(r+1)m} \sim \mathbb{P}_X$ (batch from real data);
        Sample $z_r = \left\{ z_r^{(i)} \right\}_{i=1}^{m} \sim \mathbb{P}_Z$ (batch of latent samples);
        $G_\theta \leftarrow -\nabla_\theta \hat{\mathcal{E}}(x_r, G_\theta(z_r))$;
        $\theta^{k+1} = \theta^k - \alpha \, \mathrm{Adam}\,(\theta, G_\theta)$;
    end
end
```

4 References

[1] T.W Anderson and D.A. Darling. Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes. *The annals of mathematical statistics*, 193-769212, 1952.

[2] T.W Anderson and D.A. Darling. A test of goodness of ft. *Journal of the American statistical association*, 765-769, 1954.

[3] C. Genest and L.-P. Rivest. Statistical inference procedures for bivariate Archimedean copulas. *Journal of the American Statistical Association*, 88(423):1034-1043, 1993.

[4] M. Allouche, S. Girard, E. Gobet. EV-GAN: Simulation of extreme events with ReLU neural networks *Journal of Machine Learning Research*, 23(150):1-39, 2022.

[5] Ian J. Goodfellowm, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio. Generative Adversarial Networks, *arXiv*, 1406.2661, 2014

[6] M. Arjovsky, S. Chintala and L. Bottou, Wasserstein GAN, *arXiv*, 1701.07875, 2017

[7] T. Huster, J. EJ Cohen, Z. Lin, K. Chan, C. Kamhoua, N. Leslie, CY. J. Chiang, and V. Sekar. Pareto GAN: Extending the representational power of GANs to heavy-tailed distributions. *arXiv*: 2101.09113, 2021.