Drawdown as a non-linear dynamic system

Emiel Lemahieu[1,2,3]

Ghent University, InvestSuite

Abstract

The drawdown ξ_t of a portfolio value path S at time t is the path function that compares the running maximum of S at t with the current S_t. There are two ways to look at ξ_t. Either one defines ξ_t as the exact non-linear function Ξ of S_t, or one investigates the effect of a change dS_t to the drawdown $d\xi_t$, called the physics $f(\xi_t)$. The former approach requires Ξ to be continuously differentiable in a context where one wants to extract gradient information from a path (e.g. in machine learning on paths). The latter approach only requires f to have a sufficient level of smoothness or regularity (e.g. $Lip(K)$) in order for one to (1) linearly approximate a $\hat{\Xi}$ by means of the path signature $\Phi_M(S)$, (2) extract gradient information from $\hat{\Xi}$ as a function of S (through linear approximation with $\Phi_M(S)$). This paper discusses drawdown as a non-linear dynamic system in detail, reiterates the most important properties of path signatures, and proposes a linear signature-based drawdown approximator which yields a differentiable approximation $\hat{\Xi}$ for e.g. machine learning applications. A simple application is proposed for a generative market model, where the synthetic paths are evaluated in terms of $\hat{\Xi}$.

Keywords: Path functions, Signatures, Drawdowns
MSC: 91B28, 91B84

1. Introduction

Portfolio drawdown ξ_t - or the 'peak-to-valley loss' - is a popular risk measure in quantitative finance. It is defined as the difference between the

[1]Ghent University, Sint-Pietersplein 6, 9000 Gent, Belgium
[2]InvestSuite, `www.investsuite.com`
[3]Corresponding author at emiel.lemahieu@ugent.be

running maximum portfolio value $\max_{k \leq t}(S_k)$ and S_t, and it is non-negative, i.e. when a new running maximum is achieved the risk is zero $\xi_t = 0$. For a portfolio value path $S : [0, T] \to \mathbb{R}$ define drawdown function Ξ as:

$$\xi_t = \Xi(S)_t = \max(\max_{k \leq t}(S_k) - S_t, 0) \tag{1}$$

Clearly Ξ is non-linear - by the max operators - and not continously differentiabe - the max operator is not differentiable. However, the maximum function is uniformly continuous and Lipschitz-1 smooth. When taking two bounded[4] paths, the distance between their maximum value is bounded by a norm on the distance between their path values:

$$|\max_{0<i<T}(S_i^1) - \max_{0<i<T}(S_i^2)| \leq \max_{0<i<T}|S_i^1 - S_i^2| = ||S^1 - S^2||_\infty \tag{2}$$

or the maximum is Lipschitz-K continous with distance the inf-norm and $K = 1$, i.e. $Lip(1)$. By iteration of this idea: (1) $\max_{k \leq t}(S_k)$ is not differentiable but $Lip(1)$ smooth. (2) $\max_{k \leq t}(S_k) - S_t$ is an 'untruncated' drawdown path that as a combination of a $Lip(1)$ and a continuous path is itself a $Lip(1)$ path. (3) The final max operator truncates the drawdown to zero, which corresponds to the well-known rectified unit (ReLU) function, which is not differentiable but a $Lip(1)$ function[5].

This non-differentiability but regularity property naturally leads us to think of ξ_t as a controlled differential equation (CDE), rather than a direct evaluation of Ξ. The reason is mainly that, from rough path theory, an approximation of ξ_t exists that does not rely on its differentiability, but on a smoothness assumption. The intuition is that we can evaluate the impact of intervals of S on the non-linear response on ξ_t, even when linear segments dS_t do not cause a linear effect on $d\xi_t$ (i.e. Taylor approximation). Moreover, the linear approximation is a differentiable expression of ξ_t.

2. Controlled differential equations and path signatures

Considering the above, we are generally interested in the effect of the driving signal, the portfolio path S_t, on the state of our risk measure ξ_t, as expressed by the following CDE:

$$d\xi_t = f(\xi_t)dS_t \tag{3}$$

[4] I.e. the path's variance is finite.

[5] This can be seen easily from $\max(x, 0) = \frac{1}{2}(x - |x|)$.

The main idea behind rough path theory is to stop trying to evaluate ξ_t in terms of the solution $\Xi(S)_t$ by integrating $d\xi_t/dS_t$ over the smooth path S_t, assuming $f(\xi_t)$ is differentiable in a Taylor series-like approach, but to **solve ξ_t by iterating what happens over intervals of S**, only assuming f has some regularity[6](e.g. Lipschitz continuity).

Let us first take two steps back and introduce signatures in the context of a general CDE and give some examples. We are generally interested in a CDE of the form:

$$dY_t = g(Y_t)dX_t \tag{4}$$

where X is a path on $[0,T] \to \mathbb{R}$, called the driving signal of the dynamic system. g is a $\mathbb{R} \to \mathbb{R}$ mapping called the physics that models the effect of dX_t on the response dY_t.

A series of coefficients of the path that naturally arrives from this type of equation is the series of iterated integrals of the path, or the path signature Φ. The signature of a path $X : [0,T] \to \mathbb{R}$ can be defined as the sequence of ordered coefficients:

$$\Phi(X) = (1, \Phi_1, ..., \Phi_n, ...) \tag{5}$$

where for every integer n (order of the signature):

$$\Phi_n(X) = \int ... \int_{u_1 < ... < u_n, u_1, ..., u_n \in [0,T]} dX_{u_1} \otimes ... \otimes dX_{u_n} \tag{6}$$

where we define the n-fold iterated integral as all the integrals over the n ordered intervals u_i in [0,T][7]. The signature is the infinite collection for $n \to \infty$, although typically lower level M truncations are used.

$$\Phi^M(X) = (1, \Phi_1, ..., \Phi_M) \tag{7}$$

The signature lives in the so-called tensor algebra of the path space of X. The textbook analogy is that if a function on a path or interval lives in the space of letters, the signature lives in the space of words. Indeed, a signature of level n lives in the space of all the words of length n that can be formed with these letters. Given the tensor products in Eq. (6), it is a non-commutative

[6] Actually, K-Lipschitz smooth paths can be extended to so-called p-geometric *rough* paths[1], see [2] for an introduction to regularity structure.

[7] For more information on k-fold iterated integrals and the definition of u_i see the primer on emiellemahieu.github.io .

summary of the nesting of intervals (from coarse-grained to finer-grained intervals) of a path, that allows us to evaluate the nesting of their effects of intervals on the output.

To make the latter comprehensible, we summarize below how signatures have a natural nascence in CDEs and Picard iterations, and provide a simple example for a linear physics g evolving over a linear path X.

The idea behind a Picard iteration is to define for:

$$dY_t = g(Y_t)dX_t \tag{8}$$

a sequence of mapping functions $Y(n) : [0, T] \to \mathbb{R}$ recursively such that for every $t \in [0, T]$:

$$Y(0)_t \equiv y_0 \tag{9}$$

$$Y(1)_t = y_0 + \int_0^t g(y_0)dX_s \tag{10}$$

$$Y(n+1)_t = y_0 + \int_0^t g(Y(n)_s)dX_s \tag{11}$$

Now by simple recursion one finds that:

$$Y(n)_t = y_0 + \sum_k^n g^{\otimes k} \int ... \int_{u_1 < ... < u_n, u_1, ..., u_n \in [0,T]} dX_{u_1} \otimes ... \otimes dX_{u_n} \tag{12}$$

Such that an alternative solution for Y_t (next to direct path integration of g) would be:

$$Y_t = y_0 + \sum_k^\infty g^{\otimes k} \int ... \int_{u_1 < ... < u_k, u_1, ..., u_k \in [0,T]} dX_{u_1} \otimes ... \otimes dX_{u_k} \tag{13}$$

This result shows how the signature, as an iterative representation of a path over ordered intervals, naturally arises from solving CDEs using Picard iterations, and how it is a natural generalization of Taylor series on the path space when the physics is linear.

To comprehend the latter, just inductively define:

$$g^{\circ 1} = g \tag{14}$$

$$g^{\circ n+1} = D(g^{\circ n})g \tag{15}$$

then it is natural to define the N-step Taylor expansion for Y_t by $\hat{Y}(N)_t$ as:

$$\hat{Y}(N)_t = y_0 + \sum_{n=1}^{N} g^{\circ n}(y_0) \int ... \int_{u_1<...<u_n, u_1,...,u_n \in [0,T]} dX_{u_1} \otimes ... \otimes dX_{u_n} \quad (16)$$

Clearly, $\hat{Y}(N)_t$ is linear in the truncated signature of X of order N[8].

Example. The simplest example of:

$$dY_t = g(Y_t)dX_t \quad (17)$$

Is a linear physics for a linear path X:

$$dY_t = Y_t dX_t \quad (18)$$

where:

$$g = I = g^{\circ} \quad (19)$$

$$g^{\circ n+1} = D(g^{\circ n})g = Y^{\circ n+1} \quad (20)$$

$$X_t = X_0 + \frac{X_T - X_0}{T} t \quad (21)$$

and assuming:

$$y_0 = 1 \quad (22)$$

$$X_0 = 0 \quad (23)$$

Indeed, this yields the exponential function $Y_t = \exp(X_t) = Y_t^{\circ n}$. For non-linear driving signals (where the order of the events matter), one generally gets a non-commutative version of the exponential function in Eq. (13)! For linear time, the order of events does not matter and we generally get the increment of the path raised to the level of the iterated integral, divided by the level factorial (i.e. the area of an n-dimensional simplex).

This can be seen from:

$$Y_t = y_0 + \sum_{n=1}^{N} Y^{\circ n} \int ... \int_{u_1<...<u_n, u_1,...,u_n \in [0,T]} dX_{u_1} \otimes ... \otimes dX_{u_n} \quad (24)$$

[8]Moreover, the error bounds of $\hat{Y}(N)_t$ to approximate Y_t yield a factorial decay in terms of N, i.e. $|Y_t - \hat{Y}(N)_t| \leq C\frac{|X|^{N+1}_{1,[0,t]}}{N!}$. This result can be extended to p-geometric rough paths where g is a $Lip(K)$ where $K > p-1$ [1].

now it is easy to see that:

$$\begin{aligned}
\Phi^n &= \int ... \int_{u_1<...<u_n, u_1,...,u_n \in [0,t]} dX_{u_1} \otimes ... \otimes dX_{u_n} \\
&= \int ... \int_{0<u_1<...<u_n<t} d(\frac{X_t}{t}u_1)...d(\frac{X_t}{t}u_n) \\
&= \prod_{j=1}^{n}(\frac{X_t}{t}) \int ... \int_{0<u_1<...<u_n<t} du_1...du_n \\
&= \frac{1}{t^N}\prod_{j=1}^{n}(X_t)\frac{t^N}{n!} = \frac{(X_t)^n}{n!}
\end{aligned} \tag{25}$$

such that:

$$Y_t = y_0 + \sum_{n=1}^{N} y_0 \frac{1}{n!}(X_t)^n = 1 + (X_t) + \frac{(X_t)^2}{2!} + \frac{(X_t)^3}{3!} + ... \tag{26}$$

Which is the classical Taylor expansion for the exponential function, i.e. linear physics (identity) integrated over a linear path (time). Now the signature is the generalization of this idea to the path space, where X_t need not be a linear map of time and g need not be linear.

Importantly, the regularity condition to work with Eq. (13) which justifies the linearization of response Y_t as a function of signature terms, is that the function g is sufficiently smooth. This stems from the fact that the approximation error of a truncated $\hat{Y}(N)_t$ decays factorial with N (both for smooth paths and p-geometric rough paths) and the decay constant or rate of decay depends on the smoothness ($Lip(K)$) of the path[9]. Indeed, intuitively one would expect that the more complexity is intrinsic to the path, i.e. the rougher the path, the worse the approximation becomes.

3. Drawdown revisited

So for the following dynamic system:

$$d\xi_t = f(\xi_t)dS_t \tag{27}$$

[9]That is, the decay of Footnote 8 now goes proportionally with 1 over K-factorially fast.

how good is the linear approximation? We know from direct path integration we would have a bad approximation of ξ_t since it cannot be well approximated by linear segments. However, we argue it is smooth enough (Lipschitz-1) to approximate the effect of an increment in the portfolio path on the drawdown using Eq. (13). One of the foremost insights is that, contrary to Taylor expansions where we want to evaluate the derivatives $g^{\circ n}(y_0)$ and use them as coefficients for the power series of X, we just rely on the fact that the signature linearizes the relationship between ξ_t and $\Phi(S)$ and we are not interested in the explicit values of f (or $f^{\otimes n}$) for any value of ξ_t. Given the above, we are just interested in:

$$\hat{\xi}(N)_t = \xi_0 + \sum_{n=1}^{N} L_n \int ... \int_{u_1 < ... < u_n, u_1, ..., u_n \in [0,t]} dX_{u_1} \otimes ... \otimes dX_{u_n} \tag{28}$$

Where L_n is a vector of linear coefficients linking the drawdown at t with the signature terms of order n of the path S. Leveraging factorial decay of the approximation for $Lip(1)$, when taking the full signature $N \to \infty$ we get an arbitrarily close approximation of ξ_t.

A more direct way to interpret L is from the Stone-Weierstrass theorem [3]. Stone-Weierstrass universal approximation theorem is a crucial theorem in proving the universal approximating capabilities of neural networks [4]. Neural networks concatenate linear weighing functions and non-linear sigmoids or ReLU activation functions, that serve as a learnable[10] basis to linearly approximate (the final fully-connected layer) the relationship between any output Y and input X, i.e. where f(X) = Y is some complex polynomial. The same intuition can be applied to signatures. As Eq. (13) implies, it proves to be a natural basis to express the path function Y_t in. This motivated Levin and Lyons [5] to introduce the signature approximation theorem, which states than any continuous[11] h on the path space can be linearly approximated on the signature space:

$$|h(X) - \langle L, \Phi(X) \rangle| \leq \epsilon \tag{29}$$

The proof is extremely concise, and that is why we summarize it here. Since $\Phi(X)$ is a tensor algebra of X, the family of all linear combinations of $\Phi(X)$

[10] I.e. as a function of the network weights.

[11] Sufficiently smooth, i.e. Lipschitz-K function.

is an algebra[12]. Constant functions are preserved by the constant zeroth term of $\Phi(X)$. The algebra separates the points as comes naturally from [6] Corollary 2.16. Stone-Weierstrass then yields that the linear family on $\Phi(X)$ is dense in the space of continuous functions on X.

Now we propose to replace Eq. (29) by:

$$|\Xi(S) - \langle L, \Phi(S)\rangle| \leq \mu \tag{30}$$

for our use case. In theory, this approximation can be done with arbitrary precision μ, but from the above we know that for truncated signatures at level M, there will be an error κ that decays factorially in M.

$$\Xi(S) = \langle L, \Phi^M(S)\rangle + \kappa \tag{31}$$

Note that we could also do an equivalent truncation of the number of linear coefficients $len(L)$ rather than the signature order. However, from Eq. (28) we know that it is more natural to choose a set of linear coefficients that corresponds to a number of signature terms following the choice of M.

Remaining is the choice of L. As the relationship is essentially linear, Eq. (31) can be estimated using simple linear regression (OLS) and higher order polynomials are not required. However, since in practice the sample is limited and the number of signature terms scales exponentially with M and T, one has to be mindful of the overfitting problem, i.e. $len(\Phi) > T$. Therefore, we study the impact of regularized linear regression of $\Xi(X)$ on $\Phi^M(X)$ with a penalty for the number (absolute shrinkage) and size (proportional shrinkage / selection) of estimated coefficients (i.e. the elastic net (ElNet) regression). Hence, we conclude by specifying:

$$\hat{L} = \min_L(||\Xi(S) - \langle L, \Phi^M(S)\rangle||_2 + \lambda_1||L||_1 + \lambda_2||L||_2) \tag{32}$$

$$\hat{\Xi}(S) = \langle \hat{L}, \Phi^M(S)\rangle \tag{33}$$

where $\lambda_1 = \lambda_2 = 0$ corresponds to OLS, $\lambda_1 = 0$ corresponds to Ridge and $\lambda_2 = 0$ to LASSO regression [7].

[12] This can be seen from the shuffle product property, as explained on emiellemahieu.github.io.

	S&P 500 COMP	FTSE NAREIT	S&P GSCI	US TREAS. INDEX
count	5840	5840	5840	5840
mean	1.6369	3.2700	1.4958	2.0467
std	1.0513	1.5413	0.5985	0.6189
min	0.5469	0.9115	0.4515	0.9886
25%	0.8979	1.9393	0.9654	1.4644
50%	1.1244	3.2208	1.4531	2.0515
75%	2.0615	4.4787	1.8236	2.5210
max	5.0199	6.7890	3.9389	3.2731

Table 1: Data description paths

	S&P 500 COMP	FTSE NAREIT	S&P GSCI	US TREAS. INDEX
count	5839	5839	5839	5839
mean	0.0003	0.0003	0.0001	0.0001
std	0.0122	0.0108	0.0146	0.0046
min	-0.1197	-0.1379	-0.1177	-0.0279
25%	-0.0043	-0.0038	-0.0072	-0.0024
50%	0.0003	0.0006	0.0001	0.0001
75%	0.0056	0.0051	0.0079	0.0028
max	0.1158	0.0902	0.0791	0.0413

Table 2: Data description returns

4. Numerical experiments

4.1. Data description

Consider $N = 4$ investible instruments: equity (S&P500), fixed income (US Treasuries), commodities (GSCI index) and real estate (FTSE NAREIT index). We collect price data (close prices) between Jan 2000 and May 2022, which gives us T=5840 daily observations. An overview of the data and descriptive stats are shown in Figure 1 and Table 1 & 2 for the price paths (rescaled to 1 on the first date) and returns respectively. Clearly, these 4 different asset classes have different return, volatility and drawdown characteristics.

As an investor, we are interested in the drawdown of our portfolio $\mathbf{w}$, $w_i, i \in 1, ..., N$, which allocates a weight to each investible asset. In this experiment, we attach random weights to each asset, as shown in Figure 2, and do this $N_p = 100$ times.

This gives rise to sample paths like shown below in Figure 3, where we pick a $\tau < T$ such that we have $T - \tau$ overlapping sample paths, or $N_p(T - \tau)$ portfolio paths. Here, $\tau = 20$, or approximately monthly sample paths:

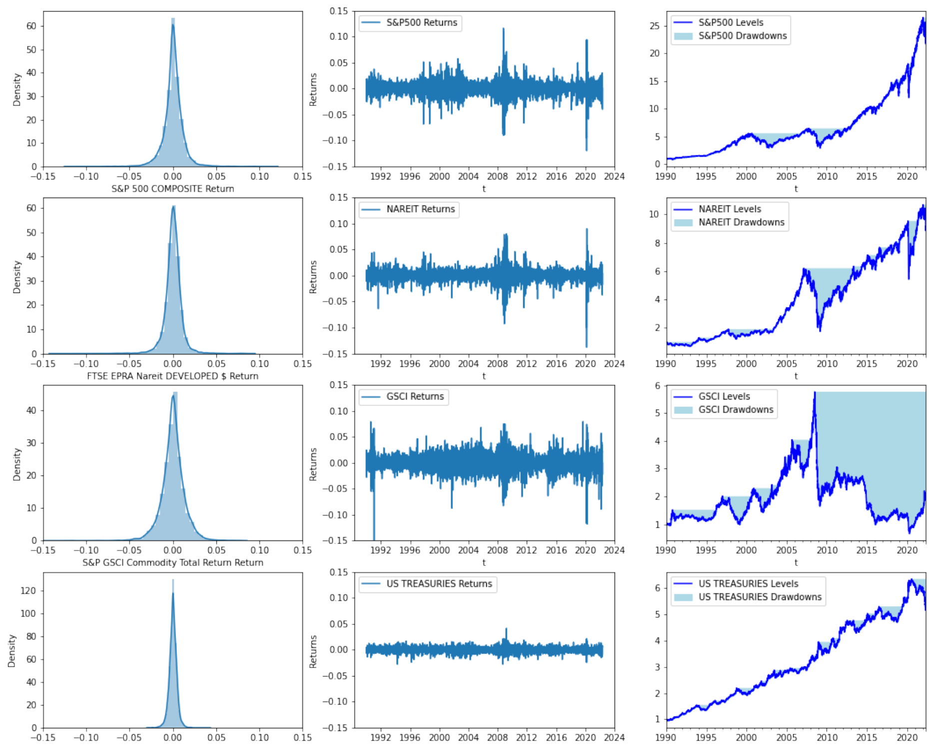

Figure 1: Data overview

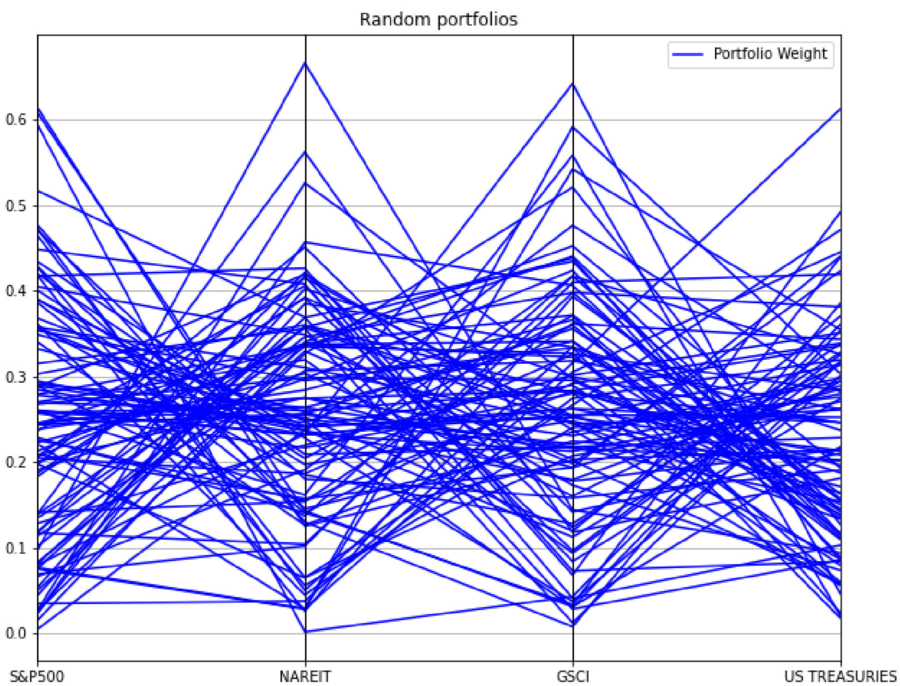

Figure 2: Portfolio simulation

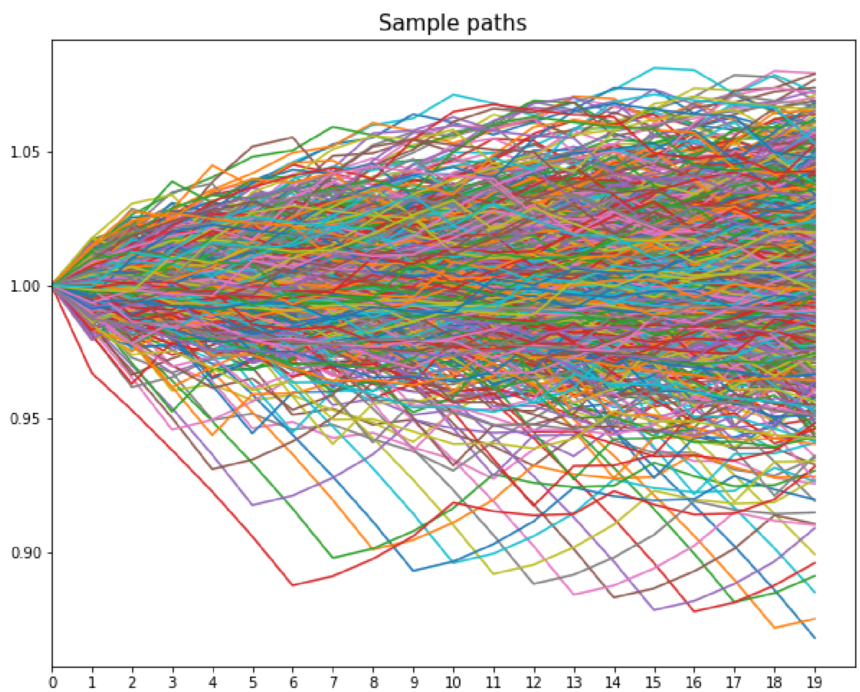

Figure 3: Sample paths

Sig Order	Train R2 %	Time	Len	Test R2 %	Train RMSE	Test RMSE
1	0.01	0.00211	2	0.0110	0.0142	0.0141
2	0.1973	0.00460	6	0.2036	0.0128	0.0126
3	0.4833	0.00268	14	0.4819	0.0102	0.0102
4	0.6670	0.00283	30	0.6728	0.0082	0.0081
5	0.6991	0.00208	62	0.7065	0.0078	0.0077
6	0.7335	0.00209	126	0.7408	0.0074	0.0072
7	0.7488	0.00271	254	0.7559	0.0071	0.0070
8	0.7677	0.00201	510	0.7752	0.0069	0.0067
9	0.7843	0.00284	1022	0.7899	0.0066	0.0065
10	0.8021	0.00301	2046	0.8070	0.0063	0.0062

Table 3: Linear Regression Fit

The drawdown distribution of these paths is shown in Figure 4. This is the probability measure of crucial interest in our future applications. The focal question in drawdown optimization is whether we can adapt input sample paths (or the parametric representation thereof) such that this distribution converges towards the expected drawdown distribution of our portfolio. For now, we are interested in a representation of the drawdown of a portfolio path, on which gradient-based methods can be applied.

The regression results are now easily obtained. For each sample path, we calculate ξ as Eq. (1) explicitly (label). As features, we use the *signatory*[13] package to obtain the signature coefficients of the path, for orders up till M=10. We use simple linear regression, LASSO with 10-fold cross-validation for the shrinkage parameter λ_1 and Elastic Net with 10-fold cross-validation on the parameters λ_1 and λ_2. We take a 80-20 train-test split and further evaluate the out-of-sample accuracy in terms of explained variability (R^2) and root mean squared error ($RMSE$).

4.2. Results and discussion

The results of the linear regression fit are shown in Table 3 and Figures 5,6 and 7. Table 3 shows that the train R^2 initially mounts to relatively high levels (R^2=50-67%) given relatively low (M=3-4) orders of the signature, then converges to about 80% accuracy for M=10 (RMSE=0.0063). Moreover, the train results are consistent with the test accuracies, which makes a case against overfitting and initially does not motivate the use of shrinkage. Furthermore, it is remarkable that for a truncation level of 3-4 we already

[13]https://pypi.org/project/signatory/

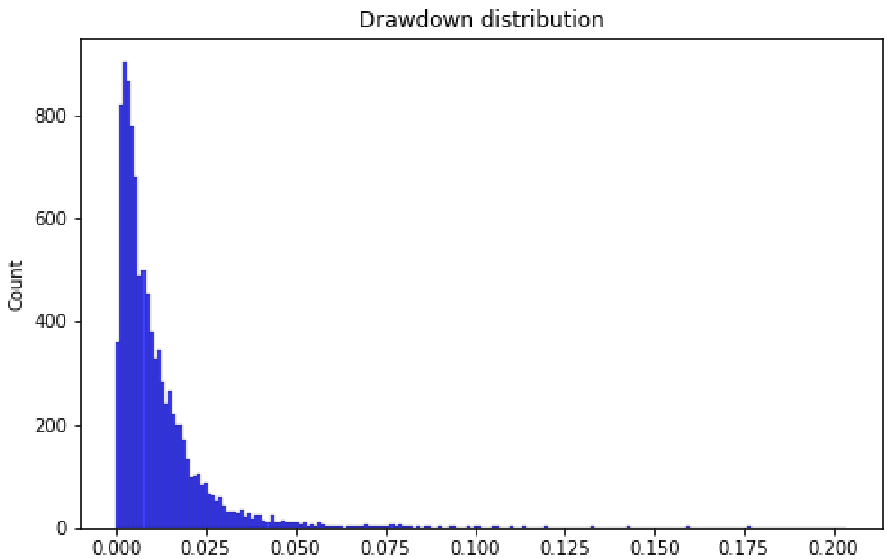

Figure 4: Drawdown distribution

Sig Order	Train R2	Time	Len	Test R2	Train RMSE	Test
1	0.000944	0.00229	2	-0.000416	0.014585	0.016044
2	0.182810	0.00199	6	0.202762	0.013191	0.014323
3	0.471231	0.00178	14	0.500754	0.010611	0.011334
4	0.643187	0.0016	30	0.677618	0.008716	0.00910
5	0.659251	0.00172	62	0.695760	0.008518	0.008848
6	0.657763	0.002	126	0.692452	0.008536	0.008896
7	0.655829	0.00176	254	0.689835	0.008561	0.008933
8	0.654309	0.00209	510	0.687186	0.008579	0.008971
9	0.653079	0.00182	1022	0.685198	0.008595	0.009000
10	0.652002	0.00186	2046	0.683447	0.008608	0.009025

Table 4: Lasso CV(10) Fit

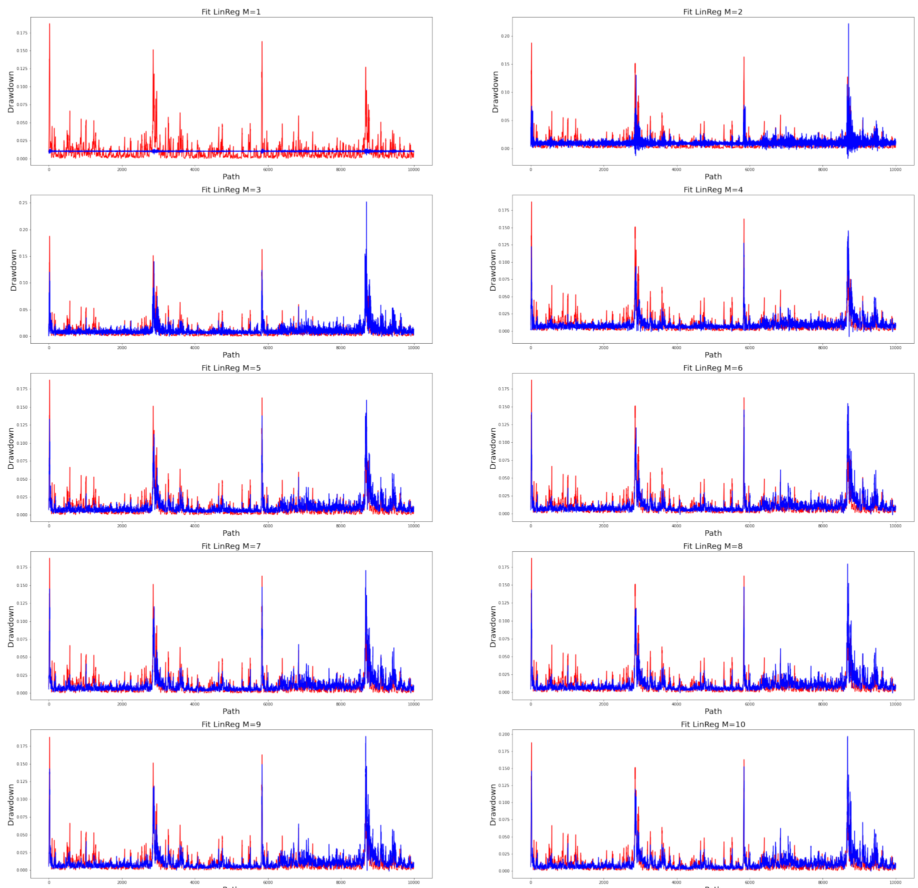

Figure 5: Linear regression fit

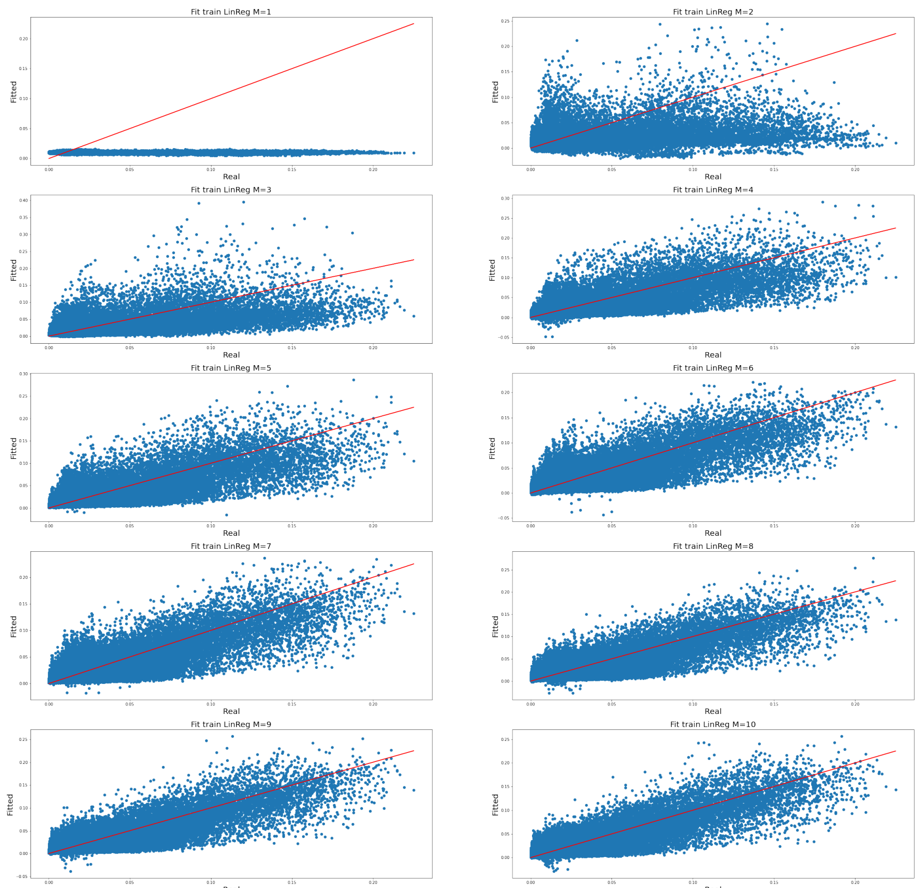

Figure 6: Linear regression train fit scatter

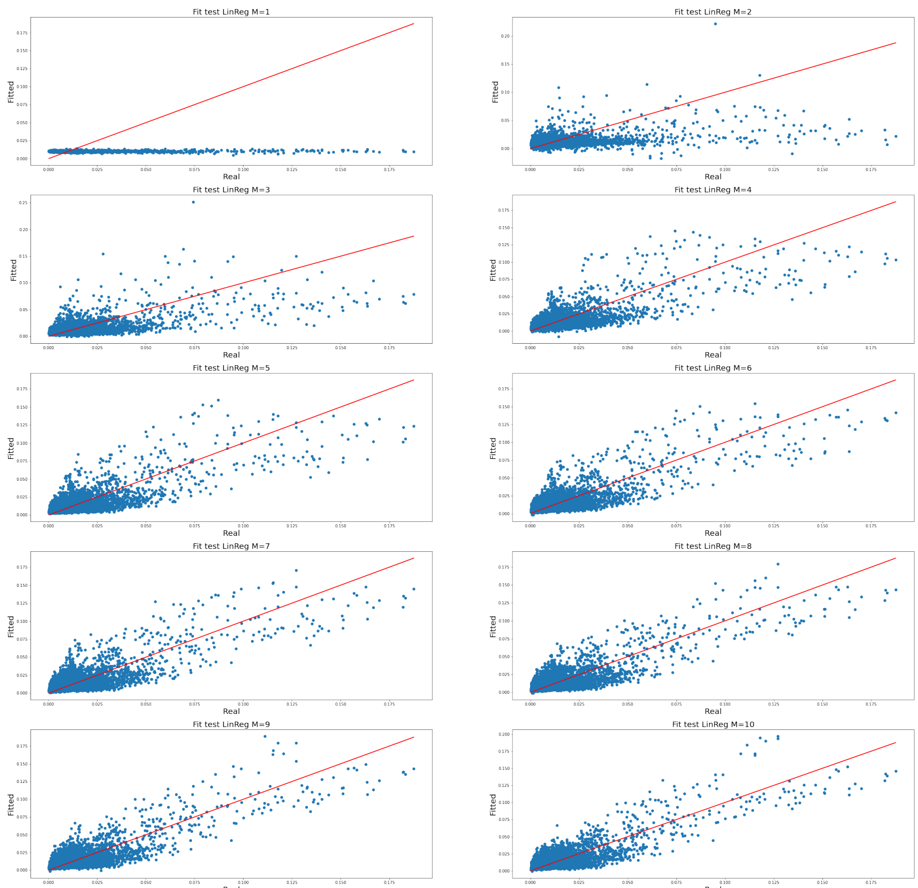

Figure 7: Linear regression test fit scatter

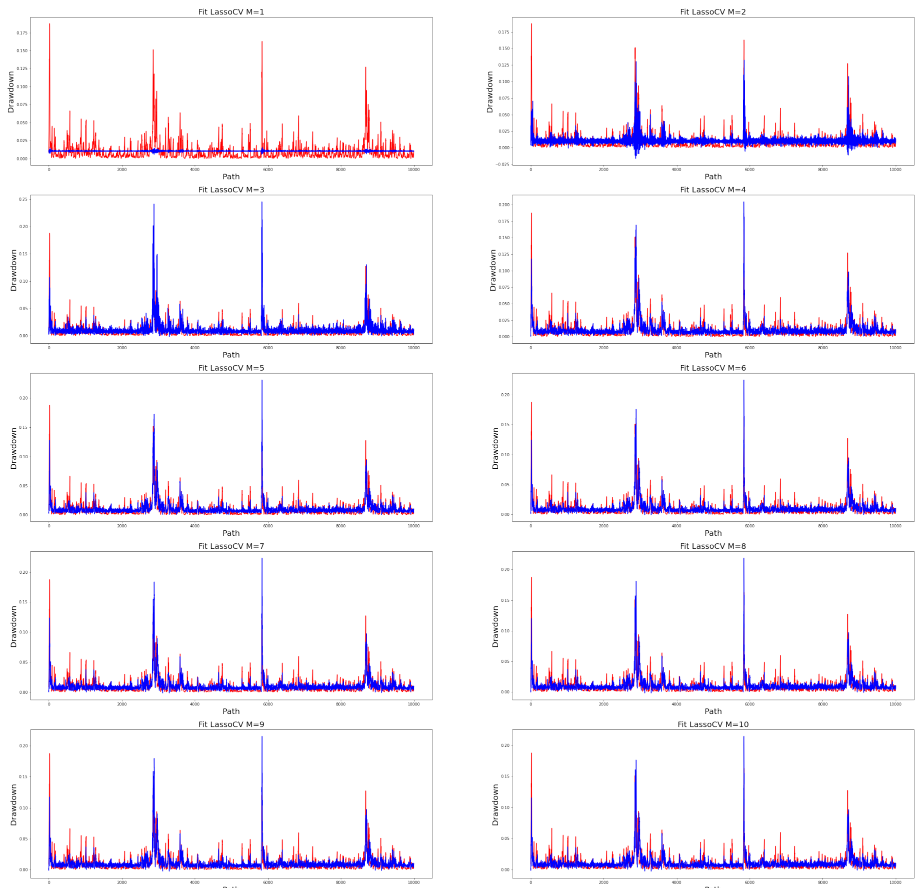

Figure 8: LASSO CV(10) fit

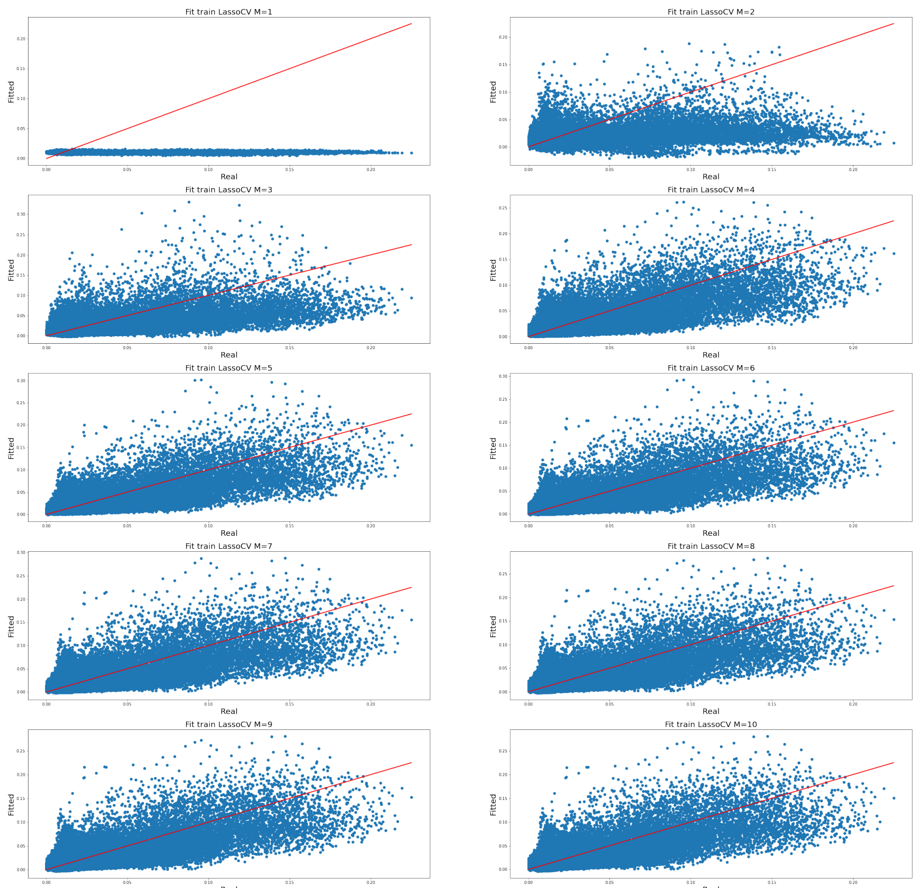

Figure 9: LASSO CV(10) train fit scatter

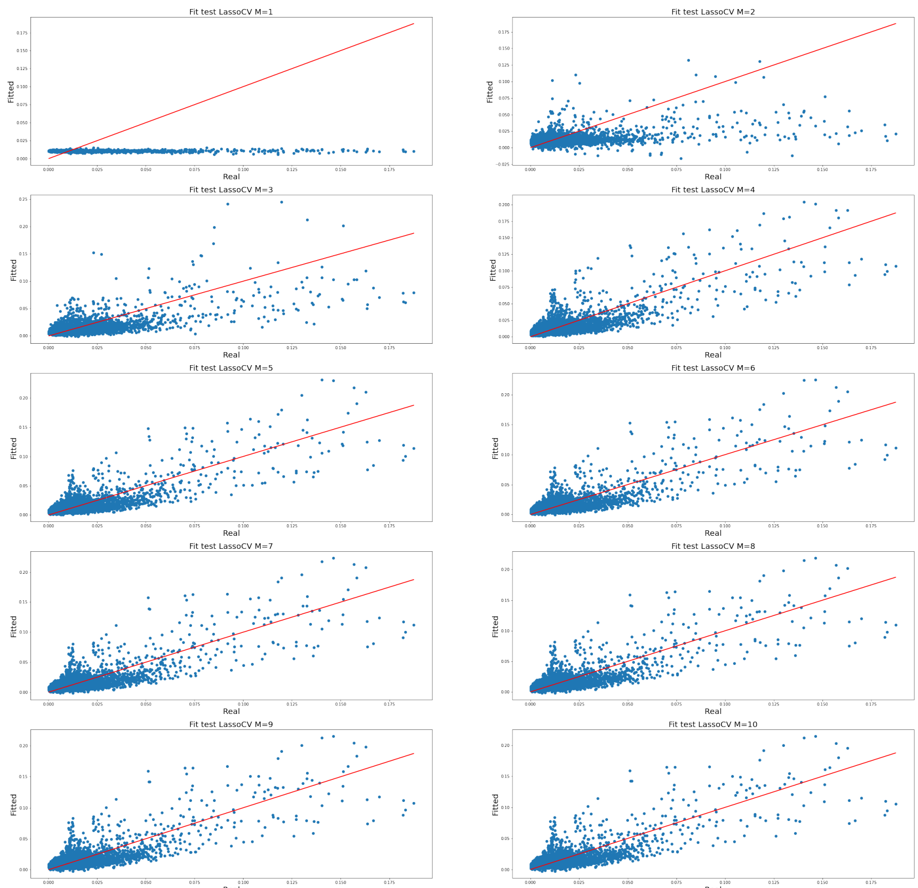

Figure 10: LASSO CV(10) test fit scatter

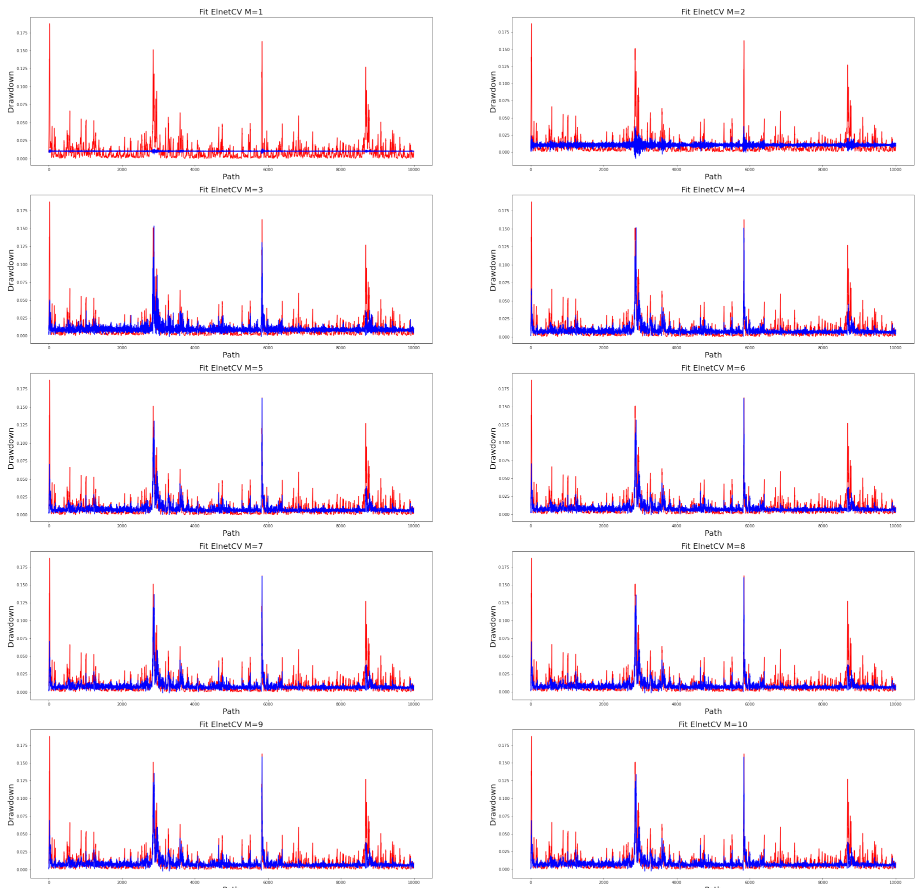

Figure 11: ElNet CV(10) fit

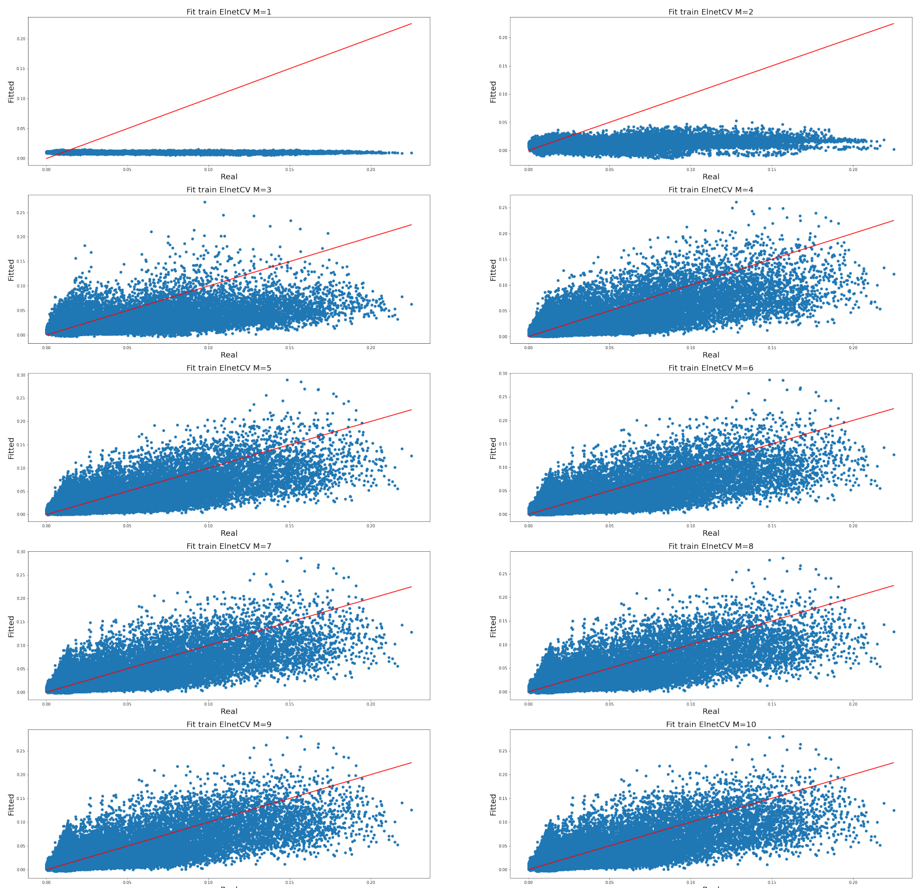

Figure 12: ElNet CV(10) train fit scatter

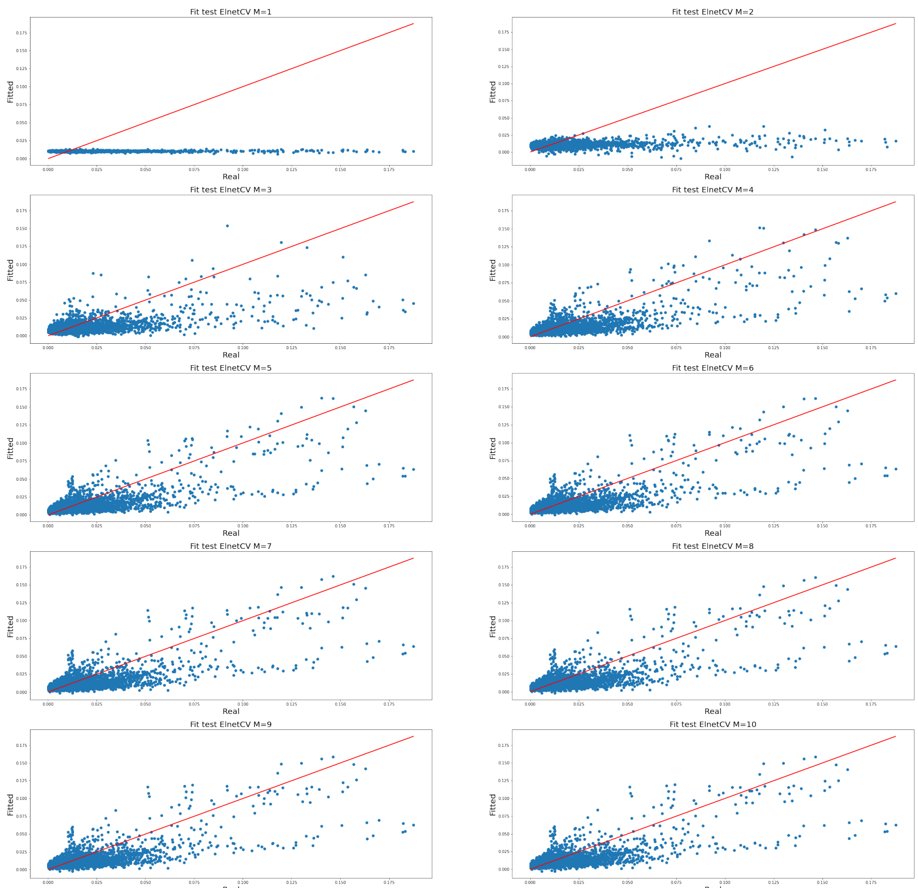

Figure 13: ElNet CV(10) test fit scatter

Sig Order	Train R2	Time	Len	Test R2	Train RMSE	Test RMSE
1	0.000805	0.00212	2	-0.002004	0.014061	0.013651
2	0.061699	0.00227	6	0.060046	0.013626	0.013221
3	0.436781	0.00169	14	0.445956	0.010557	0.010151
4	0.634848	0.00156	30	0.644901	0.008500	0.008126
5	0.661145	0.00157	62	0.672511	0.008188	0.007804
6	0.661234	0.00158	126	0.673225	0.008187	0.007795
7	0.659477	0.00243	254	0.671974	0.008208	0.007810
8	0.658006	0.00184	510	0.670762	0.008226	0.007825
9	0.656879	0.00268	1022	0.669780	0.008240	0.007836
10	0.655936	0.00211	2046	0.668894	0.008251	0.007847

Table 5: Elastic Net CV(10) Fit

have a reasonable approximation of drawdown in both R^2 and RMSE terms. In theory, higher order signatures let the accuracy converge to 100% and 0 respectively, assuming $T \to \infty$. However, from a practical point of view, higher orders do not seem necessary given (1) the low orders' accuracies, (2) they take considerably more time (cf. Table 3 'Time', which is the time to compute 1 signature of the respective order), (3) as the coefficients scale in number they will perfectly overfit the weights $\hat{L}$ on a limited sample, i.e. $T < len(\Phi^M)$.

An intuitive reason why ξ does not require many terms is that the second order signature corresponds to the Levy area, or the area covered between the path and the chord connecting the first and last value of the path. It is a measure of variability related to volatility[14], but geometrically a measure for deviation from monotonicity that is close to drawdown. With drawdown we do not look at the linear chord, but the monotonic growth of the running maximum of the path. This difference explains why the second order certainly does not suffice, but why adding higher order terms fills this extra geometrical area or granularity rather quickly. From the above, we argue that empirical simulation and intuition suggests that signatures are not only an efficient representation of paths in general, but also for drawdowns in particular.

The effect of shrinkage is shown in Table 4 and Figures 8,9 and 10 for LASSO, and in Table 5 and Figures 11,12 and 13 for Elastic Net respectively. We immediately notice that the performance now converges after M=5 al-

[14]For a primer on signatures, including the link between Levy area and volatility, please visit the webpage emiellemahieu.github.io.

ready to about 70% and RMSE of about 80 bps. The rapid increase in number of regressors with the order of the signature explains the domination of the regularization term in the regression objective, biasing the in-sample fit with the aim to have a parsimonious, more robust out-of-sample fit. Indeed, we again find that the train results are highly consistent with the test results. However, given that this was already the case for the unpenalised OLS regression, we argue in favor of OLS given its higher accuracy in terms of both R^2 and RMSE.

5. Application (conceptual)

Imagine a path with a certain level of drawdown. If we would invite you to gradually transform the path such that it better reflects a known or predefined level of drawdown, how would you do this? One way is to craft a formula that generated the path, i.e. the data generating process, and evaluate the sensitivity of its parameters to the resulting drawdown. But how does one (1) handcraft a process that generates a path dependence - where the vast majority of DGPs in finance assumes no path dependences, (2) evaluate the drawdown as a function of this path - if drawdown is not differentiable?

For the former issue, we introduce a market generator model, i.e. a generative machine learning model that uses a flexible mapping to universally approximate the data generating process. This necessitates a differentiable expression of drawdown. For the latter issue, we propose to use the differentiable linear approximation.

In brief, we propose that $\hat{\Xi}(S) = \langle \hat{L}, \Phi^M(S) \rangle$ is a differentiable approximation of the non-differentiable $\Xi(S)$. Intuitively, we just switch between the path space and the signature space to see how a change in a path distribution impacts the drawdown through linear loadings on the transformed basis. This gives us leeway to embed drawdown evaluation in a (neural) system of differentiable equations, i.e. machine learning. Work in progress - Other Paper. In a nutshell: The application that we discuss in the main paper is evaluating synthetic samples in terms of the distribution of Ξ, i.e. Figure 4, versus the known drawdown distribution. The argument is that through data compression we denoise the original data and have closer convergence to true expected drawdown, i.e. denoising by construction in ensemble mean rather than denoising by assumption.

6. Conclusion

This paper takes the perspective of portfolio drawdown as a non-linear dynamic system (a controlled differential equation $d\xi$), rather than a perspective on the solution, the expression of Ξ, directly. It first summarizes the most important insights from the theory of rough paths and path signatures. Further, it pinpointed that using this perspective, rather than differentiability, a smoothness condition is sufficient to accurately describe portfolio drawdown as a function of the path, without having to evaluate Ξ. The paper then built on this to propose a linear signature-based drawdown approximator, which yields a differentiable approximation of portfolio drawdown. Regression results show a good fit for ordinary least squares linear regression, and do not favor regularized versions (LASSO, ElNet). Finally, an application was (conceptually) proposed. More specifically, a market generator model that evaluates the synthetic sample in terms of drawdown. Future work will focus on implementing, testing and extending this application.

References

[1] H. Boedihardjo, T. Lyons, D. Yang, Uniform factorial decay estimates for controlled differential equations, Electronic Communications in Probability 20 (2015) 1–11.

[2] P. K. Friz, M. Hairer, A course on rough paths, Springer, 2020.

[3] L. De Branges, The stone-weierstrass theorem, Proceedings of the American Mathematical Society 10 (1959) 822–824.

[4] N. E. Cotter, The stone-weierstrass theorem and its application to neural networks, IEEE transactions on neural networks 1 (1990) 290–295.

[5] D. Levin, T. Lyons, H. Ni, Learning from the past, predicting the statistics for the future, learning an evolving system, arXiv preprint arXiv:1309.0260 (2013).

[6] T. J. Lyons, M. Caruana, T. Lévy, Differential equations driven by rough paths, Springer, 2007.

[7] T. Hastie, R. Tibshirani, J. H. Friedman, J. H. Friedman, The elements of statistical learning: data mining, inference, and prediction, volume 2, Springer, 2009.