A Study of Data-driven Market Simulator

Master Thesis

Songyan Hou

Supervisor:
Prof. Dr. Josef Teichmann
Department of Mathematics
ETH Zürich

Submitted: March 2, 2021

Abstract

asd

Contents

1. Introduction

2. Financial times-series stimulation

2.1 Challenges in financial scenario

2.2 Stylised facts and evaluation of similarity

2.3 Classical and modern model

3. Signature feature

In this chapter, we give a brief introduction to the signature feature of a path and some variation of signature feature such as log-signature, low-rank-signature and random-signature. Moreover, we discuss the expectation of these signature based features of a stochastic process and emphasize their characteristic ability of the law of stochastic process. For simplicity and readability, we restrict our introduction to the space of continuous bounded variation paths from a compact time interval J to a Euclidean space $E = \mathbb{R}^d$ denoted by $\mathcal{C}_0^1(J, E)$. We postpone the discussion of semi-martingales and rough paths taking values in Banach space E to the appendix and kindly invite interested readers to the comprehensive and rigorous introduction of signature feature in [13] [28].

3.1 Signature

We start the introduction from the tensor algebra in which our signature feature takes value. We consider the non-commuting formal power series of tensors with formal indeterminates as basis of E.

Definition 3.1 (Formal power series)**.** We denote $T((E))$ the space of formal power series of tensors in E that

$$T((E)) = \left\{ \mathbf{a} = (\mathbf{a}_k)_{k\geqslant 0} \colon \mathbf{a}_k \in E^{\otimes k} \right\} = \bigoplus_{m\geqslant 0} E^{\otimes m} \tag{3.1}$$

endowed with addition and multiplication defined as follows: Let $\mathbf{a} = (\mathbf{a}_k)_{k\geqslant 0}, \mathbf{b} =$

$(\mathbf{b}_k)_{k\geqslant 0} \in T((E))$ and $\lambda \in \mathbb{R}$. Then

$$\begin{aligned} \mathbf{a} + \mathbf{b} &= (\mathbf{a}_k + \mathbf{b}_k)_{k\geqslant 0} \\ \mathbf{a} \otimes \mathbf{b} &= \Big(\sum_{i+j=k} \mathbf{a}_i \otimes \mathbf{b}_j \Big)_{k\geqslant 0} \\ \lambda \mathbf{a} &= (\lambda \mathbf{a}_k)_{k\geqslant 0} \end{aligned} \tag{3.2}$$

Let $T^{(n)}(E)$ denote the truncated tensor algebra space up to order n

$$T^{(n)}(E) = \bigoplus_{k\leqslant n} E^{\otimes k} \tag{3.3}$$

Let $e_1, \cdots, e_d$ be a finite basis of $E = \mathbb{R}^d$. Then $\mathbf{a} \in T((E))$ has the following linear form

$$\mathbf{a} = \sum_{k\geqslant 0} \Big(\sum_{i_1,\cdots,i_k=1}^{d} a_{i_1,\cdots,i_k} e_{i_1} \otimes \cdots \otimes e_{i_k} \Big), \quad a_{i_1,\cdots,i_k} \in \mathbb{R}. \tag{3.4}$$

and $T^{(n)}(E)$ can be considered as a subspace of $T((E))$.

Definition 3.2. We denote $\mathbf{T}(E)$ the Banach space

$$\mathbf{T}(E) \coloneqq \Big\{ \mathbf{t} \in T((E)) \colon \|\mathbf{t}\|_{\mathbf{T}(E)} \coloneqq \sqrt{\sum_{k\geqslant 0} \|\mathbf{t}_k\|^2_{E^{\otimes k}}} < \infty \Big\}. \tag{3.5}$$

Similarly, we denote $\mathbf{T}^{(n)}(E)$ the truncation of $\mathbf{T}(E)$ up to order n.

Definition 3.3 (Signature)**.** We denote $\mathbf{Sig}_J \colon \mathcal{C}_0^1(J, E) \to \mathbf{T}(E)$ the signature map such that for all $X \in \mathcal{C}_0^1(J, E)$

$$\mathbf{Sig}_J(X) = (1, \mathbf{s}_1, \cdots) \in \mathbf{T}(E) \tag{3.6}$$

where

$$\begin{aligned} \mathbf{s}_k &= \int_{t_1 < \cdots < t_k \in J} dX_{t_1} \otimes \cdots \otimes dX_{t_k} \\ &= \sum_{i_1,\cdots,i_k=1}^{d} \int_{t_1 < \cdots < t_k \in J} dX^{i_1}_{t_1} \cdots dX^{i_k}_{t_k} \cdot e_{i_1} \otimes \cdots \otimes e_{i_k} \end{aligned} \tag{3.7}$$

Let $\mathbf{Sig}_J^{(n)}$ denote the truncated signature map up to order n

$$\mathbf{Sig}_J^{(n)}(X) = (1, \mathbf{s}_1, \cdots, \mathbf{s}_M) \in \mathbf{T}^{(n)}(E). \tag{3.8}$$

Example 3.4. *Let $X_t = t\mathbf{x} \in \mathbb{R}^d$, then*

$$\mathbf{Sig}_{[0,1]}(X) = (1, \mathbf{x}, \frac{\mathbf{x}^{\otimes 2}}{2!}, \cdots) \in \mathbf{T}(E) \tag{3.9}$$

3.1.1 Signature map

We expect a good feature map to capture important information while ignoring irrelevant ones. First observation is the the signature feature is invariant of starting point because $d(X_t - X_0) = dX_t$. Moreover, it is invariant of reparametrization.

Proposition 3.5 (Invariant under reparametrization)**.** *Let $X \in \mathcal{C}_0^1([S_1, T_1], E)$ and $\tau\colon [S_1, T_1] \to [S_2, T_2]$ a non-decreasing surjective reparametrization. Then*

$$\mathbf{Sig}_{[S_2,T_2]}(X_{\tau(\cdot)}) = \mathbf{Sig}_{[S_1,T_1]}(X) \tag{3.10}$$

Proof.

$$\begin{aligned}\mathbf{Sig}_{[S_2,T_2]}(X_{\tau(\cdot)})_k &= \int_{\tau(t_1)<\cdots<\tau(t_k)\in[S_2,T_2]} dX_{\tau(t_1)} \otimes \cdots \otimes dX_{\tau(t_k)} \\ &= \int_{t_1<\cdots<t_k\in[S_1,T_1]} dX_{t_1} \otimes \cdots \otimes dX_{t_k)} = \mathbf{Sig}_{[S_1,T_1]}(X)\end{aligned} \tag{3.11}$$

□

From the invariant property of reparametrization, we notice that signature map is not injective. However, signature feature is injective up to tree-like equivalence $\sim_t$ which we detailed in appendix, and obviously time-reparametrization is included in tree-like equivalence. We define $\mathcal{P}_0^1 = \mathcal{C}_0^1([0, T], E)/\sim_t$ the quotient space up to tree-like equivalence endowed with quotient metric.

Theorem 3.6 (Weak uniqueness [3])**.** $\mathbf{Sig}_{[0,T]}$ *is injective on $\mathcal{P}_0^1$.*

Proof. See [3]. □

If we add time as an additional strictly increasing coordinate into path ie. $\overline{X_t} = (t, X_t) \in \overline{E} \coloneqq \mathbb{R} \oplus E$, then $\mathbf{Sig}_{[0,T]}$ is injective on the time-augmented space.

Corollary 3.7 (Uniqueness [3])**.** $\mathbf{Sig}_{[0,T]}$ *is injective on $\mathcal{C}_0^1([0, T], \overline{E})$.*

Proof. Since the first coordinate is strictly increasing, the only tree-like equivalent path is itself i.e. $\mathcal{P}_0^1 = \mathcal{C}_0^1([0, T], \overline{E})$, which concludes the proof. □

Proposition 3.8. $\mathbf{Sig}_{[0,1]}$ *is neither surjective nor the range of which a linear subspace of $\mathbf{T}(E)$.*

Proof. Let $X \in \mathcal{C}_0^1([0,T],E)$ and w.l.o.g assume $X_0 = 0$. By integration by part

$$\begin{aligned}\mathbf{Sig}_{[0,1]}(X)_{1,2} + \mathbf{Sig}_{[0,1]}(X)_{2,1} &= \int_{t_1<t_2\in[0,1]} dX^1_{t_1}dX^2_{t_2} + \int_{t_1<t_2\in[0,1]} dX^1_{t_1}dX^2_{t_2} \\ &= \int_{t\in[0,1]} d(X^1_t X^2_t) = \mathbf{Sig}_{[0,1]}(X)_1\ \mathbf{Sig}_{[0,1]}(X)_2.\end{aligned} \tag{3.12}$$

Thus, $\mathbf{Sig}_{[0,1]}$ is neither surjective nor the range of it a linear subspace of $\mathbf{T}(E)$. □

Theorem 3.9 (Weak universality)**.** *Let A be a compact set of $\mathbf{Sig}(\mathcal{C}_0^1(J,E))$, then for all $f\colon A \to \mathbb{R}$ continuous and for all $\epsilon > 0$, there exists a linear functional $L \in \mathbf{T}(E)^*$ such that*

$$\sup_{\mathbf{a}\in A} \|f(\mathbf{a}) - L(\mathbf{a})\| \leqslant \epsilon \tag{3.13}$$

Proof. Proof relies on the Stone-Weierstrass theorem and the shuffle product property of the signature detailed in appendx, see [25]. □

3.1.2 Signature stream

Instead of viewing signature as a static object, we can consider signature stream of a path $X \in \mathcal{C}_0^1([0,T],E)$ as a process $\big(\mathbf{Sig}_{[0,t]}(X)\big)_{[0,T]}$ taking values in $\mathbf{T}(E)$.

Proposition 3.10. *Let $X \in \mathcal{C}_0^1([0,T],E)$ and define $\pi_n\colon \mathbf{T}(E) \to \mathbf{T}^{(n)}(E)$ the projection such that for all $\mathbf{x} \in \mathbf{T}(E)$*

$$\pi_n\big((\mathbf{x}_k)_{k\geqslant 0}\big) = (\mathbf{x}_k)_{k\leqslant n} \tag{3.14}$$

then $S_t = \mathbf{Sig}^{(n)}_{[0,t]}(X)$ satisfies for all $t \in [0,T]$ that

$$dS_t = \pi_n(S_t \otimes dX_t), \quad S_0 = (1,0,\cdots), \tag{3.15}$$

and moreover $(S_t)_{t\in[0,T]}$ is the unique solution of (3.15)*.*

Proof. See Lemma 2.10 in [28]. □

Definition 3.11. Let $X \in \mathcal{C}_0^1([0,s],E)$ and $X \in \mathcal{C}_0^1([s,t],E)$. The concatenated path $X \star Y \in \mathcal{C}_0^1([0,t],E)$ is defined by

$$(X \star Y)_u = \begin{cases} X_u & u \in [0,s] \\ Y_u + (X_s - Y_s) & u \in [s,t] \end{cases} \tag{3.16}$$

Theorem 3.12 (Chen's identity)**.** *Let $X \in \mathcal{C}_0^1([0,s],E)$ and $Y \in \mathcal{C}_0^1([s,t],E)$. Then*

$$\mathbf{Sig}_{[0,t]}(X \star Y) = \mathbf{Sig}_{[0,s]}(X) \otimes \mathbf{Sig}_{[s,t]}(Y) \tag{3.17}$$

Proof. See Theorem 2.9 in [28]. □

Example 3.13. *Let X be linear on $[n, n+1]$ and let $X_{n+1} - X_n = \mathbf{x}_n$ for $n \in \mathbb{N}$, then*

$$\mathbf{Sig}_{[0,N]}(X) = \bigotimes_{n \leqslant N} (1, \mathbf{x}_n, \frac{\mathbf{x}_n^{\otimes 2}}{2!}, \cdots) \tag{3.18}$$

Example 3.14 (Linear controlled differential equation)**.** *Let $E = \mathbb{R}^d, W = \mathbb{R}^n$. let $X \in \mathcal{C}_0^1([0,T],E)$ and let $B \colon E \to \mathbf{L}(W)$ be a bounded linear map. Consider*

$$dY_t = B(dX_t)(Y_t), \quad Y_0 \in W \tag{3.19}$$

If we denote $B^k \coloneqq B(e_k)$, $k = 1, \cdots, d$ then

$$dY_t = \sum_{k=1}^{d} B^k(Y_t) dX_t^k, \quad Y_0 \in W. \tag{3.20}$$

It follows from Picard's iteration that

$$\begin{aligned} Y_t^n &= \left(I + \sum_{k=1}^{n} B^{\otimes k} \int_{t_1 < \cdots < t_k \in [0,t]} dX_{t_1} \otimes \cdots \otimes dX_{t_k} \right) Y_0 \\ &= \left(I + \sum_{k=1}^{n} \sum_{i_1, \cdots, i_k = 1}^{d} B^{i_k} \cdots B^{i_1} \int_{t_1 < \cdots < t_k \in [0,t]} dX_{t_1}^{i_1} \cdots dX_{t_k}^{i_k} \right) Y_0. \end{aligned} \tag{3.21}$$

Let the variation of $X \in \mathcal{C}_0^1([0,T],E)$ denoted by $\|X\|_{[0,T]}$, then

$$\left\| \int_{t_1 < \cdots < t_k \in [0,t]} dX_{t_1} \otimes \cdots \otimes dX_{t_k} \right\|_{E^{\otimes k}} \leqslant \frac{\|X\|_{[0,T]}^k}{k!}. \tag{3.22}$$

Therefore, Y_t^n converges to Y_t as $n \to \infty$ i.e.

$$\|Y_t - Y_t^n\|_W \leqslant \sum_{k>n} \frac{\|B\|_{\mathcal{L}(E,\mathcal{L}(W))}^k \|X\|_{[0,T]}^k}{k!} \leqslant \frac{\|B\|_{\mathcal{L}(E,\mathcal{L}(W))}^{n+1} \|X\|_{[0,T]}^{n+1}}{n!} \to 0, \quad \text{as } n \to \infty \tag{3.23}$$

and

$$Y_t = \left(I + \sum_{k=1}^{\infty} B^{\otimes k} \int_{t_1 < \cdots < t_k \in [0,t]} dX_{t_1} \otimes \cdots \otimes dX_{t_k} \right) Y_0. \tag{3.24}$$

In the language of signature

$$Y_t = \Big(\sum_{k=0}^{\infty} B^{\otimes k} \Big) \Big(\mathbf{Sig}_{[0,t]}(X) \Big) Y_0. \tag{3.25}$$

which implies that the solution of controlled SDE could be written as a linear function on signature stream of control path. This implies that signature stream is a promising feature for controlled ODE.

Remark. Similar result holds for smooth vector field with some additional boundedness assumption and the proof is in alignment with the argument above, see [25].

Despite nice properties as feature map, signature feature suffers from dimension explosion w.r.t truncation order. Consider a path $X \in \mathcal{C}_0^1([0,T], \mathbb{R}^d)$ and the truncated signature of X up to order n in $\mathbf{T}^{(n)}(\mathbb{R}^d) \subseteq \bigoplus_{k \leqslant n} (\mathbb{R}^d)^{\otimes k}$ which has dimension $1 + d + \cdots + d^n = \frac{d^{n+1}-d}{d-1}$ growing exponentially w.r.t the truncation order n. Therefore, we would like to explore some low-rank approximations of signature feature.

3.2 Log-signature

From Proposition 3.8, we know that $\mathbf{Sig}_{[0,1]}$ is neither surjective nor the range of which a linear subspace of $\mathbf{T}(E)$. In Example 3.4, for $X_t = t\mathbf{x} \in \mathbb{R}^d$, then

$$\mathbf{Sig}_{[0,1]}(X) = \sum_{k \geqslant 0} \frac{\mathbf{x}^{\otimes k}}{k!} \tag{3.26}$$

which is the power series expansion of exponential function. Motivated by this, we curve the signature space by taking 'logarithm' which we defined below.

Definition 3.15. Define $\mathbf{T}_1(E) = \{\mathbf{a} \in \mathbf{T}(E) \colon \mathbf{a}_0 = 1\}$. Let $\mathbf{a} \in \mathbf{T}_1(E)$, we define the exponential be

$$\mathbf{exp}(\mathbf{a}) = \sum_{n \geqslant 0} \frac{\mathbf{a}^{\otimes n}}{n!}. \tag{3.27}$$

and we define the logarithm be

$$\mathbf{log}(\mathbf{a}) = \mathbf{log}(1 + \mathbf{t}) = \sum_{n \geqslant 1} \frac{(-1)^{n-1}}{n} \mathbf{t}^{\otimes \mathbf{n}}. \tag{3.28}$$

Lemma 3.16. $\mathbf{exp}(\cdot)$ *and* $\mathbf{log}(\cdot)$ *are inverse of each other on* $\mathbf{T}_1(E)$.

Proof. See Lemma 2.21 in [28]. □

Therefore, taking logarithm of signature loses no information.

Definition 3.17 (Log-signature)**.** We denote $\mathbf{LogSig}_J \colon \mathcal{C}_0^1(J,E) \to \mathbf{T}(E)$ the log-signature map such that for all $X \in \mathcal{C}_0^1(J,E)$

$$\mathbf{LogSig}_J(X) = \mathbf{log}\big(\mathbf{Sig}_J(X)\big) \tag{3.29}$$

Let $\mathbf{LogSig}_J^{(n)}$ denote the truncated signature map up to order n

$$\mathbf{LogSig}_J^{(n)}(X) = \pi_n\big(\mathbf{LogSig}_J(X)\big) \tag{3.30}$$

where π_n define in (3.14).

3.2.1 Lie Series

Besides storing the same information, thanks to the shuffle product property of signature, log-signature space is linear and admits a more concrete representation. Precisely speaking, log-signature space is a linear subspace of Lie formal series over E. Recall the Lie bracket $[\cdot,\cdot]$ on $T((E))$ such that

$$[\mathbf{a},\mathbf{b}] = \mathbf{a}\otimes\mathbf{b} - \mathbf{b}\otimes\mathbf{a} \tag{3.31}$$

If F_1 and F_2 are two linear subspaces of $T((E))$, let us denote by $[F1,F2]$ the linear span of all the elements of the form $[\mathbf{a},\mathbf{b}]$, where $\mathbf{a}\in F_1$ and $\mathbf{b}\in F_2$.

Definition 3.18 (Lie formal series)**.** We denote $\mathcal{L}((E))$ the space of Lie formal series over E such that

$$\mathcal{L}((E)) = \{\mathbf{l}\in T((E)) \colon \mathbf{l}_n \in L_n\} \tag{3.32}$$

where

$$L_0 = 0, L_1 = [E,E], L_2 = [E,L_1], \cdots, L_n = [E,L_{n-1}], \cdots \tag{3.33}$$

and denote $\mathcal{L}^{(n)}(E) = \pi_n(\mathcal{L}((E)))$ the truncated Lie series up to order n.

Theorem 3.19. *The range of log-signature is a linear subspace of Lie series over* E.

$$\mathbf{LogSig}(\mathcal{C}_0^1) \subseteq \mathcal{L}((E)). \tag{3.34}$$

Proof. See Theorem 2.23 in [28]. □

Theorem 3.20. *The range of truncated log-signature is the truncated Lie series over E up to the same order*

$$\mathbf{LogSig}^{(n)}(\mathcal{C}_0^1) = \mathcal{L}^{(n)}(E) \tag{3.35}$$

Proof. See Proposition 2.27 in [28]. □

With Lie series form, we compute the dimension of truncated log-signature space.

Proposition 3.21. *The dimension of the space of truncated log-signature up to order n is*

$$w(d,n) = \sum_{k=1}^{n} \frac{1}{k} \sum_{i|k} \mu\Big(\frac{k}{i}\Big) d^i \tag{3.36}$$

which is the Witt's formula and μ is the Möbius function.

Proof. See Corollary 4.14 in [29]. □

If we let $d = 5$ and $n = 5$, then $\dim(\mathbf{Sig}) = 3905$ while $\dim(\mathbf{LogSig}) = 829$. In summary, log-signature curves the signature space and reduces the redundancy without losing any information. However, there is no free lunch because log-signature loses the universality of signature, therefore requiring nonlinear models.

Example 3.22. *For $X_t = t\mathbf{x} \in \mathbb{R}^d$, then*

$$\mathbf{Sig}_{[0,1]}(X) = \mathbf{exp}(\mathbf{x}), \quad \mathbf{LogSig}_{[0,1]}(X) = \mathbf{x}. \tag{3.37}$$

Log-signature pick non-redundant information in signature.

3.3 Universality and characteristics

Recall the weak universality of signature, we restrict ourselves on a compact set of $\mathbf{Sig}(\mathcal{C}_0^1(J,E))$. To generalize the theorem beyond compactness, we would like to normalize the set $\mathbf{Sig}(\mathcal{C}_0^1(J,E))$ into a bounded ball. The first idea comes to mind is to normalize signatures by scaling them on $\mathbf{T}(E)$. However, because signature map is neither surjective nor the range of it a linear subspace, scaled tensor of a signature might not be signature. Thus, we drive ourselves out of the region of $\mathbf{Sig}(\mathcal{C}_0^1(J,E))$ where we have the shuffle product property. An alternative is to normalize the

signature by scaling the path. For $X \in \mathcal{C}_0^1(J, E)$ and $\lambda > 0$, the signature of the scaled path

$$\begin{aligned}\mathbf{Sig}_J(\lambda X)_k &= \int_{t_1 \leqslant \cdots \leqslant t_k \in J} d\lambda X_{t_1} \otimes \cdots \otimes d\lambda X_{t_k} \\ &= \lambda^k \int_{t_1 \leqslant \cdots \leqslant t_k \in J} dX_{t_1} \otimes \cdots \otimes dX_{t_m} = \lambda^k \mathbf{Sig}_J(X)_k\end{aligned} \tag{3.38}$$

Thus, the norm of the scaled path

$$\left\|\mathbf{Sig}_J(\lambda X)\right\|_{\mathbf{T}(E)}^2 = \sum_{k \geqslant 0} \lambda^{2k} \|\mathbf{Sig}_J(X)_k\|_{E^{\otimes k}}^2 \tag{3.39}$$

Therefore, we define the scaling map δ_λ on $\mathbf{T}(V)$ such that for all $\mathbf{a} \in \mathbf{T}(V)$

$$\delta_\lambda(\mathbf{a}) = (\lambda^k \mathbf{a}^k)_{k \geqslant 0} \tag{3.40}$$

Definition 3.23. A tensor normalization is a continuous injective map

$$\Lambda \colon \mathbf{T}(V) \to \{\mathbf{t} \in \mathbf{T}(V) \colon \|\mathbf{t}\|_{\mathbf{T}(E)} \leqslant K\} \tag{3.41}$$

$$\mathbf{t} \mapsto \delta_{\lambda(\mathbf{t})}(\mathbf{t}) \tag{3.42}$$

where $K > 0$ and $\lambda \colon \mathbf{T}(V) \to (0, \infty)$ a function.

The existence of tensor normalization is not trivial because of the nonlinear relationship between the scaling factor $\lambda(\mathbf{t})$ and the norm of scaled tensor $\|\delta_{\lambda(\mathbf{t})}(\mathbf{t})\|$ shown above. The proof of existence and the construction methodology can be found in the appendix.

Theorem 3.24 ([6])**.** *Let $\Lambda \colon \mathbf{T}(E) \to \mathbf{T}(E)$ be a tensor normalization. The normalized signature*

$$\Phi = \Lambda \circ \mathbf{Sig}_J \tag{3.43}$$

(i) is a continuous injection from $\mathcal{P}_0^1$ in to a bounded subset of $\mathbf{T}(E)$,

(ii) is universal to $C_b(\mathcal{P}_0^1, \mathbb{R})$, equipped with the strict topology,

(iii) is characteristic to the space of finite regular Borel measures on $\mathcal{P}_0^1$.

Proof. See Proposition 4.1 in [6]. □

Corollary 3.25. *Let X a stochastic process on $[0,1]$ and measurable $(\Omega, \mathcal{F})$. Let $\mathbb{P}$ and $\mathbb{Q}$ be two regular probability measures such that $X \in \mathcal{P}_0^1$ almost surely. Then*

$$\mathbb{E}_{\mathbb{P}}[\Phi(X)] = \mathbb{E}_{\mathbb{Q}}[\Phi(X)] \quad \textit{iff} \quad \mathbb{P} = \mathbb{Q} \tag{3.44}$$

Proof. Proof directly from (iii) of Theorem 3.24. □

Remark. If we consider the regular probability measure on time augmented path space $\mathcal{C}_0^1([0,T], \overline{E})$, similar result holds since $\mathcal{C}_0^1([0,T], \overline{E}) = \mathcal{P}_0^1$.

Definition 3.26 (Maximum mean distance)**.** We define the maximum mean distance (MMD) as

$$d_{\mathcal{G}}(\mathbb{P}, \mathbb{Q}) = \sup_{f \in \mathcal{G}} |\mathbb{E}_{\mathbb{P}}[f] - \mathbb{E}_{\mathbb{Q}}[f]| \tag{3.45}$$

where $\mathcal{G} \subseteq \mathbb{R}^{\mathcal{P}_0^1}$.

Let $\mathcal{G} = \{\langle \mathbf{l}, \Phi(\cdot) \rangle_{\mathbf{T}_1} : \mathbf{l} \in \mathbf{T}_1, \|\mathbf{l}\|_{\mathbf{T}_1} \leqslant 1\}$, then by Corollary 3.25

$$d_{\mathcal{G}}(\mathbb{P}, \mathbb{Q}) = 0 \quad \text{iff} \quad \mathbb{P} = \mathbb{Q} \tag{3.46}$$

which implies that a metric, see [12]. Moreover

$$\begin{aligned} d_{\mathcal{G}}(\mathbb{P}, \mathbb{Q}) &= \sup_{f \in \mathcal{G}} |\mathbb{E}_{\mathbb{P}}[f] - \mathbb{E}_{\mathbb{Q}}[f]| \\ &= \sup_{f \in \mathcal{G}} \Big| \mathbb{E}_{(X,Y) \sim \mathbb{P} \otimes \mathbb{Q}}[f(X) - f(Y)] \Big| \\ &= \sup_{\mathbf{l} \in \mathbf{T}_1, \|\mathbf{l}\|_{\mathbf{T}_1} \leqslant 1} \Big| \mathbb{E}_{(X,Y) \sim \mathbb{P} \otimes \mathbb{Q}} \big[\langle \mathbf{l}, \Phi(X) - \Phi(Y) \rangle_{\mathbf{T}_1} \big] \Big| \end{aligned} \tag{3.47}$$

Since $\mathbb{E}_{(X,Y) \sim \mathbb{P} \otimes \mathbb{Q}} \big[\Phi(X) - \Phi(Y) \big] \in \mathbf{T}_1$, then

$$d_{\mathcal{G}}(\mathbb{P}, \mathbb{Q}) = \mathbb{E}\big[\langle \Phi(X) - \Phi(Y), \Phi(X') - \Phi(Y') \rangle_{\mathbf{T}_1} \big] \tag{3.48}$$

where X, Y, X', Y' are independent with $X, X' \sim \mathbb{P}$ and $Y, Y' \sim \mathbb{Q}$

Definition 3.27. We denote $\mathbf{k_{Sig}} \colon \mathbf{T}_1 \times \mathbf{T}_1 \to \mathbb{R}$ the signature kernel such that

$$\mathbf{k_{Sig}}(\cdot, \cdot) = \langle \Phi(\cdot), \Phi(\cdot) \rangle_{\mathbf{T}_1} \tag{3.49}$$

Then we can rewrite the MMD as

$$d_{\mathcal{G}}(\mathbb{P}, \mathbb{Q}) = \mathbb{E}\big[\mathbf{k_{Sig}}(X, X') \big] - 2\mathbb{E}\big[\mathbf{k_{Sig}}(X, Y) \big] + \mathbb{E}\big[\mathbf{k_{Sig}}(Y, Y') \big] \tag{3.50}$$

where X, Y, X', Y' are independent with $X, X' \sim \mathbb{P}$ and $Y, Y' \sim \mathbb{Q}$. Similar to the kernel trick in machine learning, the kernel representation of MMD provides a very efficient way evaluating the metric, which we will elaborate in the next section.

3.4 Discrete signature feature

Let $X \in \mathcal{C}_0^1([0,1],E)$ and consider a discrete sequence $\mathbf{X} = (X_0, X_{1/N}, \cdots, X_1) \in E^{N+1}$. We extend the concept of signature on discrete sequence E^{N+1} by computing signature of the linear interpolation of sequence. From Example 3.13 we know the signature of piecewise-linear path is

$$\mathbf{exp}(\mathbf{X}_1 - \mathbf{X}_0) \otimes \cdots \otimes \mathbf{exp}(\mathbf{X}_N - \mathbf{X}_{N-1}) \tag{3.51}$$

Therefore we define the discrete signature as follows

Definition 3.28. Let $\mathbf{X} = (\mathbf{X}_i)_{i=0}^N \in E^{N+1}$ and let $\triangle\mathbf{X}_i = \mathbf{X}_i - \mathbf{X}_{i-1}$. We denote $\mathbf{sig}_{[0,N]}$ the discrete signature of order m and depth n such that

$$\mathbf{sig}_{[0,N]}(\mathbf{X}) = \pi_n\bigg(\bigotimes_{i=1}^N \pi_m\Big(\mathbf{exp}\big(\triangle\mathbf{X}_i\big)\Big)\bigg) \tag{3.52}$$

In particular, if $m = 1$ we call it growing discrete signature

$$\begin{aligned}\mathbf{sig}_{[0,N]}(\mathbf{X}) &= \pi_n\bigg(\prod_{i=1}^N \big(1 + \triangle\mathbf{X}_i\big)\bigg) \\ &= \sum_{k=0}^n \sum_{i_1<\cdots<i_k=1}^N \triangle\mathbf{X}_{i_1} \otimes \cdots \otimes \triangle\mathbf{X}_{i_k}.\end{aligned} \tag{3.53}$$

If $m > 1$, we call it high order discrete signature, and if $m \geqslant n$, we call it truncated discrete signature

$$\begin{aligned}\mathbf{sig}_{[0,N]}(\mathbf{X}) &= \pi_n\bigg(\prod_{i=1}^N \mathbf{exp}\big(\triangle\mathbf{X}_i\big)\bigg) \\ &= \sum_{k=0}^n \sum_{i_1\leqslant\cdots\leqslant i_k=1}^N \triangle\mathbf{X}_{i_1} \otimes \cdots \otimes \triangle\mathbf{X}_{i_k}\end{aligned} \tag{3.54}$$

Lemma 3.29. *Let $X \in \mathcal{C}_0^1([0,1],E)$ and $\mathbf{X} = (X_{i/N})_{i=0}^N \in E^{N+1}$, then*

$$\|\mathbf{Sig}_{[0,1]}(X) - \mathbf{sig}_{[0,N]}(\mathbf{X})\|_{\mathbf{T}_1} \to 0 \quad as \quad N \to 0 \tag{3.55}$$

Proof. See Theorem 5 in [23]. □

Remark. If we let X to be the Brownian motion (not in $\mathcal{C}_0^1$), the growing discrete signature converges to the signature defined with Ito integral, while the high order discrete signature converges to the signature defined with Stratonovich integral. More generally, if we want to approximate a geometric p-rough path, we need high order discrete siganture at least order $\lfloor p \rfloor$, see [23].

3.4.1 Kernel trick

In the same fashion of horner algorithm

$$\begin{aligned} &a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n \\ &\quad = a_0 + x\Big(a_1 + x\Big(a_2 + x\big(a_3 + \cdots + x(a_{n-1} + x\, a_n)\cdots\big)\Big)\Big). \end{aligned} \tag{3.56}$$

we can reduce the computational complexity of discrete signature by dynamic programming principle.

Proposition 3.30. *Let* $\mathbf{X} = (\mathbf{X}_i)_{i=0}^N \in E^{N+1}$ *and let* $\mathbf{sig}_{[0,N]}$ *be the growing signature*

$$\begin{aligned} \mathbf{sig}_{[0,N]}(\mathbf{X}) &= \mathbf{sig}_{[0,N-1]}(\mathbf{X}) \otimes (1 + \triangle \mathbf{X}_N) \\ &= 1 + \sum_{i_1=1}^{N} \triangle \mathbf{X}_{i_1} \Big(1 + \sum_{i_2=i_1+1}^{N} \triangle \mathbf{X}_{i_2}(\cdots)\Big) \end{aligned} \tag{3.57}$$

This is too surprise because we just rewrite the Chen's identity. However, this greatly reduce the complexity when computing the kernel.

Definition 3.31. Let $\mathbf{sig}_{[0,N]}$ be the discrete signature, then we denote $\mathbf{k}_{\mathbf{sig}}$ the discrete signature kernel such that

$$\mathbf{k}_{\mathbf{sig}}(\cdot,\cdot) = \langle \mathbf{sig}_{[0,N]}(\cdot), \mathbf{sig}_{[0,N]}(\cdot) \rangle_{\mathbf{T}_1} \tag{3.58}$$

Proposition 3.32. *Let* $\mathbf{X} = (\mathbf{X}_i)_{i=0}^N \in E^{N+1}$ *and let* $\mathbf{Y} = (\mathbf{Y}_i)_{i=0}^N \in E^{N+1}$. *Let* $\mathbf{k}_{\mathbf{sig}}$ *be the kernel of discrete growing signature*

$$\mathbf{k}_{\mathbf{sig}}(\mathbf{X},\mathbf{Y}) = 1 + \sum_{\substack{i_1=1 \\ j_1=1}}^{N} \langle \triangle \mathbf{X}_{i_1}, \triangle \mathbf{Y}_{j_1} \rangle \Big(1 + \sum_{\substack{i_2=i_1+1 \\ j_2=j_1+1}}^{N} \langle \triangle \mathbf{X}_{i_2}, \triangle \mathbf{Y}_{j_2} \rangle (\cdots)\Big) \tag{3.59}$$

Remark. Similar formulas hold for high order discrete signature and kernel, see [23].

This result is important for numerical implementation in two aspects. First is that we avoid computing the tensor but the inner product directly by kernel trick. If somehow (for better characteristic capacity) you would like lift paths to a RKHS space (H, κ) before applying the signature kernel, this formula avoids explicitly compute $\triangle \kappa_X$ and $\triangle \kappa_Y$. but approximating with

$$\langle \triangle \kappa_{X_i}, \triangle \kappa_{Y_j} \rangle \approx \kappa(X_{i+1}, Y_{j+1}) + \kappa(X_i, Y_j) - \kappa(X_i, Y_{j+1}) - \kappa(X_{i+1}, Y_j), \tag{3.60}$$

which reduce a potentially infinite computational complexity if (H, κ) is an infinite dimensional space, see [6]. Secondly, the recursive structure in time implies that computing the inner product of signature stream is as cheap as computing the inner product of signature. This recursive feature can also be used for more general inner product between tensor and discrete signature.

3.5 Low-rank-signature

We extend the recursive method from discrete signature to tensor with similar recursive structure. Let $\mathbf{l} = (\mathbf{l}_k)_{k=0}^{\infty} \in \mathbf{T}_1(E)$, let $\mathbf{X} = (\mathbf{X}_i)_{i=0}^{N} \in E^{N+1}$,and let $\mathbf{sig}_{[0,N]}$ be the growing signature. If $\mathbf{l}_k = \mathbf{l}_{k-1} \otimes l_k$ then

$$\begin{aligned}\langle \mathbf{l}, \mathbf{sig}_{[0,N]}(\mathbf{X})\rangle_{\mathbf{T}_1(E)} &= \sum_{k\geqslant 0}\langle \mathbf{l}_k, \mathbf{sig}^{(k)}_{[0,N]}(\mathbf{X})\rangle_{\mathbf{T}_1(E)} \\ &= \sum_{k\geqslant 0}\sum_{i=1}^{N}\langle \mathbf{l}_{k-1}, \mathbf{sig}^{(k-1)}_{[0,i-1]}(\mathbf{X})\rangle_{\mathbf{T}_1(E)} \cdot \langle l_k, \triangle \mathbf{X}_i\rangle_{\mathbf{T}_1(E)} \\ &= 1 + \sum_{\substack{i_1=1\\ j_1=1}}^{N}\langle l_{i_1}, \triangle \mathbf{X}_{j_1}\rangle\Big(1 + \sum_{\substack{i_2=i_1+1\\ j_2=j_1+1}}^{N}\langle l_{i_2}, \triangle \mathbf{X}_{j_2}\rangle(\cdots)\Big)\end{aligned} \tag{3.61}$$

This is exactly the formula (3.59) in Proposition 3.32. In the intermediate step of computing the inner product, we have obtained the value of $\langle \mathbf{l}_k, \mathbf{sig}^{(k)}_{[0,N]}(\mathbf{X})\rangle_{\mathbf{T}_1(E)}$ for all k. Therefore, if we consider a sequence of L many tensors such that $\mathbf{l}_k = \mathbf{l}_{k-1} \otimes l_k$, then we can leverage the recursive structure in order and compute their inner product with signature in linear computational complexity w.r.t L.

Remark. This result can also be generalized to higher order case by considering the recursive algorithm computing high order discrete signature and kernel, see [23]. However, in the paper [33] introducing low-rank tensor projection of ordered sequential data, only first order recursive structure is considered. We believe that higher order recursive structure also exists and we wish to prove this claim in further research.

If we only assume $\mathbf{l}_k = l_{k,1} \otimes \cdots \otimes l_{k,k}$, then we lose the recursive in order, but still we have the recursive in time of discrete signature and the kernel trick.

$$\langle \mathbf{l}_k, \mathbf{sig}^{(k)}_{[0,N]}(\mathbf{X})\rangle_{\mathbf{T}_1(E)} = \sum_{i=1}^{N}\langle l_{k,1} \otimes \cdots \otimes l_{k-1,k}, \mathbf{sig}^{(k-1)}_{[0,i-1]}(\mathbf{X})\rangle_{\mathbf{T}_1(E)} \cdot \langle l_{k,k}, \triangle \mathbf{X}_i\rangle_{\mathbf{T}_1(E)} \tag{3.62}$$

Definition 3.33. We call $\mathbf{l} \in \mathbf{T}_1$ rank-1 tensor if

$$\mathbf{l} = l_1 \otimes \cdots \otimes l_k, \quad \text{for some } k \geqslant 1 \tag{3.63}$$

Similar to log-signature, the inner product between truncated signature and rank-1 tensor map the signature space to a low dimensional space $\mathbb{R}$. This projection is linear, computational cheap, and more importantly trainable. The choice of rank-1 tensors gives us the freedom to choose the output space dimension as well as the direction of projection.

Definition 3.34. Let $(\mathbf{l}_l)_{l=0}^L$ a sequence of rank-1 tensors in $\mathbf{T_1}$ and let $\mathbf{sig}_{[0,N]}$ the discrete signature, then we denote $\mathbf{LRsig}_{[0,N]}$ the low-rank signature such that

$$\mathbf{LRsig}_{[0,N]} = \left(\langle \mathbf{l}_l, \mathbf{sig}_{[0,N]} \rangle_{\mathbf{T}_1(E)}\right)_{l=0}^L \in \mathbb{R}^L \tag{3.64}$$

and we define the low-rank signature stream $(\mathbf{LRsig}_{[0,i]})_{i=0}^N$.

From the recursive algorithm above, we know that computing the low-rank signature stream is as cheap as computing the low-rank signature. Unlike signature stream, low-rank signature stream still in a low dimensional space, so we may again apply the low-rank signature stream to it. Thus, we may stack many levels of signature stream on the original path and this has been proved successful dealing with sequential data, see [33].

4. Variational Autoencoders

In this chapter, we give a brief introduction to the variational autoencoders (VAEs) [21] and some variations of VAE such as conditional VAEs (CVAEs) [31], $\beta-$VAEs [16], student-VAEs [2], and gaussian process VAEs (GP-VAEs) [10]. Variational autoencoders is a generative model which simulates how the data is generated under certain presumed distribution. In short, VAEs are variational bayesian inference on latent variable models with likelihood and posterior described by neural networks. In the scenario of generating time series, there are in general two frameworks to apply variational autoencoders. First is to assume a bayesian model with both observable variables and latent variables taking values in space of time series [10] [33], for example taking latent space as a gaussian process in GP-VAEs. Another approach is to first consider an appropriate feature map from time series to a finite dimensional features space and apply VAE on this feature space. After the training of VAE on features space, we generate new samples on the feature space and transform those samples on feature space back to time series with the inverse map of the feature map [5]. More details and background information on VAEs can be found in [22].

4.1 Variational autoencoders

Variational autoencoders are variational inference on deep latent variable model. Deep latent variable models consist of observable random variable x, latent random variable z, and parameter θ. The joint distribution $p_\theta(x, z)$ of the deep latent variable model is parameterized by neural networks. For example, we may choose z be standard normal distribution and x a normal distribution with mean $\mu = \mathbf{NN}(z)$ and variance 1, where $\mathbf{NN}$ is a neural network. Moreover, we call $p_\theta(z) = p(z)$ the *prior* (independent with θ), $p_\theta(z|x)$ the *posterior*, $p_\theta(x|z)$ the *conditional distribution* and $p_\theta(x)$ the *marginal likelihood*. The big advantage of deep latent variable model

is that even when the prior $p(z)$ and conditional distribution $p(x|z)$ are explicit and simple, the marginal likelihood $p_\theta(x)$ can be very expressive due to the universal approximating capacity of neural network. However, the price to pay for expressive marginal likelihood is the intractability of marginal likelihood, namely having no analytic solution or efficient estimator for it. By Bayes' rule

$$p_\theta(x) = \frac{p_\theta(x,z)}{p_\theta(z|x)}.$$

The joint distribution is not hard to write down since

$$p_\theta(x,z) = p\big(x|\mathbf{NN}_\theta(z)\big)p(z)$$

but the posterior $p_\theta(z|x)$ in general has no analytic solution or efficient estimator. Therefore, we need to leverage the idea of variational inference, introducing a distribution $q_\phi(z|x)$ approximating the true posterior $p_\theta(z|x)$.

4.1.1 Variational inference

Let $x_1, \cdots, x_n$ be i.i.d. observable samples drawn from a random variable x. Maximizing the log-marginal likelihood by Bayes's rule, we obtain

$$\begin{aligned}
\log p_\theta(x_i) &= \log\Big(\frac{p_\theta(x_i,z)}{p_\theta(z|x_i)}\Big) \\
&= \log\Big(\frac{p_\theta(x_i,z)}{q_\phi(z|x_i)}\frac{q_\phi(z|x_i)}{p_\theta(z|x_i)}\Big) \\
&= \mathbb{E}_{q_\phi(z|x_i)}\bigg[\log\Big(\frac{p_\theta(x_i,z)}{q_\phi(z|x_i)}\Big) + \log\Big(\frac{q_\phi(z|x_i)}{p_\theta(z|x_i)}\Big)\bigg] \\
&= \mathbb{E}_{q_\phi(z|x_i)}\bigg[\log\Big(\frac{p_\theta(x_i,z)}{q_\phi(z|x_i)}\Big) + \log\Big(\frac{q_\phi(z|x_i)}{p_\theta(z|x_i)}\Big)\bigg] \\
&= \mathbb{E}_{q_\phi(z|x_i)}\bigg[\log\Big(\frac{p_\theta(x_i,z)}{q_\phi(z|x_i)}\Big)\bigg] + KL\big(q_\phi(z|x_i)||p_\theta(z|x_i)\big) \\
&\geqslant \mathbb{E}_{q_\phi(z|x_i)}\bigg[\log\Big(\frac{p_\theta(x_i,z)}{q_\phi(z|x_i)}\Big)\bigg]
\end{aligned} \tag{4.1}$$

Let us denote that $\mathcal{L}_{\theta,\phi}(x)$ the *evidence lower bound* (ELBO) such that

$$\mathcal{L}_{\theta,\phi}(x) = \mathbb{E}_{q_\phi(z|x)}\bigg[\log\Big(\frac{p_\theta(x,z)}{q_\phi(z|x)}\Big)\bigg] \tag{4.2}$$

Notice that

$$\log p_\theta(x) = \max_\phi \mathcal{L}_{\theta,\phi}(x). \tag{4.3}$$

Thus, we can solve the maximization problem of ELBO instead of the marginal likelihood. Also if

$$q^*_\phi(z|x) \in \arg\max \mathcal{L}_{\theta,\phi}(x) \tag{4.4}$$

then it satisfies that $q^*_\phi(z|x) = p_\theta(z|x)$ because

$$KL\big(q_\phi(z|x_i)\big|\big|p_\theta(z|x_i)\big) = 0 \quad \text{iff} \quad q_\phi(z|x) = p_\theta(z|x). \tag{4.5}$$

Therefore maximizing the ELBO also gives us the true posterior. Moreover, observe that

$$\begin{aligned}
\mathcal{L}_{\theta,\phi}(x) &= \mathbb{E}_{q_\phi(z|x)}\left[\log\Big(\frac{p_\theta(x,z)}{q_\phi(z|x)}\Big)\right] \\
&= \mathbb{E}_{q_\phi(z|x)}\left[\log\Big(\frac{p_\theta(x|z)p(z)}{q_\phi(z|x)}\Big)\right] \\
&= -KL\big(q_\phi(z|x)\big|\big|p_\theta(z)\big) + \mathbb{E}_{q_\phi(z|x)}\big[p_\theta(x|z)\big].
\end{aligned} \tag{4.6}$$

Thus, we are minimizing the KL-distance between approximating posterior and true prior, meanwhile maximizing expected conditional distribution under approximating posterior distribution.

4.1.2 Reparameterization Trick

In order the solve the variational problem, we compute the gradient of the objective function w.r.t θ and ϕ. However, the gradient w.r.t ϕ is in general hard to compute because the expectation also depends on ϕ.

$$\begin{aligned}
\nabla_\phi \mathcal{L}_{\theta,\phi}(x) &= \nabla_\phi \mathbb{E}_{q_\phi(z|x)}\big[\log p_\theta(x,z) - \log q_\phi(z|x)\big] \\
&\neq \mathbb{E}_{q_\phi(z|x)}\big[\nabla_\phi(\log p_\theta(x,z) - \log q_\phi(z|x))\big]
\end{aligned} \tag{4.7}$$

Therefore, we need to factorize the randomness taking expectation out of the parameter ϕ. So we consider

$$\tilde{z} = g(\epsilon, \phi, x) \tag{4.8}$$

where g is a differentiable function and ϵ is independent with ϕ and x. In this case, for all differentiable function f

$$\begin{aligned}\nabla_\phi \mathcal{L}_{\theta,\phi}(x) &= \nabla_\phi \mathbb{E}_{q_\phi(z|x)}\Big[f(x,z)\Big] \\ &= \nabla_\phi \mathbb{E}_{g(\epsilon,\phi,x))}\Big[f(x,z)\Big] \\ &= \nabla_\phi \mathbb{E}_\epsilon\Big[f\big(x,g(\epsilon,\phi,x)\big)\Big] \\ &= \mathbb{E}_\epsilon\Big[\nabla_\phi f\big(x,g(\epsilon,\phi,x)\big)\Big]\end{aligned} \tag{4.9}$$

Let $\mathbf{NN}_\phi$ and $\mathbf{NN}_\theta$ be two neural networks, then by reparameterization trick, we define

$$\tilde{z} = g(\epsilon, \mathbf{NN}_\phi(x)) \tag{4.10}$$

Thus, we compute the gradient by backward propagation and Monte Carlo

$$\begin{aligned}\nabla_\phi \mathcal{L}_{\theta,\phi}(x) &= \mathbb{E}_\epsilon\Big[\nabla_\phi f\big(x,g(\epsilon,\mathbf{NN}_\phi(x))\big)\Big] \\ &\approx \frac{1}{L}\sum_{i=1}^{L}\nabla_\phi f\big(x,g(\epsilon_i,\mathbf{NN}_\phi(x))\big)\end{aligned} \tag{4.11}$$

In some case, the KL-divergence term even has a analytic formula.

Example 4.1. *Consider a deep latent variable model with gaussian prior and gaussian conditional distribution:*

$$\begin{gathered}q_\phi(z|x) = \mathcal{N}\Big(z;\mu_x,\sigma_x^2\Big) \\ \mu_x = [\mu_1,\cdots,\mu_d]^T, \quad \sigma_x^2 = Diag([\sigma_1^2,\cdots,\sigma_d^2]) \\ p_\theta(z) = \mathcal{N}\Big(z;0,I_d\Big)\end{gathered} \tag{4.12}$$

Then

$$\begin{aligned}-KL\big(q_\phi(z|x)\big|\big|p_\theta(z)\big) &= \mathbb{E}_{q_\phi(z|x)}\big[\log p_\theta(z)\big] - \mathbb{E}_{q_\phi(z|x)}\big[\log q_\phi(z|x)\big] \\ &= \int \log p_\theta(z) q_\phi(z|x) dz - \int \log p_\theta(z|x) q_\phi(z|x) dz \\ &= \frac{1}{2}\sum_{i=1}^{d}\Big(1+\log(\sigma_i^2) - \mu_i^2 - \sigma_i^2\Big)\end{aligned} \tag{4.13}$$

4.2 Variations of VAEs

The classical VAEs assume gaussian prior and ELBO loss. There are multiple variations of classical VAEs suitable for different scenario. In this section we briefly introduce some variations of VAEs which we use as a block of our market generator in numerical experiments .

4.2.1 Conditional VAEs

Condition VAEs [31] are typically used when your data points are labeled and you want to generate data points in a specific class. In financial time-series generating task, we aim at generating the distribution of future stock process given the information known, namely the conditional distribution of time-series. Therefore, we apply the conditional VAEs with conditions being the information known and data being the time series. Let c be conditions, x be time series, and z be latent variable. Let $\mathbf{NN}_\phi$ and $\mathbf{NN}_\theta$ be two neural networks. Then conditional VAEs has the following structure

$$\begin{aligned} q_\phi(z|x,c) &= q\Big(z; \mathbf{NN}_\phi(x,c)\Big) \\ p_\theta(z|c) &= p(z) \\ p_\theta(x|z,c) &= p\Big(x; \mathbf{NN}_\theta(z,c)\Big) \end{aligned} \tag{4.14}$$

$$\max_{\theta,\phi} \sum_{i=1}^{m} \mathcal{L}_{\theta,\phi}(x_i) = \max_{\theta,\phi} \sum_{i=1}^{m} \mathbb{E}_{q_\phi(z|x_i,c_i)} \left[\log\Big(\frac{p_\theta(x_i,z|c_i)}{q_\phi(z|x_i,c_i)}\Big) \right] \tag{4.15}$$

Or we can let $y = (x,c)$ then

$$\max_{\theta,\phi} \sum_{i=1}^{m} \mathcal{L}_{\theta,\phi}(y_i) = \max_{\theta,\phi} \sum_{i=1}^{m} \mathbb{E}_{q_\phi(z|y_i)} \left[\log\Big(\frac{p_\theta(y_i,z|c_i)}{q_\phi(z|y_i)}\Big) \right] \tag{4.16}$$

This is mathematically the same by adding the reconstruction loss between c and reconstructed c, which is zero, because identity gives reconstruction. However, it becomes different if the reconstruction of c and x share some layers of neural network. In this case, there is a tradeoff between expressive capacity and zero reconstruction loss of c.

4.2.2 $\beta-$VAEs

$\beta-$VAEs [16] add an adjustable parameter β to balance the KL-divergence term and the likelihood term in the ELBO loss. We denote $\mathcal{L}^{\beta}_{\theta,\phi}$ the β-ELBO loss such that

$$\mathcal{L}_{\theta,\phi} = -\beta KL\big(q_\phi(z|x)\big|\big|p_\theta(z)\big) + \mathbb{E}_{q_\phi(z|x)}\big[p_\theta(x|z)\big]. \tag{4.17}$$

An interpretation of β-VAEs is that $\mathcal{L}_{\theta,\phi}$ is a log likelihood with penalty on KL-distance between posterior and prior. This similar to Ridge regression which maximizes the log likelihood with penalty on L^2-distance between parameters to zero. β-VAEs is also closely related to the information bottleneck principle, see [16].

4.2.3 Student-VAEs

In financial scenario, heavy tail distribution are commonly observed in the returns of stock prices. Also, normal priors could be outlier-resistant and never reject outliers in data modeling. However Student's t-distributions can generate heavy tails is used in conditional distribution and can reject outliers if used as prior. This motivates us to implement a VAE with Student's t distribution in prior [2] as well as conditional distribution. Recall student's t distribution

$$p(z;\mu,\sigma,\nu) = \frac{\Gamma(\frac{\mu+p}{2})}{\Gamma(\frac{\mu}{2})\sqrt{(\pi\nu)^p}\sigma}\left(1+\frac{(z-\mu)^T\sigma^{-2}(z-\mu)}{\nu}\right)^{-\frac{\nu+p}{2}} \tag{4.18}$$

We consider the following variational autoencoder

$$\begin{aligned} q_\phi(z|x) &= \mathrm{St}\Big(z;\mathbf{NN}_{\phi,\mu}(x),\mathbf{NN}_{\phi,\sigma}(x),\nu_\theta\Big) \\ p_\theta(z) &= \mathrm{St}\Big(z;0,I_d,\nu_\theta\Big) \\ p_\theta(x|z) &= \mathrm{St}\Big(x;\mathbf{NN}_\theta(z),I_d,1\Big) \end{aligned} \tag{4.19}$$

In this case we have no simple expression of ELBO as normal case, but still we have an analytic expression of KL-divergence term, see [1]. More general, we can consider when prior is stacked independent student-t distribution with different parameter ν, i.e.

$$\begin{aligned} q_\phi(z|x) &= \prod_i \mathrm{St}\Big(z;\mathbf{NN}_{\phi,\mu_i}(x),\mathbf{NN}_{\phi,\sigma_i}(x),\mathbf{NN}_{\phi,\nu_i}(x)\Big) \\ p_\theta(z) &= \prod_i \mathrm{St}\Big(z;0,1,\nu_{\theta,i}\Big) \\ p_\theta(x|z) &= \mathrm{St}\Big(x;\mathbf{NN}_\theta(z),I_d,1\Big) \end{aligned} \tag{4.20}$$

In this case, we no longer have a analytic formula of the KL-divergence term, so we can use monte carlo to approximate the expectation.

4.2.4 Gaussian process VAEs

All variations of VAEs above are designed for samples and latent variables on Euclidean space. Gaussian process VAEs [10]generalize the idea of variational autoencoder by choosing a latent time series space with prior to be a distribution of gaussian process. Let $(\mathbf{x}_t)_{t\in J} = \mathbf{x} \in \text{Seq}(\mathbb{R}^d)$ be a time series random variable and $(\mathbf{z}_t)_{t\in J'} = \mathbf{z} \in \text{Seq}(\mathbb{R}^{d'})$ be a latent time series random variable. We consider the following model

$$\begin{aligned} \mu_i &= \mathbf{NN}_{\phi,\mu,i}(\mathbf{x}) \\ \Lambda_i &= \mathbf{NN}_{\phi,\Lambda,i}(\mathbf{x}) \\ q_\phi(\mathbf{z}|\mathbf{x}) &= \prod_i \mathcal{GP}(\mathbf{z}; \mu_i, \Lambda_i^{-1}) \\ p_\theta(\mathbf{z}) &= \mathcal{GP}(\mathbf{z}; m(\cdot), k(\cdot,\cdot)) \\ p_\theta(\mathbf{x}_t|\mathbf{z}) &= \mathcal{N}\Big(\mathbf{x}_t; \mathbf{NN}_\theta(\mathbf{z}), \sigma^2 I_d\Big) \end{aligned} \tag{4.21}$$

The encoding part from $\mathbf{x}$ to $\mathbf{z}$ is almost the same as VAEs but instead of generating independent normal random variables, we generate independent gaussian process and stack them to multi-dimensional gaussian process. The decoding part is applying a neural network transformation to the latent time series. The prior can be chosen as any gaussian process. An observation in [10] is that classical RBF kernel does not reflect the dynamics of data very well. However a mixture kernel

$$\begin{aligned} &k(\lambda|\alpha,\beta,\lambda) = \int p(\lambda|\alpha,\beta) k_{RBF}(r,\lambda) d\lambda^{(\alpha-1)} \\ &k_{RBF}(r,\lambda) = \exp(-\lambda r^2/2) \\ &p(\lambda|\alpha,\beta) \propto \lambda \exp(\alpha\lambda/\beta) \end{aligned} \tag{4.22}$$

is successful if used in the context of robust dynamic topic modeling where similar multi-scale time dynamics occurs.

5. Market Generator

In this chapter, we combine signature features and variational autoencoders introduced above to construct a market generator. Recall our problem setting: By observing one realisation of stochastic process, we wish to infer the distribution of the process and stimulate paths under the learnt distribution. For example, given the S&P index of last 12 months, we wish to construct market generator which generates possible paths of S&P index of next months for us, and hopefully generating paths of S&P index to the next two months. Our algorithm can be subdivided into the following five parts, of which we present a brief overview below:

(Step 1): **Time series splitting:** Split single realisation of path $(x_{0:N})$ to multiple subpaths $(x_{i:i+n})_{i=1}^{m}$. This subdivision helps to generate multiple samples for training.

(Step 2): **Path to signature features:** Applying the signature feature map to m subpaths $(x_{i:i+n})_{i=1}$ gives m samples on $\mathbb{R}^d$, where d depends on the choice of siganture feature map.

(Step 3): **Variational autoencoder on signature features:** Train a variational autoencoder on the signature feature space and generate samples on the siganture feature space.

(Step 4): **Signature features to path:** Inverse samples on signature feature space to samples on paths space.

(Step 5): **Evaluation of model:** Compare the distribution of generated paths to true paths.

5.1 Time series splitting

In many generative machine learning tasks, such as images generating, abundant of data points are available for training. However, for financial time series, scarcity of data can be a systemic problem due to the following reasons. Firstly, financial time series data has time inconsistency problem. Time inconsistency problem is well studied in behavioral economics where preferences change from time to time. Inconsistency in dynamic implies that if we use very old data, we are actually using data from a different dynamic, in another word, samples from a different distribution [8]. This is not a problem in image classification because a image of cat drawn 100 years before also looks like a cat today, but the dynamic of stocks market 100 years before can be greatly different from the one today. The second problem is that financial data usually suffers from roughness in paths. Therefore, a frequent observation results in a rougher path which leads to noise as well as difficulties in stable numerical methods. The last problem is that usually we only have one realisation of path in financial scenario, unlike speech generating multipe experiments can be conducted. Therefore, if we need i.i.d samples, we need to split the path and assume further stationary condition to guarantee the independence among samples. If we consider stock prices under Black-scholes model

$$S_t = S_0 \exp\left\{ (\mu - \frac{\sigma^2}{2})t + \sigma W_t \right\}$$

and if we consider the log-return X_t i.e.

$$X_t = (\mu - \frac{\sigma^2}{2})t + \sigma W_t$$

then we consider the difference of log return

$$X_{t+\triangle t} - X_t = (\mu - \frac{\sigma^2}{2})\triangle t + \sigma(W_{t+\triangle t} - W_t).$$

Since Brownian motion has increment increment, splitting the difference of log return gives i.i.d samples even we choose length of subpath to be 1. Inspired by this idea, we preprocess the stock price by taking the difference of log return before splitting. Given the preprocessed path, the longer we choose the length of subpath, fewer samples we have. The more samples we have, the more we violate the independence. Therefore, there is actually a tradeoff in the choice of length of subpaths. In our experiments, we subdivide the time seires into intervals: (i) 1 day (ii) 1 week (iii) 1 month. Mathematically speaking, given a path $x_{0:N}$ of with $N+1$ times observation.

$$x_i = x_{t_{i-1}:t_i}, \quad i = 1, \cdots, m$$

where $0 = t_0 \leqslant t_1 \leqslant \cdots \leqslant t_{m+1} = N$. Note that this is not a equal length division since days in months different and there might be some days data missing. However, this inconsistency in division as well as missing data is not a problem to our model, because we will later consider the log-signature feature of each subpath. The dimension of log-siganture only depends on our truncation order and the calculation of log-signature is robust to missing data.

5.2 Path to signature features

After splitting path to samples of subpaths, we apply a feature map on each subpath and here our feature map is the log-signature. We choose log-signature motivated by the following reasons

(i) Siganture type features stores path information in an efficient way by Example 3.14.

(ii) Expected normalized signature characterise the law of stochastic process by Corollary 3.25.

(iii) Signature can be directly used to compute signature kernel evaluating the MMD distance between distribution on paths by (3.48).

(iv) log-sigature has the same information as sigature but store them in a more compact way, by Proposition 3.21.

(v) The image of signature space is not a linear subspace of tensor space but the range of truncated log-signature is the truncated Lie series over E up to the same order by Theorem 3.20.

(vi) Tensor-algebra exponential of the generated log-signatures recover the group-like (shuffle product) property of signature, see [28].

(vii) the numerical log-signature is more robust than siganture, by Example

(viii) signature feature eliminates pricing and hedging ambiguities problem addressed in and signature can be directly used for pricing, see.

We additional apply a lead-lag transformation [9] to path before computing the log-signature features of the subpath. This is because the signature of a path after lead-lag transformation helps to capture the volatility of a path, which is of vital imporatance in finance.

5.3 Variational autoencoder on signature features

Given samples on log-signature space denoted by X_i, we build a generative model and train it with samples. Arguably, the most popular generative models are GANs and VAEs. We choose VAEs motivated by the following considerations.

(i) VAEs require considerably less data for training than GANs

(ii) VAEs are in genral more stable in training than GANs

(iii) VAEs are less prone to overfitting than GANs

Here we use conditional VAEs choosing the sample of subpath right before as condition, i.e. $c_i = X_{i-1}$. Moreover, we consider student-t prior in order to recover a heavy-tail distribution observed commonly in financial data. Also, we optimize the β-ELBO loss in training to balance the reconstruction loss and KL-divergence. Details of network structure are dicussed with numerical results in chapter 4.

5.4 Signature features to path

After the training of model, we generate samples on the log-sigature space. Now we need to inverse a log-signature to a path. The signature of a path uniquely determines the path itself up to tree-like equivalence but reverting siganture to path is a computationally a highly non-trivial task and currently a topic of active research. In ,they develop an evolutionaty algorithm by mimicking to some extent biological evolution. They start with an initial population of random paths, and iteratively select the paths whose signatures are closest to the target signature, and breed these paths and introduce mutations to generate a new generation of paths until we get a population of paths whose signature are close to the target signature. However, the revolution algorithm is very slow in optimization and reverting task of each signature need a restart of the evolution procedure. Therefore, we develop a new neural network based solution to inverse the siganture. Our method is much faster than evolution algorithm and can be trained to deal with a collection of signatures once trained. Our approach is to construct an autoencoder with encoder be a neural network and decoder be the log-siganture transformation. We train our autoencoder by minimizing the L^2 error between input log-signature with the output of autoencoder. Once trained, we consider the encoder as a good inverse function of log-signature. This method can be trained w.r.t one samples in order to reverse this single sample in log-signature space. Moreover, the parameter in encoder does not change too much

when we are training it to reverse different samples. Therefore, we can leverage the parameter trained before to start our training for the next sample, which accelerates the training procedure. Also, we can pre-trained the model on a collection of log-siganture and use the pre-trained model to inverse a sample without training.

5.5 Evaluation of model

We evaluate the goodness of market generator by computing the MMD difference between the true distribution $\mathbb{P}$ and the generated distribution $\mathbb{Q}$ on path space.

$$d_{\mathcal{G}}(\mathbb{P},\mathbb{Q}) = \mathbb{E}\big[\mathbf{k_{Sig}}(X,X')\big] - 2\mathbb{E}\big[\mathbf{k_{Sig}}(X,Y)\big] + \mathbb{E}\big[\mathbf{k_{Sig}}(Y,Y')\big] \tag{5.1}$$

where X, Y, X', Y' are independent with $X, X' \sim \mathbb{P}$ and $Y, Y' \sim \mathbb{Q}$. For discrete approximation, we compute the MMD test statistic:

$$T_U^2(X_1,\cdots,X_m;Y_1,\cdots,Y_m) \coloneqq \frac{1}{m(m-1)}\sum_{i,j,i\neq j}\mathbf{k_{Sig}}(X_i,X_j) - \frac{2}{mn}\sum_{i,j}\mathbf{k_{Sig}}(X_i,Y_j) + \frac{1}{n(n-1)}\sum_{i,j,i\neq j}\mathbf{k_{Sig}}(Y_i,Y_j)$$

[14, 15] prove that T_U^2 is an unbiased estimators for $d_{\mathcal{G}}$ and provide a confidence interval of rejecting $\mathbb{P} = \mathbb{Q}$.

Theorem 5.1. *Assume that $m = n$, and $0 \leqslant \mathbf{k_{Sig}}(\cdot,\cdot) \leqslant K$, then*

$$\mathbb{P}\bigg\{T_U^2(X_1,\cdots,X_m;Y_1,\cdots,Y_m) - d_{\mathcal{G}}(\mathbb{P},\mathbb{Q}) > t\bigg\} \leqslant \exp\Big(\frac{-t\lfloor m/2\rfloor}{8K^2}\Big)$$

They also pointed out that the choice of threshold is conservative and can be improved by using data-dependent bounds. An alternative is to apply a permutation test. We refer to the MMD testing literature for many more details and improvements [14, 15, 7, 30, 17, 32].

6. Numerical Implementation

In this chapter, we present numerical implementations of our algorithm. We would first preset the numerical result of market generator introduced last chapter on both real market data as well as time series generated by models. Moreover, we would present some numerical comparisons between different evaluation metrics and inversion algorithms and blenchmark our contribution to blenchmarks. At the end of this chapter, we supplent with numerical examples highlighting the efficiency of signature features and provide a new approach to stimulate time series.

6.1 Market generator

6.2 Evaluation

6.3 Inversion algorithm

6.4 Path stimulation with signature

Let us consider numerical stimulations of the following stochastic differential equation

$$dX_t = \sum_{k=1}^{m} B^k X_t dW_t^k, \quad X_t \in \mathbb{R}^n, W_t \in \mathbb{R}^m, B^k \in \mathbb{R}^{n\times n}$$

where W_t is a Brownian motion on $\mathbb{R}^m$. We choose $m = n = 2$ and

$$B^1 = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix}, \quad B^2 = \begin{bmatrix} 0.4 & 0.3 \\ 0.2 & 0.1 \end{bmatrix}.$$

Then we stimulate the differential equation by signature

$$\begin{aligned} X_t &= \sum_{d=0}^{\infty} \sum_{i_1,\cdots,i_d=1}^{n} \left(\int_{0\leqslant t_1\leqslant\cdots\leqslant t_d\leqslant t} dW_{t_1}^{i_1}\cdots dW_{t_d}^{i_d} \right) B^{i_d}\cdots B^{i_1} X_0 \\ &= \sum_{d=0}^{\infty} \sum_{i_1,\cdots,i_d=1}^{n} \mathbf{Sig}_{[0,t]}^{i_1,\cdots,i_d}(W) B^{i_d}\cdots B^{i_1}\cdot X_0 \end{aligned}$$

Here we approximate the signature by growing signature. This is important because if we use higher order, we implicitly assume Stratonovich integral which is not the case here. Or we are using a Milstein scheme and implicitly assume the commutativity of B^1 and B^2 here which is also not the case because

$$\begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix}\begin{bmatrix} 0.4 & 0.3 \\ 0.2 & 0.1 \end{bmatrix} - \begin{bmatrix} 0.4 & 0.3 \\ 0.2 & 0.1 \end{bmatrix}\begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix} = \begin{bmatrix} -0.05 & -0.15 \\ 0.15 & 0.05 \end{bmatrix} \neq 0$$

Unfortunately, all python packages I know available so far (including esig, iisignature, and Signatory) only support truncated discrete signature. We blenchmark with Euler approximation on a very fine grid.

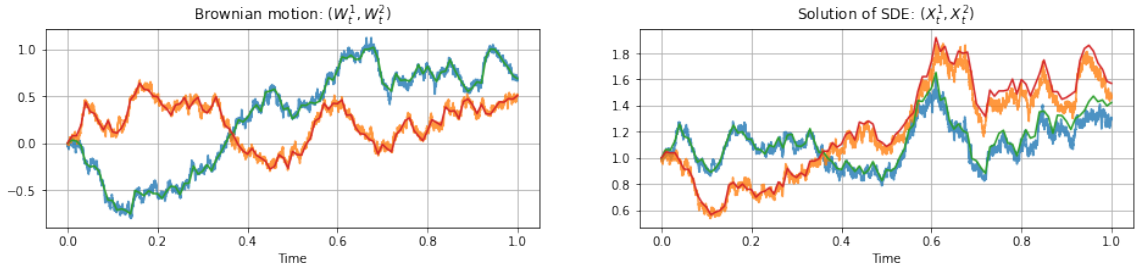

Figure 6.1: Solution of SDE stimulated with Euler method on fine grid

Below we compare the stimulation between growing discrete signature and truncated discrete signature.

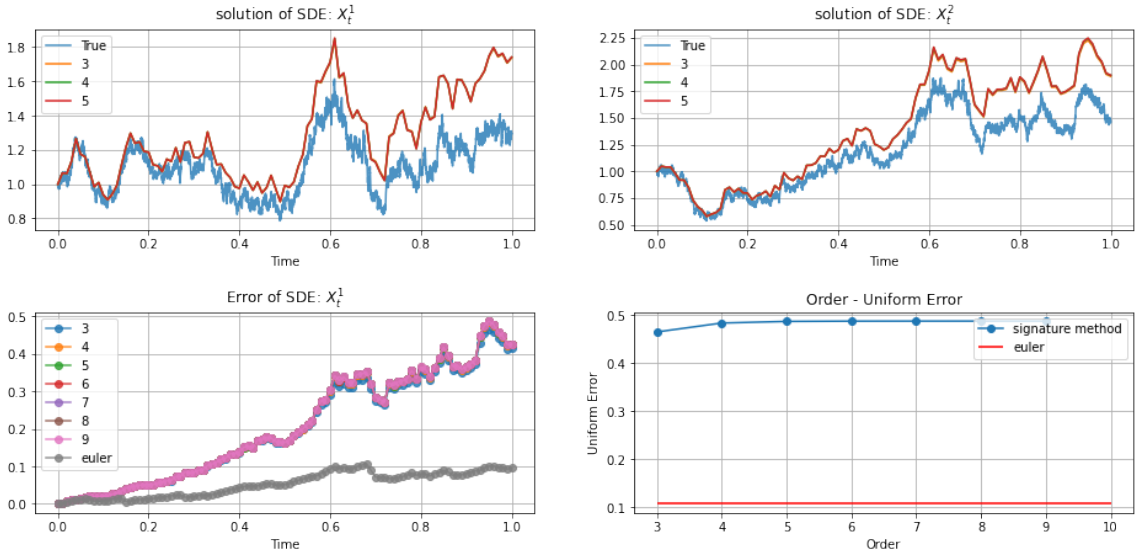

Figure 6.2: Truncated discrete signature

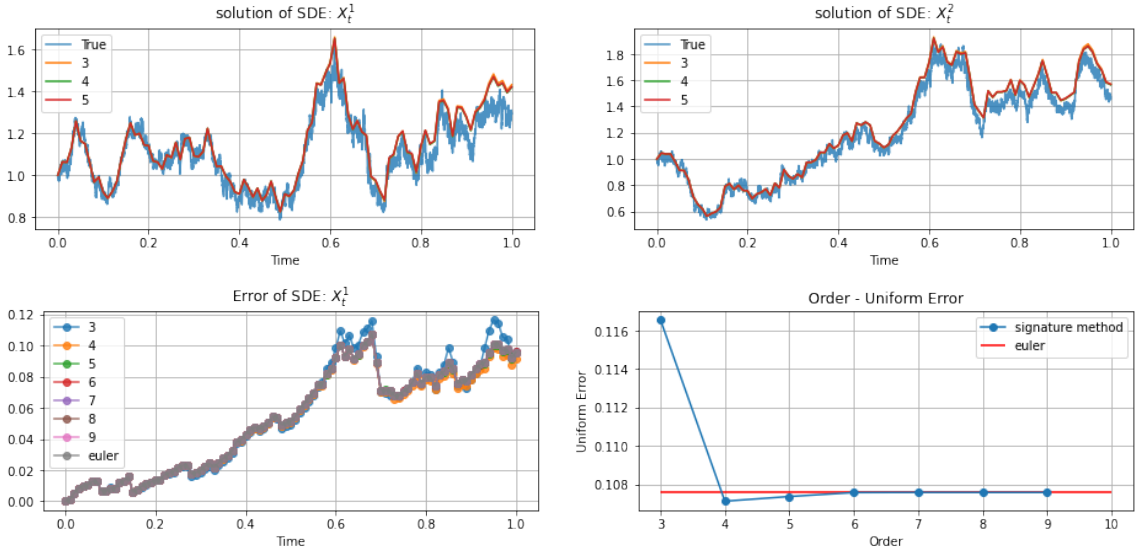

Figure 6.3: Growing discrete signature

Growing discrete signature outperform the truncated signature. As the order of growning discrete signature increase, the performance tends to the Euler method.

This is useful because if we know the signature of our control at T, then multiplying the signature with vectorfield gives a stimulation as good as Euler stimulation running from 0 to T. The key here is the linear relation between signature and the value of SDE. Inversely, this linear relation can be easily learnt by a simple linear regression.

6.5 Learning from path to path

Consider controlled differential equations

$$dX_t = \sum_{k=1}^{m} V^k(X_t) dW_t^k, \quad X_t \in \mathbb{R}^d \tag{6.1}$$

By generalized Taylor's expansion we have

$$\begin{aligned} X_t^{(i)} &= \sum_{m=0}^{\infty} \sum_{i_1,\cdots,i_m=1}^{d} \mathbf{Sig}_{[0,t]}^{i_1,\cdots,i_m}(W) \sum_{j=1}^{d} M_{i,j}^{i_1,\cdots,i_m} X_s^{(j)} \\ &= \sum_{m=0}^{\infty} \sum_{i_1,\cdots,i_m=1}^{d} \sum_{j=1}^{d} M_{i,j}^{i_1,\cdots,i_m} \mathbf{Sig}_{[0s,t]}^{i_1,\cdots,i_m}(W) X_0^{(j)} \\ &= \langle M_i, \mathbf{Sig}_{[s,t]} \otimes X_s \rangle \end{aligned} \tag{6.2}$$

Where $M_{i,j}^{i_1,\cdots,i_m}$ are matrixs depending on s. If we let $\mathcal{X} = \mathbf{Sig}_{[s,t]} \otimes X_s$ and $\mathcal{Y} = X_t$ Therefore, we can learn the relation in (6.2) by a linear regression on $(\mathcal{X}, \mathcal{Y})$. Now, we consider $m = d = 2$ and

$$V^1(X) = \begin{bmatrix} 2|X^{(2)}|^{0.7} \\ X^{(2)} \end{bmatrix}, \quad V^2(X) = \begin{bmatrix} 2|X^{(2)}|^{0.7} \\ 0 \end{bmatrix}, \quad X_0 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

We stimulate the real solution X with Euler scheme on $[0, 1]$ with 2000 grid points.

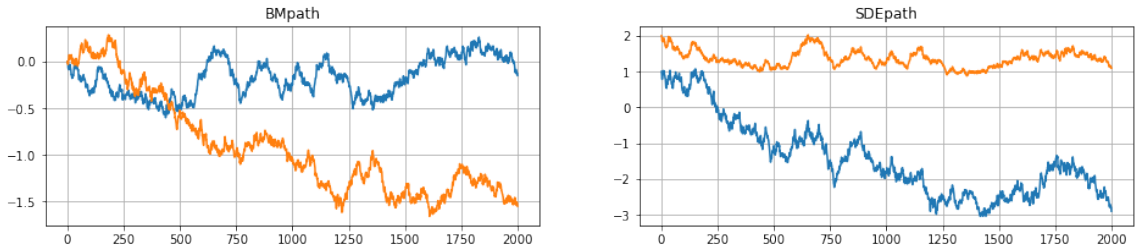

Figure 6.4: Solution of SDE stimulated with Euler method

We choose first $t = 1, \cdots 1000$ to be training samples, and $t = 1001, \cdots 2000$ to be test samples. We use ridge regression to learn matrixes $M_{i,j}^{i_1,\cdots,i_m}$ on training samples, and generate path on testing samples with learnt matrixs. We compared the generated path with true path under the same random seed.

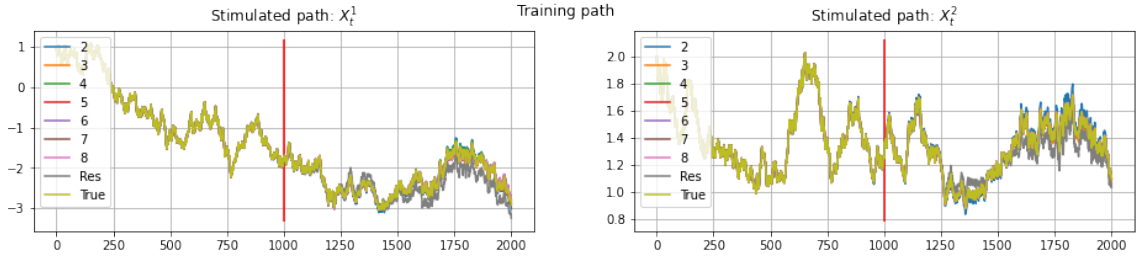

Figure 6.5: Training path

Furthermore, we can generate path starting from 0 with another random seed rs and compare the generated path with true path under the same random seed rs.

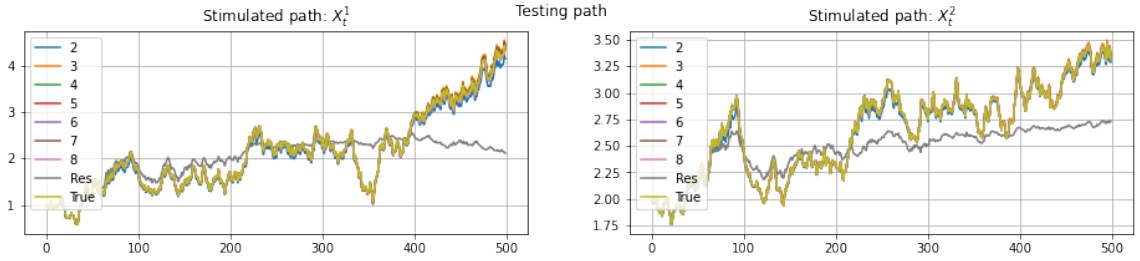

Figure 6.6: Testing path

Now we present the relation between error and order of growing discrete signature

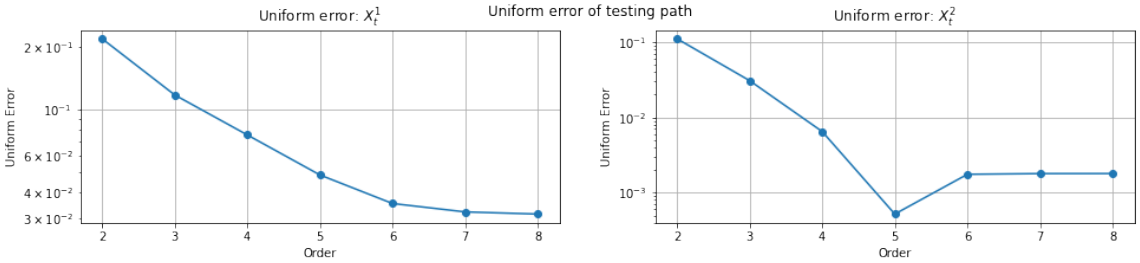

Figure 6.7: Order - Error

7. Conclusions and Future Work

compared with gan

8. Appendix

Bibliography

[1] Abiri, N., and Ohlsson, M. The advantage of using student's t-priors in variational autoencoders.

[2] Abiri, N., and Ohlsson, M. Variational auto-encoders with student's t-prior. *arXiv preprint arXiv:2004.02581* (2020).

[3] Boedihardjo, H., Geng, X., Lyons, T., and Yang, D. The signature of a rough path: uniqueness. *Advances in Mathematics 293* (2016), 720–737.

[4] Bonnier, P., Liu, C., and Oberhauser, H. Adapted topologies and higher rank signatures. *arXiv preprint arXiv:2005.08897* (2020).

[5] Buehler, H., Horvath, B., Lyons, T., Perez Arribas, I., and Wood, B. A data-driven market simulator for small data environments. *Available at SSRN 3632431* (2020).

[6] Chevyrev, I., and Oberhauser, H. Signature moments to characterize laws of stochastic processes. *arXiv preprint arXiv:1810.10971* (2018).

[7] Chwialkowski, K., Strathmann, H., and Gretton, A. A kernel test of goodness of fit. In *International conference on machine learning* (2016), PMLR, pp. 2606–2615.

[8] Cont, R. Empirical properties of asset returns: stylized facts and statistical issues.

[9] Flint, G., Hambly, B., and Lyons, T. Discretely sampled signals and the rough hoff process. *Stochastic Processes and their Applications 126*, 9 (2016), 2593–2614.

[10] Fortuin, V., Baranchuk, D., Rätsch, G., and Mandt, S. Gp-vae: Deep probabilistic time series imputation. In *International Conference on Artificial Intelligence and Statistics* (2020), PMLR, pp. 1651–1661.

[11] Foster, J., Lyons, T., and Oberhauser, H. An optimal polynomial approximation of brownian motion. *SIAM Journal on Numerical Analysis 58*, 3 (2020), 1393–1421.

[12] Freeman, J. Probability metrics and the stability of stochastic models. *Journal of the Operational Research Society 43*, 9 (1993), 923–923.

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

[14] Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *The Journal of Machine Learning Research 13*, 1 (2012), 723–773.

[15] Gretton, A., Fukumizu, K., Harchaoui, Z., and Sriperumbudur, B. K. A fast, consistent kernel two-sample test. In *NIPS* (2009), vol. 23, pp. 673–681.

[16] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework.

[17] Jitkrittum, W., Szabó, Z., Chwialkowski, K., and Gretton, A. Interpretable distribution features with maximum testing power. *arXiv preprint arXiv:1605.06796* (2016).

[18] Kidger, P., Foster, J., Li, X., Oberhauser, H., and Lyons, T. Neural sdes as infinite-dimensional gans. *arXiv preprint arXiv:2102.03657* (2021).

[19] Kidger, P., and Lyons, T. Signatory: differentiable computations of the signature and logsignature transforms, on both cpu and gpu. *arXiv preprint arXiv:2001.00706* (2020).

[20] Kidger, P., Morrill, J., Foster, J., and Lyons, T. Neural controlled differential equations for irregular time series. *arXiv preprint arXiv:2005.08926* (2020).

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

[22] Kingma, D. P., and Welling, M. An introduction to variational autoencoders. *arXiv preprint arXiv:1906.02691* (2019).

[23] Király, F. J., and Oberhauser, H. Kernels for sequentially ordered data. *Journal of Machine Learning Research 20*, 31 (2019), 1–45.

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

[25] Liao, S., Lyons, T., Yang, W., and Ni, H. Learning stochastic differential equations using rnn with log signature features. *arXiv preprint arXiv:1908.08286* (2019).

[26] Lyons, T. Rough paths, signatures and the modelling of functions on streams. *arXiv preprint arXiv:1405.4537* (2014).

[27] Lyons, T. J. Differential equations driven by rough signals. *Revista Matemática Iberoamericana 14*, 2 (1998), 215–310.

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

[29] Reutenauer, C. Free lie algebras. In *Handbook of algebra*, vol. 3. Elsevier, 2003, pp. 887–903.

[30] Sejdinovic, D., Sriperumbudur, B., Gretton, A., and Fukumizu, K. Equivalence of distance-based and rkhs-based statistics in hypothesis testing. *The Annals of Statistics* (2013), 2263–2291.

[31] Sohn, K., Lee, H., and Yan, X. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems 28* (2015), 3483–3491.

[32] Sriperumbudur, B. K., Gretton, A., Fukumizu, K., Schölkopf, B., and Lanckriet, G. R. Hilbert space embeddings and metrics on probability measures. *The Journal of Machine Learning Research 11* (2010), 1517–1561.

[33] Toth, C., Bonnier, P., and Oberhauser, H. Seq2tens: An efficient representation of sequences by low-rank tensor projections. *arXiv preprint arXiv:2006.07027* (2020).

[34] Toth, C., and Oberhauser, H. Variational gaussian processes with signature covariances. *arXiv preprint arXiv:1906.08215* (2019).

[35] Toth, C., and Oberhauser, H. Bayesian learning from sequential data using gaussian processes with signature covariances. In *International Conference on Machine Learning* (2020), PMLR, pp. 9548–9560.