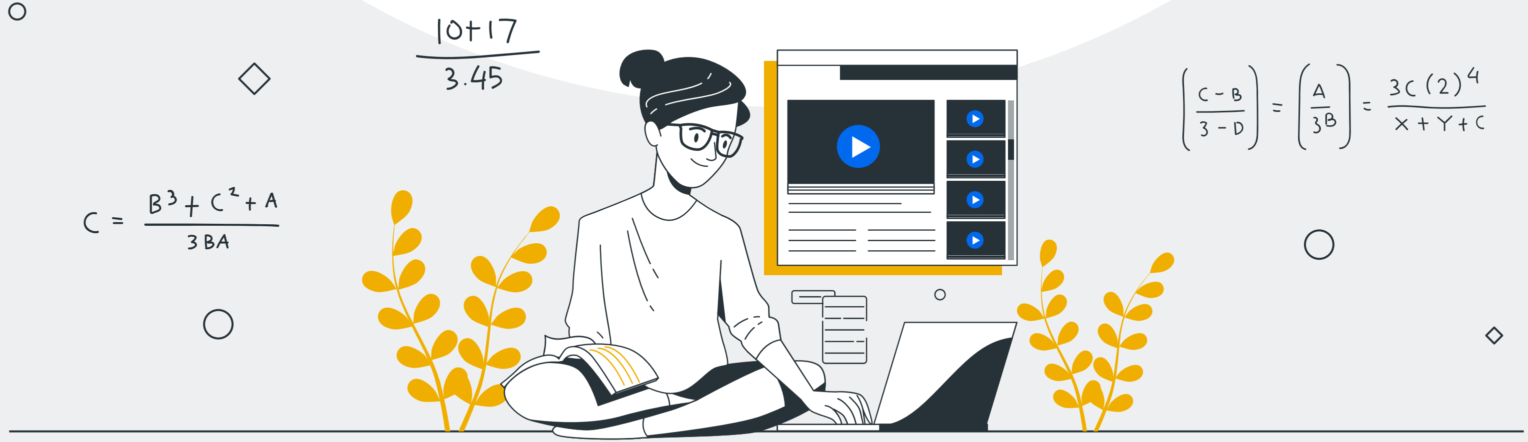
APPLICATION OF SIGNATURES FOR FORECASTING



BAT-ERDENE EGSHIGLEN

2021

SUPERVISOR : TIKOSI KINGA

Table of contents

01 Review

Definition

02 Properties

Important characteristics

03 Application

Forecasting

Review

Definition

DEFINITION: We have a path $X_t : [a,b] \rightarrow \mathbb{R}^m$, then the signature of the path is an infinite series of iterated integrals:

$$S(X)_{ab} = (1, S(X)^1_{ab}, S(X)^2_{ab}, \ldots)$$

Review

$$S(X)^{i}_{a,t} = \int_{a<s<b} dX^{i}_{s} = X^{i}_{t} - X^{i}_{0}$$

$$S(X)^{i,j}_{a,t} = \int_{a<s<b} S(X)^{i}_{a,s}\, dX^{j}_{s} = \int_{a<r<s<t} dX^{i}_{r} dX^{j}_{s}$$

```
pip install iisignature
import iisignature as isig
import numpy as np
data= ([1,1], [2,6], [5,5], [6,3])
isig.sig(data, 2, 1)
output: (array([5., 2.]), array([12.5, -9. , 19. , 2. ]))
```

$$C = \frac{B^3 + C^2 + A}{3BA}$$

Python

Concatenation

Definition 2.3. Let $X : [a, b] \mapsto \mathbb{R}^d, Y : [b, c] \mapsto \mathbb{R}^d$, then the concatenation of X and Y is a path from $[a, c] \mapsto \mathbb{R}^d$:

$$(X * Y)_t = \begin{cases} X_t, & \text{if } t \in [a, b] \\ X_b + (Y_t - Y_b), & \text{if } t \in [b, c]. \end{cases}$$

Chen's identity

Theorem 2.2 (Chen's identity). As usual, let us have two paths $X : [a,b] \mapsto \mathbb{R}^d, Y : [b,c] \mapsto \mathbb{R}^d$, then

$$S(X * Y)_{a,c} = S(X)_{a,b} \otimes S(Y)_{b,c}. \tag{9}$$

Theorem

Definition 2.4. A path $X : [0,1] \mapsto \mathbb{R}^d$ is tree-like, if $\exists f : [0,1] \mapsto [0,\infty) : f(0) = f(1) = 0$ and $\forall s,t \in [0,1], s \leq t$:

$$\|X_s - X_t\| \leq f(s) + f(t) - 2 \inf_{u \in [s,t]} f(u). \tag{13}$$

Theorem 2.3. Assume $X, Y : [a,b] \mapsto \mathbb{R}^d$, then

$$\forall t \in [a,b] : X_t = Y_t \implies \forall k \in \{1,\ldots,d\} : S^k(X) = S^k(Y).$$

Theorem

Theorem 2.3.2 (Uniqueness). *Let X be a continuous path with bounded variation. Then,*

- $S(X) = \mathbf{1}$ *if and only if X is tree-like.*
- *The signature $S(X)$ is unique up to tree-like equivalence.*

03

Application

How do we use the signature
in real life?

I would like to approximate a function, what should I do?

Taylor's theorem

But what if we do not have a differentiable function?

Approximation

$$f(X) = c_0 + c_1 S(X)^{1}_{a,b} + c_2 S(X)^{2}_{a,b} + c_{1,1} S(X)^{1,1}_{a,b} \ldots .$$

Process

How to forecast the given data?

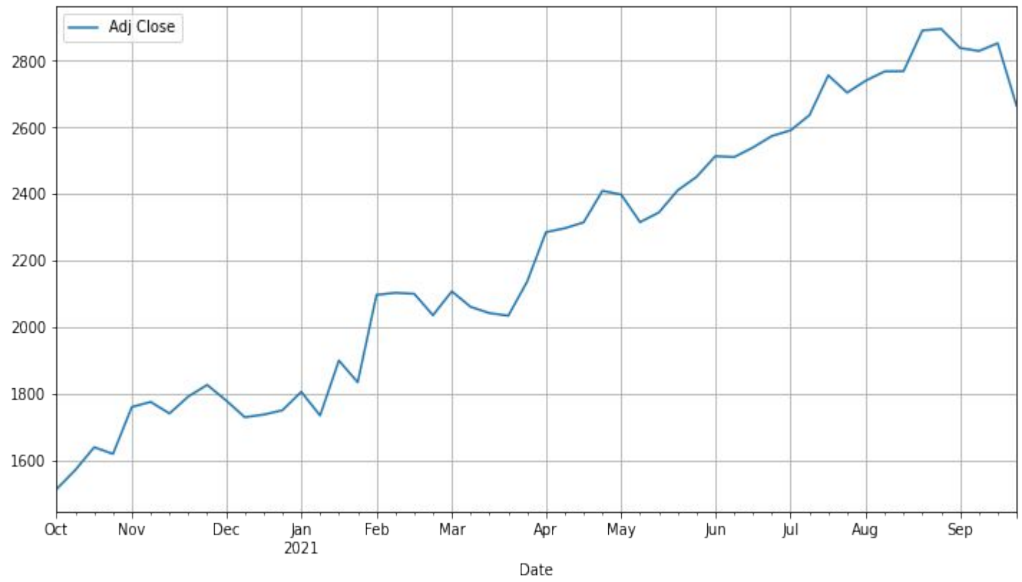
01 We have a data stream

02 Embed it to R^d

03 Compute the iterated integrals

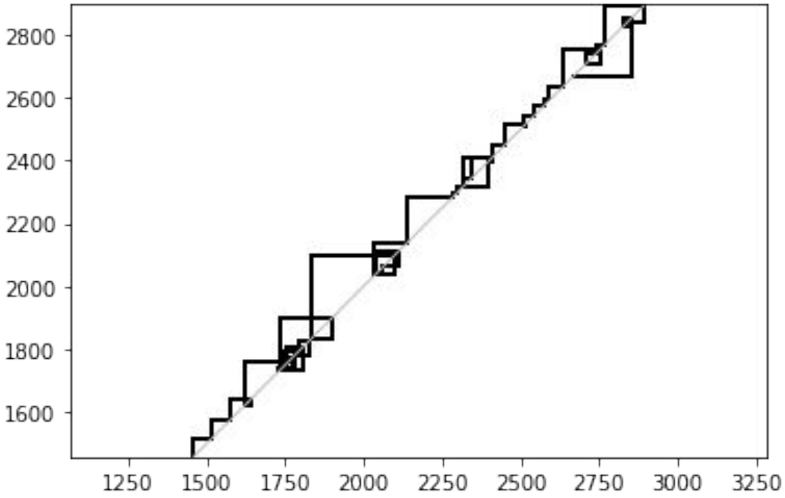
04 Use the resulting set of features for analysing the data/forecasting

STEP 1. Import data



STEP 2. Embed data to 2D

$$C = \frac{B^3 + C^2 + A}{3BA}$$



Our stream would be the concatenated lead-lag data.

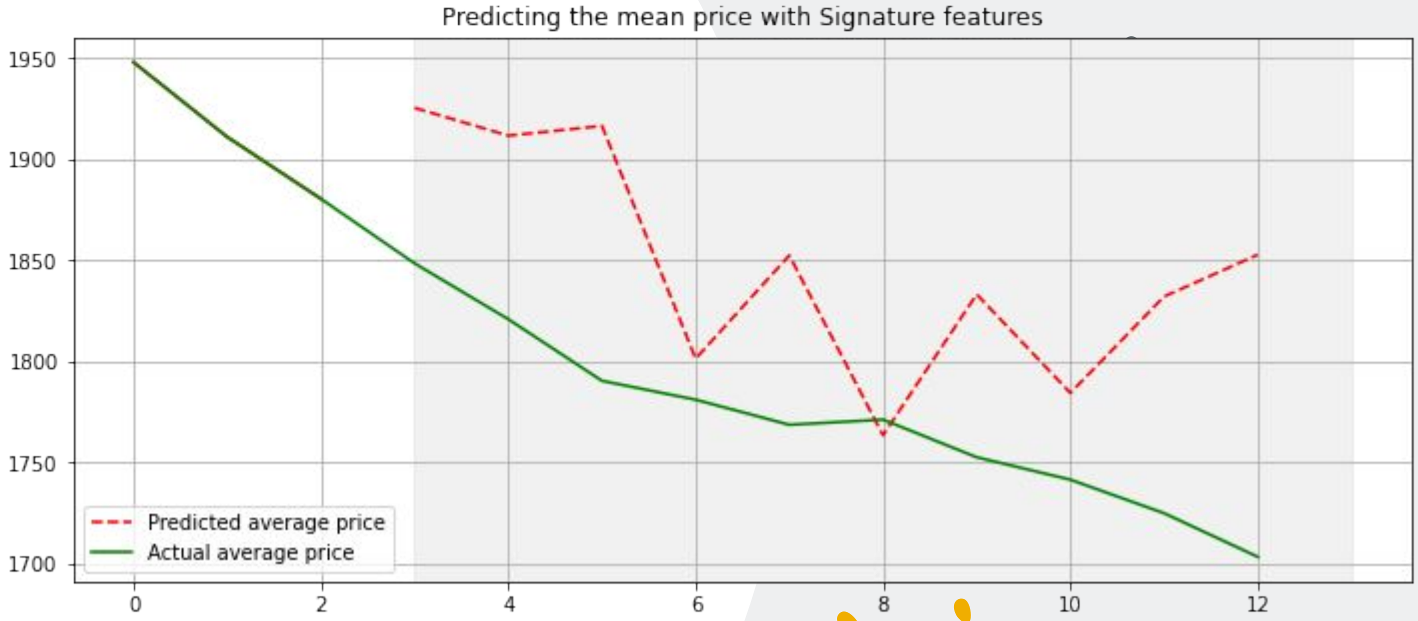
Next we will calculate the level 2 signature of the stream.

STEP 3. Compute signature

	0	1	2	3	4	5	6
0	0.707317	-603.390137	-603.390137	0.250149	-327.558707	-334.917123	-99.229439
1	0.707317	-744.119873	-744.119873	0.250149	-278.150453	-287.225086	-248.178237
2	0.707317	-792.410035	-792.410035	0.250149	-310.112810	-319.776347	-250.372337
3	0.707317	-737.280029	-737.280029	0.250149	-245.991776	-254.982996	-275.498976
4	0.707317	-791.389893	-791.389893	0.250149	-225.248577	-234.899673	-334.515006

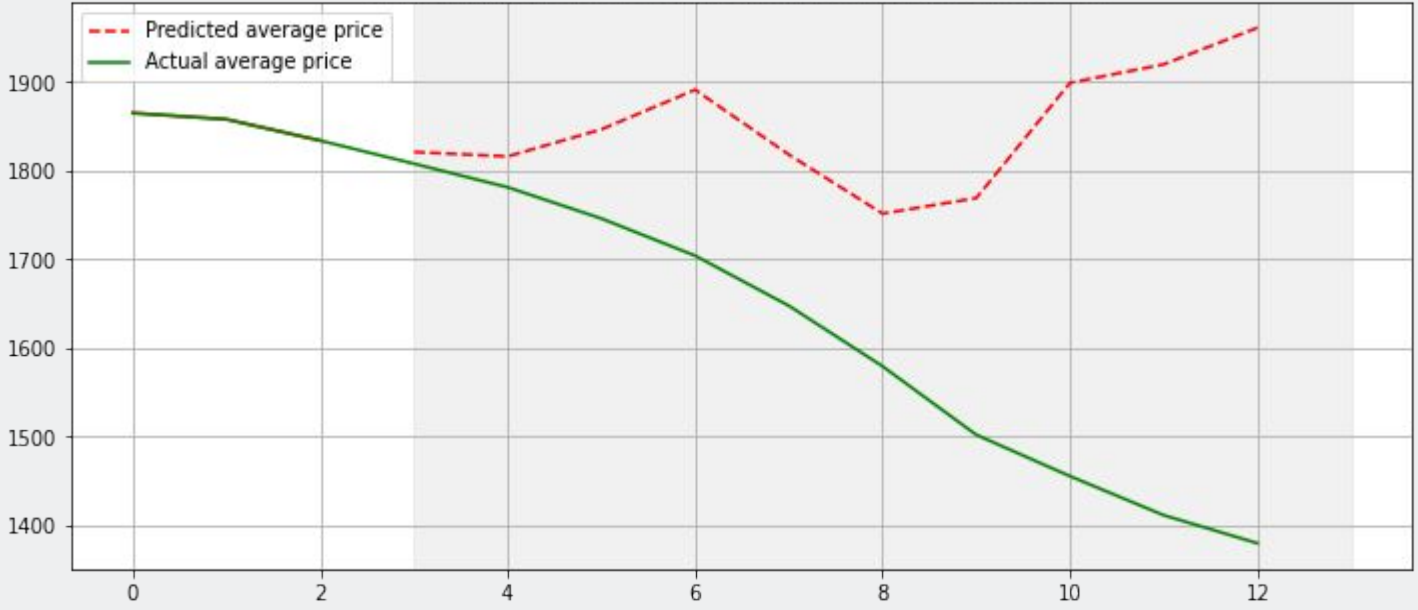
STEP 4. PREDICTION

$$C = \frac{B^3 + C^2 + A}{3BA}$$



STEP 4. PREDICTION

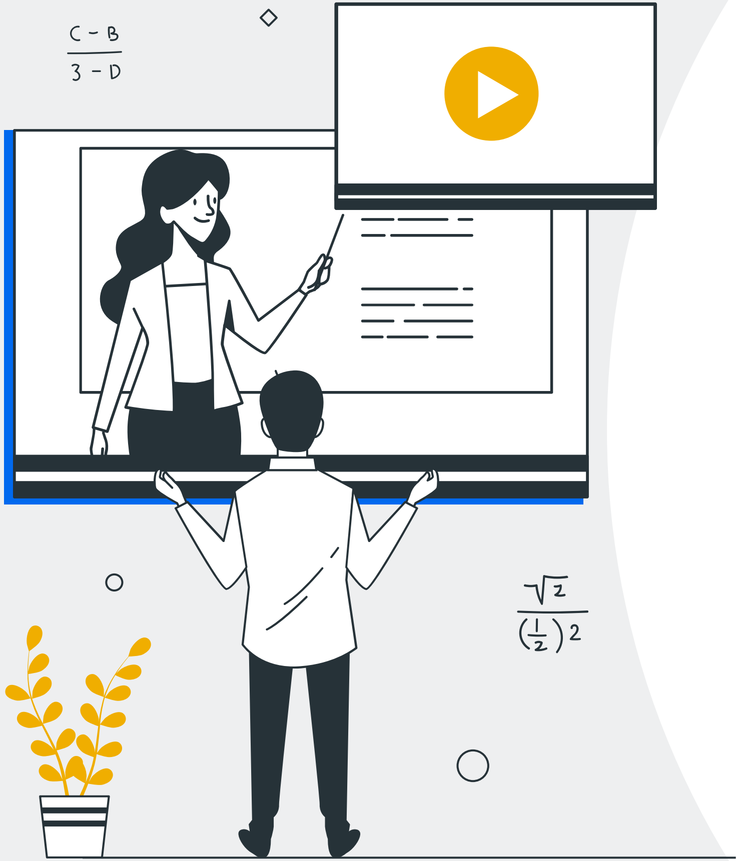
Predicting the mean price of SMSN.IL with Signature features



CONCLUSION

While we can not predict the prices accurately every time, sometimes we get a nice approximation.

Maybe calculating signature up to different levels would give us nicer approximation. But we did not do this, because of computational difficulties.



Thank you for your attention!

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**

Please keep this slide for attribution