Learning with signatures: embedding and truncation order selection

DataSig Seminar Series

Adeline Fermanian

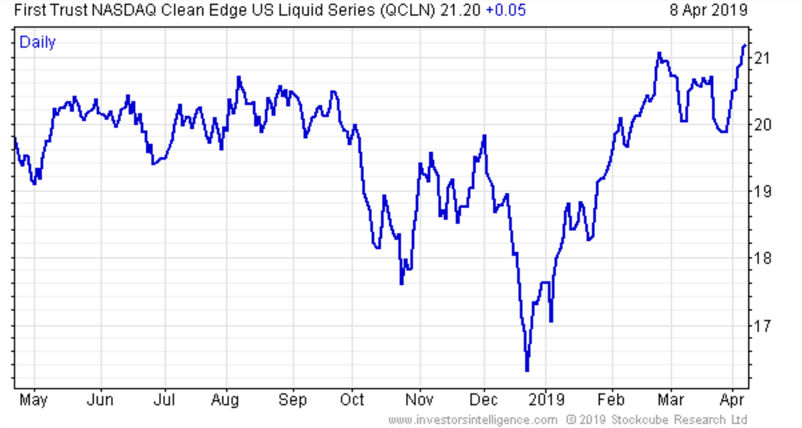
April 30th 2020

Joint work with

Benoît Cadre
University Rennes 2

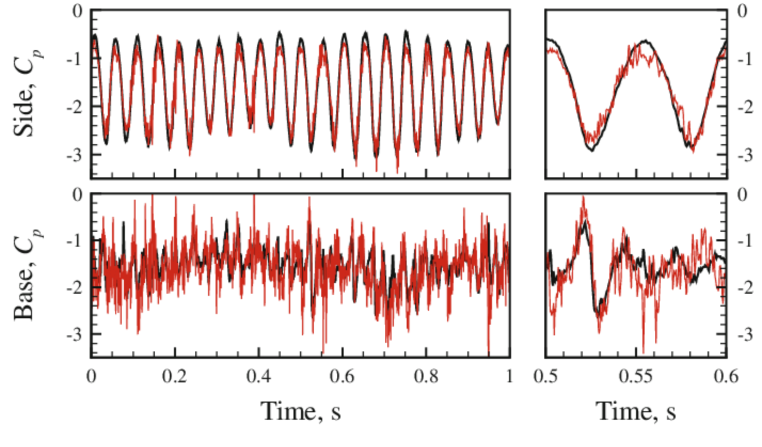
Gérard Biau
Sorbonne University

Learning from a data stream



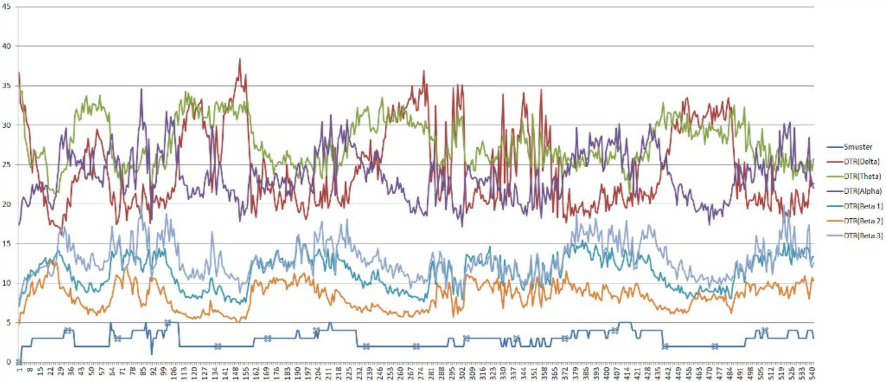
Time series prediction

Learning from a data stream



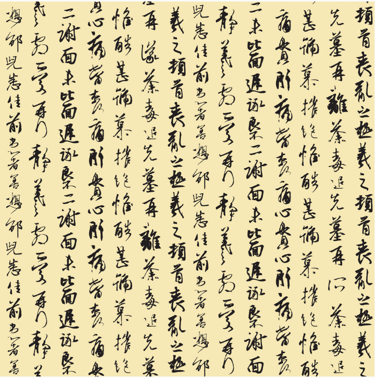
Stereo sound recognition

Learning from a data stream



Automated medical diagnosis from sensor data

Learning from a data stream

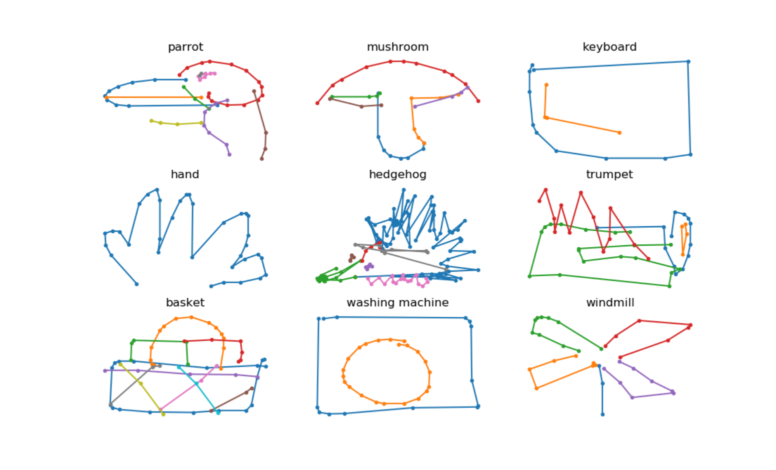


Recognition of characters or handwriting

Common feature

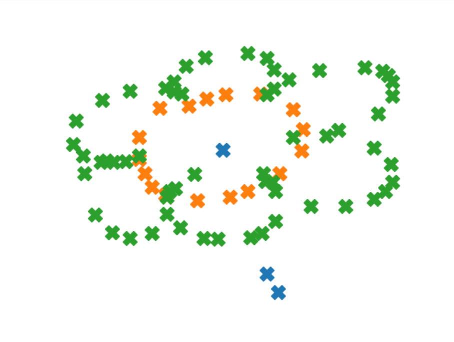
The predictor is a path $X : [a, b] \to \mathbb{R}^d$.

Google "Quick, Draw!" dataset



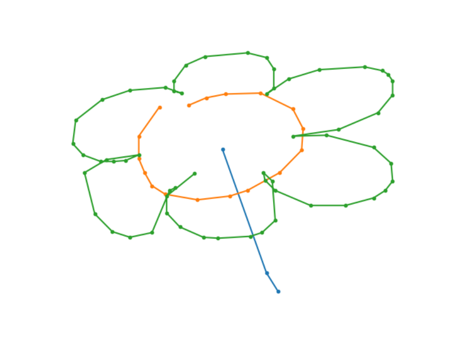
50 million drawings, 340 classes

Data representation



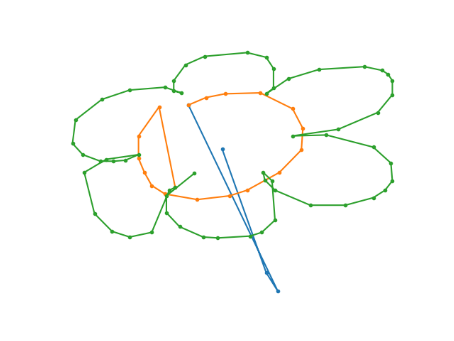
A sample from the class flower

Data representation



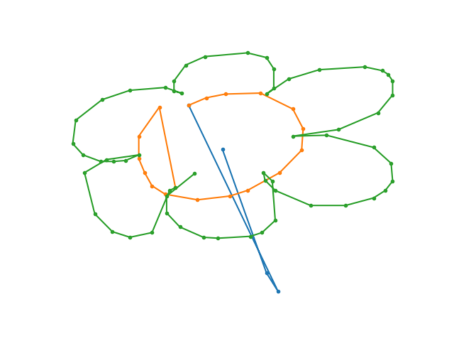
A sample from the class flower

Data representation

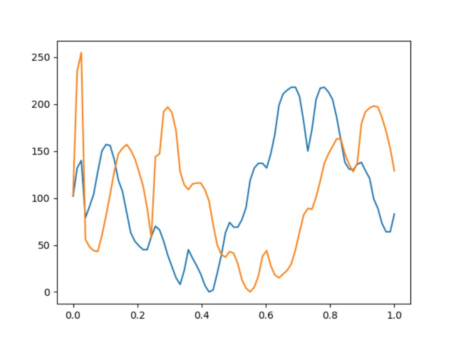


A sample from the class flower

Data representation

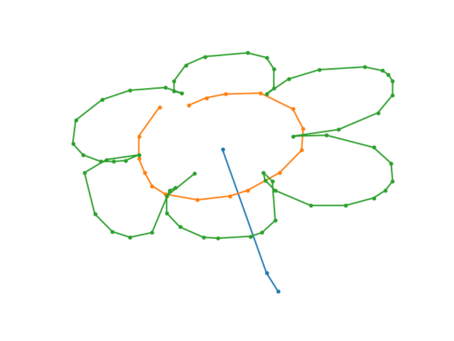


A sample from the class flower

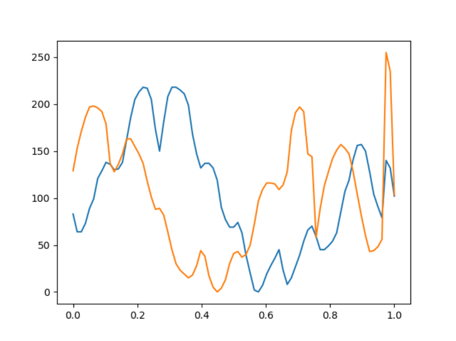


x and y coordinates

Data representation

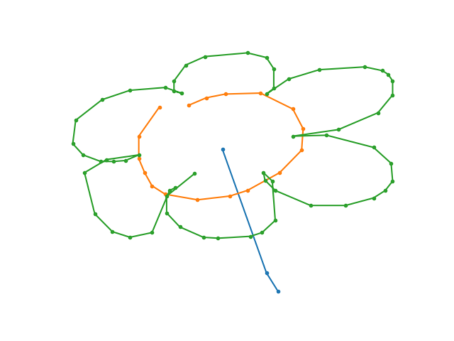


A sample from the class flower

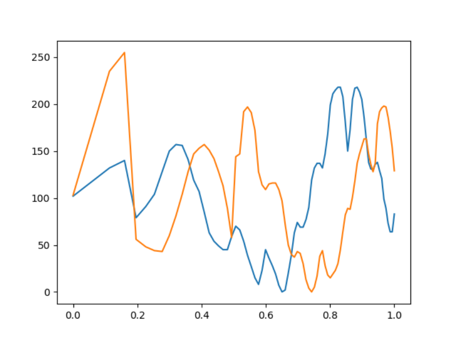


Time reversed

Data representation



A sample from the class flower



x and y at a different speed

The signature will overcome some of these problems.

The signature will overcome some of these problems.

- It is a transformation from a path to a sequence of coefficients.

The signature will overcome some of these problems.

- It is a transformation from a path to a sequence of coefficients.
- Independent of time parameterization.

The signature will overcome some of these problems.

- It is a transformation from a path to a sequence of coefficients.
- Independent of time parameterization.
- Encodes geometric properties of the path.

The signature will overcome some of these problems.

- It is a transformation from a path to a sequence of coefficients.
- Independent of time parameterization.
- Encodes geometric properties of the path.
- No loss of information.

Table of contents

Definition and basic properties

Mathematical setting

- A path $X : [0, 1] \to \mathbb{R}^d$. Notation: X_t.

Mathematical setting

- A path $X : [0, 1] \to \mathbb{R}^d$. Notation: X_t.
- Assumption: $\|X\|_{\text{1-var}} < \infty$.

Mathematical setting

- A path $X : [0,1] \to \mathbb{R}^d$. Notation: X_t.
- Assumption: $\|X\|_{\text{1-var}} < \infty$.
- $Y : [0,1] \to \mathbb{R}$ a continuous path.

Mathematical setting

- A path $X : [0,1] \to \mathbb{R}^d$. Notation: X_t.
- Assumption: $\|X\|_{\text{1-var}} < \infty$.
- $Y : [0,1] \to \mathbb{R}$ a continuous path.
- Riemann-Stieljes integral of Y against X is well-defined. Notation:

$$\int_0^1 Y_t dX_t.$$

Mathematical setting

- A path $X : [0,1] \to \mathbb{R}^d$. Notation: X_t.
- Assumption: $\|X\|_{\text{1-var}} < \infty$.
- $Y : [0,1] \to \mathbb{R}$ a continuous path.
- Riemann-Stieljes integral of Y against X is well-defined. Notation:

$$\int_0^1 Y_t dX_t.$$

Example :

- X_t continuously differentiable:

$$\int_0^1 Y_t dX_t = \int_0^1 Y_t \dot{X}_t dt$$

Mathematical setting

- A path $X : [0,1] \to \mathbb{R}^d$. Notation: X_t.
- Assumption: $\|X\|_{\text{1-var}} < \infty$.
- $Y : [0,1] \to \mathbb{R}$ a continuous path.
- Riemann-Stieljes integral of Y against X is well-defined. Notation:

$$\int_0^1 Y_t dX_t.$$

Example :

- $Y_t = 1$ for all $t \in [0,1]$:

$$\int_0^1 Y_t dX_t = \int_0^1 dX_t = X_1 - X_0.$$

Iterated integrals

- $X : [0,1] \to \mathbb{R}^d$, $X = (X^1, \dots, X^d)$.

Iterated integrals

- $X : [0,1] \to \mathbb{R}^d$, $X = (X^1, \ldots, X^d)$.
- For $i \in \{1, \ldots, d\}$,

$$S^i(X)_{[0,t]} = \int_{0<s<t} dX^i_s = X^i_t - X^i_0$$

Iterated integrals

- $X : [0,1] \to \mathbb{R}^d$, $X = (X^1, \ldots, X^d)$.
- For $i \in \{1, \ldots, d\}$,

$$S^i(X)_{[0,t]} = \int_{0<s<t} dX_s^i = X_t^i - X_0^i \quad \to \text{a path!}$$

Iterated integrals

- $X : [0,1] \to \mathbb{R}^d$, $X = (X^1, \ldots, X^d)$.
- For $i \in \{1, \ldots, d\}$,

$$S^i(X)_{[0,t]} = \int_{0<s<t} dX_s^i = X_t^i - X_0^i \quad \to \text{a path!}$$

- For $(i, j) \in \{1, \ldots, d\}^2$,

$$S^{i,j}(X)_{[0,t]} = \int_{0<s<t} S^i(X)_{[0,s]} dX_s^j = \int_{0<r<s<t} dX_r^i dX_s^j$$

Iterated integrals

- $X : [0,1] \to \mathbb{R}^d$, $X = (X^1, \ldots, X^d)$.
- For $i \in \{1, \ldots, d\}$,

$$S^i(X)_{[0,t]} = \int_{0<s<t} dX_s^i = X_t^i - X_0^i \quad \to \text{a path!}$$

- For $(i,j) \in \{1, \ldots, d\}^2$,

$$S^{i,j}(X)_{[0,t]} = \int_{0<s<t} S^i(X)_{[0,s]} dX_s^j = \int_{0<r<s<t} dX_r^i dX_s^j \quad \to \text{a path!}$$

Iterated integrals

- $X : [0,1] \to \mathbb{R}^d$, $X = (X^1, \ldots, X^d)$.
- For $i \in \{1, \ldots, d\}$,

$$S^i(X)_{[0,t]} = \int_{0<s<t} dX_s^i = X_t^i - X_0^i \quad \to \text{a path!}$$

- For $(i, j) \in \{1, \ldots, d\}^2$,

$$S^{i,j}(X)_{[0,t]} = \int_{0<s<t} S^i(X)_{[0,s]} dX_s^j = \int_{0<r<s<t} dX_r^i dX_s^j \quad \to \text{a path!}$$

- Recursively, for $(i_1, \ldots, i_k) \in \{1, \ldots, d\}^k$,

$$S^{(i_1,\ldots,i_k)}(X)_{[0,t]} = \int_{0<t_1<t_2<\cdots<t_k<t} dX_{t_1}^{i_1} \ldots dX_{t_k}^{i_k}.$$

Iterated integrals

- $X : [0,1] \to \mathbb{R}^d$, $X = (X^1, \ldots, X^d)$.
- For $i \in \{1, \ldots, d\}$,

$$S^i(X)_{[0,t]} = \int_{0<s<t} dX^i_s = X^i_t - X^i_0 \quad \to \text{a path!}$$

- For $(i,j) \in \{1, \ldots, d\}^2$,

$$S^{i,j}(X)_{[0,t]} = \int_{0<s<t} S^i(X)_{[0,s]} dX^j_s = \int_{0<r<s<t} dX^i_r dX^j_s \quad \to \text{a path!}$$

- Recursively, for $(i_1, \ldots, i_k) \in \{1, \ldots, d\}^k$,

$$S^{(i_1,\ldots,i_k)}(X)_{[0,t]} = \int_{0<t_1<t_2<\cdots<t_k<t} dX^{i_1}_{t_1} \ldots dX^{i_k}_{t_k}.$$

- $S^{(i_1,\ldots,i_k)}(X)_{[0,1]}$ is the k-fold iterated integral of X along $i_1, \ldots, i_k$.

Signature

Definition
The signature of X is the sequence of real numbers

$$S(X) = (1, S^{1}(X), \ldots, S^{d}(X), S^{(1,1)}(X), S^{(1,2)}(X), \ldots).$$

Signature

Definition
The signature of X is the sequence of real numbers

$$S(X) = (1, S^1(X), \ldots, S^d(X), S^{(1,1)}(X), S^{(1,2)}(X), \ldots).$$

- $d = 3 \rightarrow (1, 2, 3, 11, 12, 13, 21, 22, 23, 31, 32, 33, 111, 112, 113, \ldots)$

Signature

Definition
The signature of X is the sequence of real numbers

$$S(X) = (1, S^1(X), \ldots, S^d(X), S^{(1,1)}(X), S^{(1,2)}(X), \ldots).$$

- $d = 3 \rightarrow (1, 2, 3, 11, 12, 13, 21, 22, 23, 31, 32, 33, 111, 112, 113, \ldots)$
- Tensor notation:

$$\mathbf{X}^{\mathbf{k}} = \sum_{(i_1,\ldots,i_k)\subset\{1,\ldots,d\}^k} S^{(i_1,\ldots,i_k)}(X) e_{i_1} \otimes \cdots \otimes e_{i_k}.$$

Signature

Definition
The signature of X is the sequence of real numbers

$$S(X) = (1, S^1(X), \ldots, S^d(X), S^{(1,1)}(X), S^{(1,2)}(X), \ldots).$$

- $d = 3 \to (1, 2, 3, 11, 12, 13, 21, 22, 23, 31, 32, 33, 111, 112, 113, \ldots)$
- Tensor notation:

$$\mathbf{X}^{\mathbf{k}} = \sum_{(i_1,\ldots,i_k)\subset\{1,\ldots,d\}^k} S^{(i_1,\ldots,i_k)}(X) e_{i_1} \otimes \cdots \otimes e_{i_k}.$$

- Signature:

$$S(X) = (1, \mathbf{X}^{\mathbf{1}}, \mathbf{X}^{\mathbf{2}}, \ldots, \mathbf{X}^{\mathbf{k}}, \ldots) \in T(\mathbb{R}^d),$$

Signature

Definition
The signature of X is the sequence of real numbers

$$S(X) = (1, S^1(X), \dots, S^d(X), S^{(1,1)}(X), S^{(1,2)}(X), \dots).$$

- $d = 3 \to (1, 2, 3, 11, 12, 13, 21, 22, 23, 31, 32, 33, 111, 112, 113, \dots)$
- Tensor notation:

$$\mathbf{X}^{\mathbf{k}} = \sum_{(i_1,\dots,i_k)\subset\{1,\dots,d\}^k} S^{(i_1,\dots,i_k)}(X) e_{i_1} \otimes \cdots \otimes e_{i_k}.$$

- Signature:

$$S(X) = (1, \mathbf{X}^{\mathbf{1}}, \mathbf{X}^{\mathbf{2}}, \dots, \mathbf{X}^{\mathbf{k}}, \dots) \in T(\mathbb{R}^d),$$

where

$$T(\mathbb{R}^d) = 1 \oplus \mathbb{R}^d \oplus (\mathbb{R}^d)^{\otimes 2} \oplus \cdots \oplus (\mathbb{R}^d)^{\otimes k} \oplus \cdots$$

Example

For $X_t = (X_t^1, X_t^2)$,

$$\mathbf{X}^1 = \begin{pmatrix} \int_0^1 dX_t^1 & \int_0^1 dX_t^2 \end{pmatrix} = \begin{pmatrix} X_1^1 - X_0^1 & X_1^2 - X_0^2 \end{pmatrix}$$

Example

For $X_t = (X_t^1, X_t^2)$,

$$\mathbf{X}^1 = \begin{pmatrix} \int_0^1 dX_t^1 & \int_0^1 dX_t^2 \end{pmatrix} = \begin{pmatrix} X_1^1 - X_0^1 & X_1^2 - X_0^2 \end{pmatrix}$$

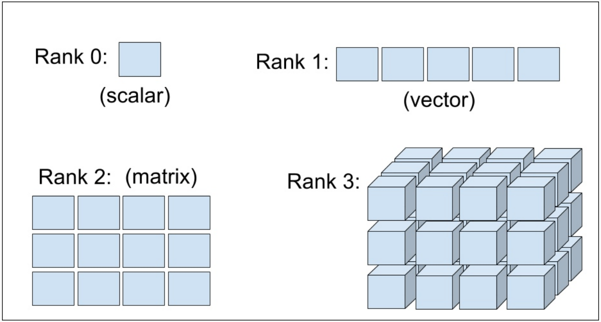
$$\mathbf{X}^2 = \begin{pmatrix} \int_0^1 \int_0^t dX_s^1 dX_t^1 & \int_0^1 \int_0^t dX_s^1 dX_t^2 \\ \int_0^1 \int_0^t dX_s^2 dX_t^1 & \int_0^1 \int_0^t dX_s^2 dX_t^2 \end{pmatrix}$$

Example

For $X_t = (X_t^1, X_t^2)$,

$$\mathbf{X}^1 = \begin{pmatrix} \int_0^1 dX_t^1 & \int_0^1 dX_t^2 \end{pmatrix} = \begin{pmatrix} X_1^1 - X_0^1 & X_1^2 - X_0^2 \end{pmatrix}$$

$$\mathbf{X}^2 = \begin{pmatrix} \int_0^1 \int_0^t dX_s^1 dX_t^1 & \int_0^1 \int_0^t dX_s^1 dX_t^2 \\ \int_0^1 \int_0^t dX_s^2 dX_t^1 & \int_0^1 \int_0^t dX_s^2 dX_t^2 \end{pmatrix}$$



Truncated signature

- Truncated signature at order m:

$$S^m(X) = (1, \mathbf{X^1}, \mathbf{X^2}, \ldots, \mathbf{X^m}).$$

Truncated signature

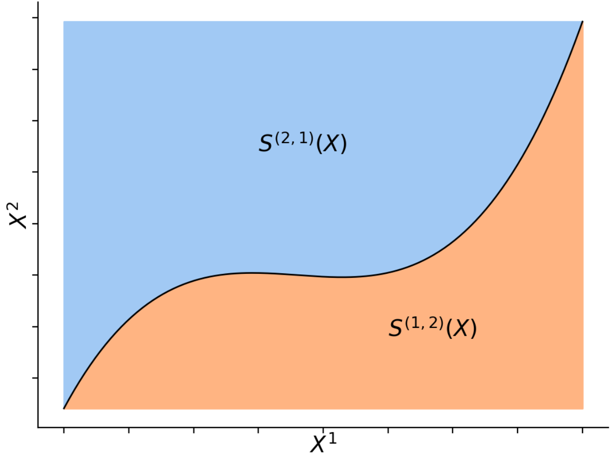
- Truncated signature at order m:

$$S^m(X) = (1, \mathbf{X^1}, \mathbf{X^2}, \ldots, \mathbf{X^m}).$$

- Dimension:

$$s_d(m) = \sum_{k=0}^{m} d^k = \frac{d^{m+1} - 1}{d - 1}.$$

Geometric interpretation



Important example

Linear path

- $X : [0,1] \to \mathbb{R}^d$ a linear path.

Important example

Linear path

- $X : [0,1] \to \mathbb{R}^d$ a linear path.
- $X_t = X_0 + X_1 t$.

Important example

Linear path

- $X : [0,1] \to \mathbb{R}^d$ a linear path.
- $X_t = X_0 + X_1 t$.
- For any $I = (i_1, \dots, i_k)$,

$$S^I(X) = \frac{1}{k!} \prod_{j=1}^{k} X_1^{i_j}.$$

Important example

Linear path

- $X : [0,1] \to \mathbb{R}^d$ a linear path.
- $X_t = X_0 + X_1 t$.
- For any $I = (i_1, \dots, i_k)$,

$$S^I(X) = \frac{1}{k!} \prod_{j=1}^{k} X_1^{i_j}.$$

▷ Very useful: in practice, we always deal with piecewise linear paths.

▷ Needed: concatenation operations.

Properties 1

Chen's identity

- $X : [a, b] \to \mathbb{R}^d$ and $Y : [b, c] \to \mathbb{R}^d$ paths.

Properties 1

Chen's identity

- $X : [a, b] \to \mathbb{R}^d$ and $Y : [b, c] \to \mathbb{R}^d$ paths.
- $X * Y : [a, c] \to \mathbb{R}^d$ the concatenation.

Properties 1

Chen's identity

- $X : [a, b] \to \mathbb{R}^d$ and $Y : [b, c] \to \mathbb{R}^d$ paths.
- $X * Y : [a, c] \to \mathbb{R}^d$ the concatenation.
- Then
$$S(X * Y) = S(X) \otimes S(Y).$$

Properties 1

Chen's identity

- $X : [a, b] \to \mathbb{R}^d$ and $Y : [b, c] \to \mathbb{R}^d$ paths.
- $X * Y : [a, c] \to \mathbb{R}^d$ the concatenation.
- Then
$$S(X * Y) = S(X) \otimes S(Y).$$

▷ We can compute the signature of piecewise linear paths!

▷ Data stream of p points and truncation at m: $O(pd^m)$ operations.

▷ Fast packages and libraries available in `C++` and `Python`.

Properties 2

Uniqueness
If X has at least one monotone coordinate, then $S(X)$ determines X uniquely.

Properties 2

Uniqueness
If X has at least one monotone coordinate, then $S(X)$ determines X uniquely.

- The signature characterizes paths.
- Trick: add a dummy monotonous component to X.
- Important concept of embedding.

Properties 3

Signature approximation

- D compact subset of paths from $[0,1]$ to $\mathbb{R}^d$ that are not tree-like equivalent.

Properties 3

Signature approximation

- D compact subset of paths from $[0,1]$ to $\mathbb{R}^d$ that are not tree-like equivalent.
- $f : D \to \mathbb{R}$ continuous.

Properties 3

Signature approximation

- D compact subset of paths from $[0,1]$ to $\mathbb{R}^d$ that are not tree-like equivalent.
- $f : D \to \mathbb{R}$ continuous.
- Then, for every $\varepsilon > 0$, there exists $w \in T(\mathbb{R}^d)$ such that, for any $X \in D$,

$$\big| f(X) - \langle w, S(X) \rangle \big| \leq \varepsilon.$$

Properties 3

Signature approximation

- D compact subset of paths from $[0,1]$ to $\mathbb{R}^d$ that are not tree-like equivalent.
- $f : D \to \mathbb{R}$ continuous.
- Then, for every $\varepsilon > 0$, there exists $w \in T(\mathbb{R}^d)$ such that, for any $X \in D$,

$$\big| f(X) - \langle w, S(X) \rangle \big| \leq \varepsilon.$$

▷ Signature and linear model are happy together!

▷ This raises many interesting statistical issues.

Learning with signatures

Supervised learning

- Goal: understand the relationship between $X \in \mathscr{X}$ and $Y \in \mathscr{Y}$.

Supervised learning

- Goal: understand the relationship between $X \in \mathscr{X}$ and $Y \in \mathscr{Y}$.
- Regression: $\mathscr{Y} = \mathbb{R}$ Classification: $\mathscr{Y} = \{1, \ldots, q\}$.

Supervised learning

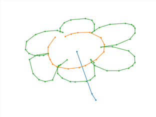
- Goal: understand the relationship between $X \in \mathscr{X}$ and $Y \in \mathscr{Y}$.
- Regression: $\mathscr{Y} = \mathbb{R}$ Classification: $\mathscr{Y} = \{1, \dots, q\}$.
- Data: $(X_1, Y_1), \dots, (X_n, Y_n) \in \mathscr{X} \times \mathscr{Y}$, i.i.d. $\sim (X, Y)$.

Supervised learning

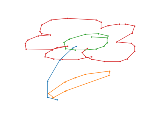
- Goal: understand the relationship between $X \in \mathscr{X}$ and $Y \in \mathscr{Y}$.
- Regression: $\mathscr{Y} = \mathbb{R}$ Classification: $\mathscr{Y} = \{1, \ldots, q\}$.
- Data: $(X_1, Y_1), \ldots, (X_n, Y_n) \in \mathscr{X} \times \mathscr{Y}$, i.i.d. $\sim (X, Y)$.
- Prediction function: $f_\theta(X) \approx Y$, $\theta \in \mathbb{R}^p$.

Supervised learning

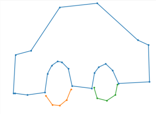
- Goal: understand the relationship between $X \in \mathscr{X}$ and $Y \in \mathscr{Y}$.
- Regression: $\mathscr{Y} = \mathbb{R}$ Classification: $\mathscr{Y} = \{1, \dots, q\}$.
- Data: $(X_1, Y_1), \dots, (X_n, Y_n) \in \mathscr{X} \times \mathscr{Y}$, i.i.d. $\sim (X, Y)$.
- Prediction function: $f_\theta(X) \approx Y$, $\theta \in \mathbb{R}^p$.



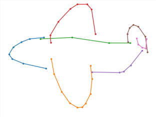
$y_1 = 1$



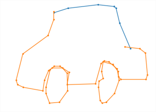
$y_2 = 1$



$y_3 = 2$



$y_4 = 3$



$y_5 = 2$

Supervised learning

- Loss function $\ell : \mathscr{Y} \times \mathscr{Y} \to \mathbb{R}^+$.

Supervised learning

- Loss function $\ell : \mathscr{Y} \times \mathscr{Y} \to \mathbb{R}^+$.
- Empirical risk minimization: choose

$$\hat{\theta} \in \underset{\theta \in \mathbb{R}^p}{\mathrm{argmin}} \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, f_\theta(X_i)).$$

Supervised learning

- Loss function $\ell : \mathscr{Y} \times \mathscr{Y} \to \mathbb{R}^+$.
- Empirical risk minimization: choose

$$\hat{\theta} \in \underset{\theta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, f_\theta(X_i)).$$

- **Least squares regression**: $\mathscr{Y} = \mathbb{R}$ and $\ell(y, f_\theta(x)) = (y - f_\theta(x))^2$.

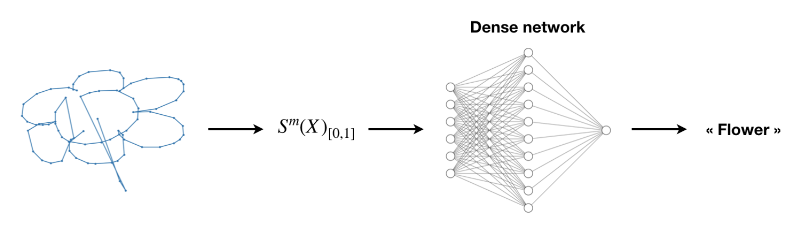
Supervised learning

- Loss function $\ell : \mathscr{Y} \times \mathscr{Y} \to \mathbb{R}^+$.
- Empirical risk minimization: choose

$$\hat{\theta} \in \underset{\theta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, f_\theta(X_i)).$$

- **Least squares regression**: $\mathscr{Y} = \mathbb{R}$ and $\ell(y, f_\theta(x)) = (y - f_\theta(x))^2$.
- **Binary classification**: $\mathscr{Y} = \{0, 1\}$ and $\ell(y, f_\theta(x)) = \mathbb{1}_{[f_\theta(x) \neq y]}$.

Signature + machine learning

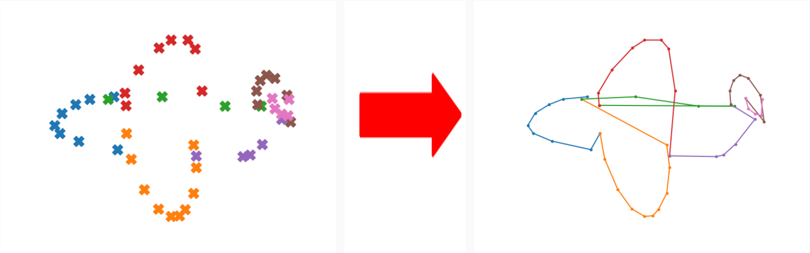


Questions

- How should we choose the order of truncation?

Questions

- How should we choose the order of truncation?
- Which embedding should we use?



Truncation order

Regression model on the signature

- $X : [0, 1] \to \mathbb{R}^d$ random path, $Y \in \mathbb{R}$ random variable.

Regression model on the signature

- $X : [0,1] \to \mathbb{R}^d$ random path, $Y \in \mathbb{R}$ random variable.
- Assumption: there exists $m^* \in \mathbb{N}$, $\beta^* \in \mathbb{R}^{s_d(m^*)}$ such that

$$\mathbb{E}[Y|X] = \langle \beta^*, S^{m^*}(X) \rangle, \quad \text{and} \quad \mathrm{Var}(Y|X) \leq \sigma^2 < \infty.$$

Regression model on the signature

- $X : [0,1] \to \mathbb{R}^d$ random path, $Y \in \mathbb{R}$ random variable.
- Assumption: there exists $m^* \in \mathbb{N}$, $\beta^* \in \mathbb{R}^{s_d(m^*)}$ such that

$$\mathbb{E}[Y|X] = \langle \beta^*, S^{m^*}(X) \rangle, \quad \text{and} \quad \mathrm{Var}(Y|X) \leq \sigma^2 < \infty.$$

- Goal: estimate m^* and β^*.

Estimation of m^*

- Data: $(X_1, Y_1), \ldots, (X_n, Y_n)$ i.i.d.

Estimation of m^*

- Data: $(X_1, Y_1), \ldots, (X_n, Y_n)$ i.i.d.
- For any $m \in \mathbb{N}$, $\alpha > 0$,

$$B_{m,\alpha} = \left\{ \beta \in \mathbb{R}^{s_d(m)} : \|\beta\|_2 \leq \alpha \right\}.$$

Estimation of m^*

- Data: $(X_1, Y_1), \dots, (X_n, Y_n)$ i.i.d.
- For any $m \in \mathbb{N}$, $\alpha > 0$,

$$B_{m,\alpha} = \Big\{\beta \in \mathbb{R}^{s_d(m)} : \|\beta\|_2 \leq \alpha\Big\}.$$

- For any $m \in \mathbb{N}$, $\beta \in B_{m,\alpha}$,

$$\mathcal{R}_{m,n}(\beta) = \frac{1}{n}\sum_{i=1}^{n}\big(Y_i - \langle \beta, S^m(X_i)\rangle\big)^2.$$

Estimation of m^*

- Data: $(X_1, Y_1), \ldots, (X_n, Y_n)$ i.i.d.
- For any $m \in \mathbb{N}$, $\alpha > 0$,

$$B_{m,\alpha} = \left\{ \beta \in \mathbb{R}^{s_d(m)} : \|\beta\|_2 \leq \alpha \right\}.$$

- For any $m \in \mathbb{N}$, $\beta \in B_{m,\alpha}$,

$$\mathcal{R}_{m,n}(\beta) = \frac{1}{n} \sum_{i=1}^{n} \big(Y_i - \langle \beta, S^m(X_i) \rangle \big)^2.$$

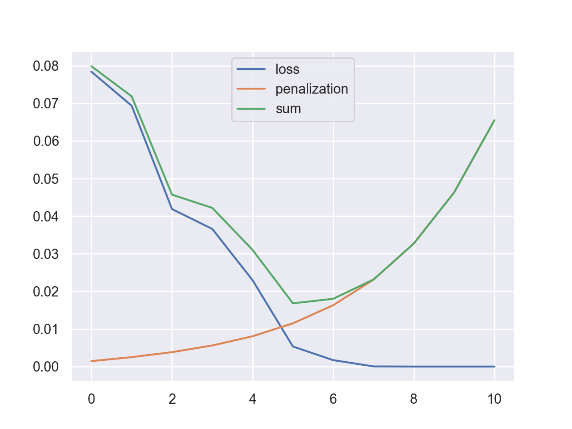
- For any $m \in \mathbb{N}$,

$$\widehat{L}_n(m) = \inf_{\beta \in B_{m,\alpha}} \mathcal{R}_{m,n}(\beta).$$

Estimation of m^*

Estimator:

$$\widehat{m} = \min\Big(\underset{m}{\mathrm{argmin}}\big(\widehat{L}_n(m) + \mathrm{pen}_n(m)\big)\Big).$$



Result

Additional assumptions:

(H_α) $\beta^* \in B_{m^*,\alpha}$.

(H_K) There exists $K_Y > 0$ and $K_X > 0$ such that almost surely

$$|Y| \leq K_Y \quad \text{and} \quad \|X\|_{\text{1-var}} \leq K_X.$$

Result

Theorem

Let $K_{\text{pen}} > 0$, $0 < \rho < \frac{1}{2}$, and

$$\text{pen}_n(m) = K_{\text{pen}} n^{-\rho} \sqrt{s_d(m)}.$$

Under the assumptions (H_α) and (H_K), for any $n \geq n_0$,

$$\mathbb{P}\left(\widehat{m} \neq m^*\right) \leq C_1 \exp\left(-C_2 n^{1-2\rho}\right),$$

where n_0, C_1 and C_2 are explicit constants.

Result

Theorem

Let $K_{\text{pen}} > 0$, $0 < \rho < \frac{1}{2}$, and

$$\text{pen}_n(m) = K_{\text{pen}} n^{-\rho} \sqrt{s_d(m)}.$$

Under the assumptions (H_α) and (H_K), for any $n \geq n_0$,

$$\mathbb{P}\left(\widehat{m} \neq m^*\right) \leq C_1 \exp\left(-C_2 n^{1-2\rho}\right),$$

where n_0, C_1 and C_2 are explicit constants.

Corollary
$\widehat{m}$ converges almost surely towards m^*.

Result

We can then estimate β^* by

$$\widehat{\beta} = \operatorname*{argmin}_{\beta \in B_{\widehat{m},\alpha}} \mathcal{R}_{\widehat{m},n}(\beta),$$

Result

We can then estimate β^* by

$$\widehat{\beta} = \operatorname*{argmin}_{\beta \in B_{\widehat{m},\alpha}} \mathcal{R}_{\widehat{m},n}(\beta),$$

and show that

$$\mathbb{E}\Big(\langle \widehat{\beta}, S^{\widehat{m}}(X)\rangle - \langle \beta^*, S^{m^*}(X)\rangle\Big)^2 = O\Big(\frac{1}{\sqrt{n}}\Big).$$

Path embeddings

Embedding
A way of mapping discrete sequential data into a continuous path.

Kaggle prediction competition

Featured Prediction Competition

Quick, Draw! Doodle Recognition Challenge

How accurately can you identify a doodle?

$25,000
Prize Money

Google AI · 1,316 teams · 4 months ago

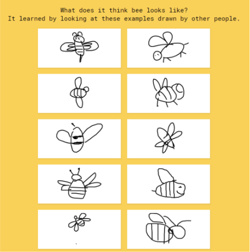
Overview | Data | Kernels | Discussion | Leaderboard | Rules | Team | My Submissions | Late Submission

Overview

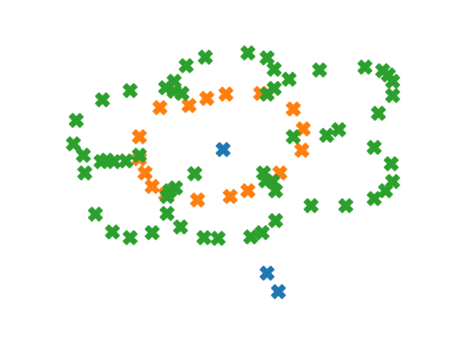
- Description
- Evaluation
- Prizes
- Timeline

"Quick, Draw!" was released as an experimental game to educate the public in a playful way about how AI works. The game prompts users to draw an image depicting a certain category, such as "banana," "table," etc. The game generated more than 1B drawings, of which a subset was publicly released as the basis for this competition's training set. That subset contains 50M drawings encompassing 340 label categories.

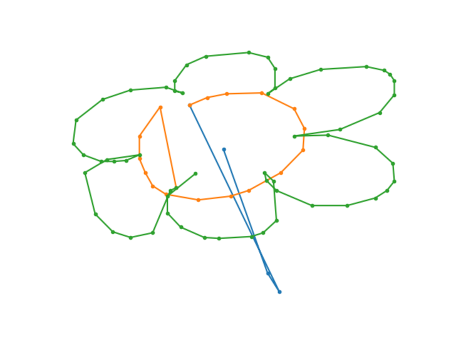
Sounds fun, right? Here's the challenge: since the training data comes from the game itself, drawings can be incomplete or may not match the label. You'll need to build a recognizer that can effectively learn from this noisy data and perform well on a manually-labeled test set from a different distribution.



Different embeddings

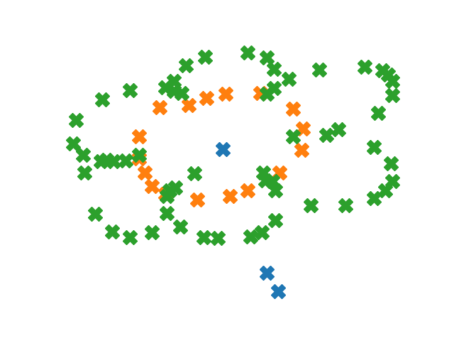


Original data

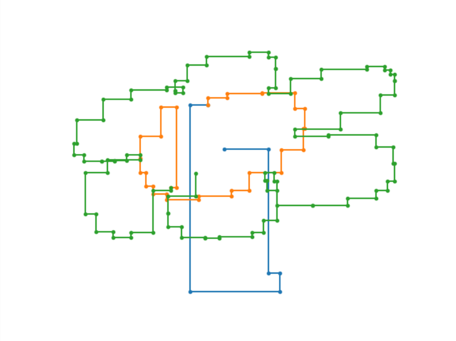


Linear path

Different embeddings

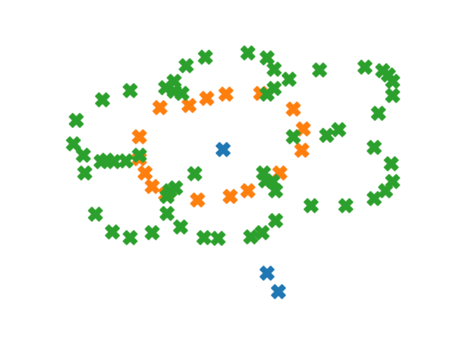


Original data

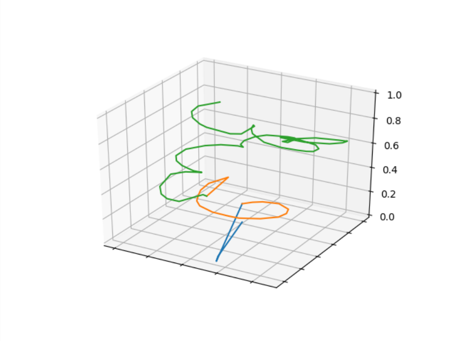


Rectilinear path

Different embeddings

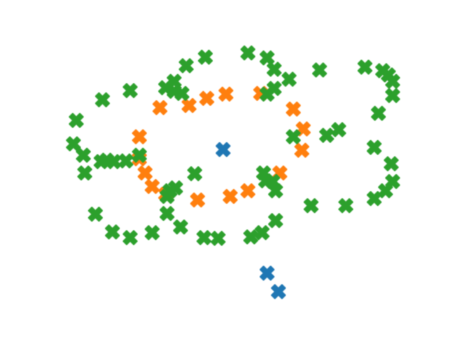


Original data

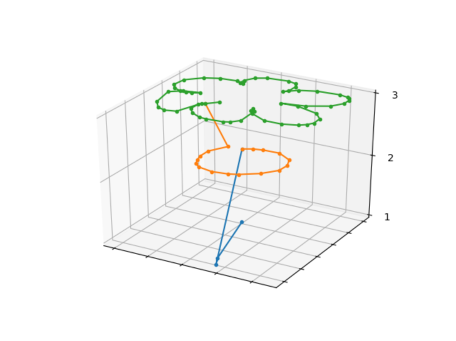


Time path

Different embeddings

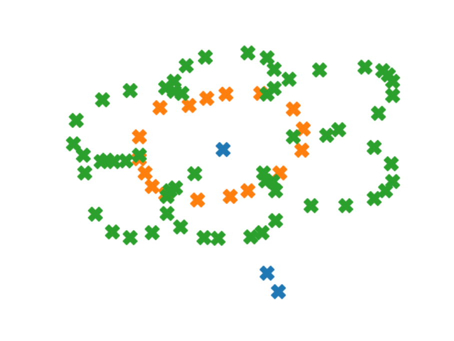


Original data

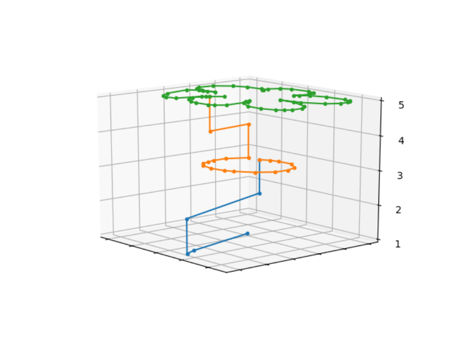


Stroke path, version 1

Different embeddings

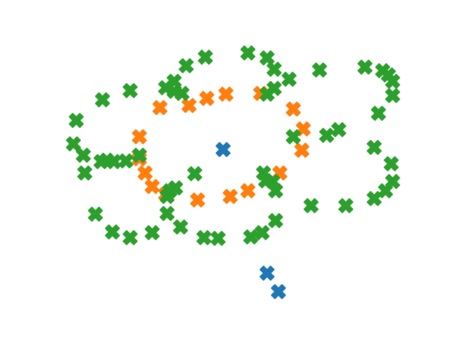


Original data

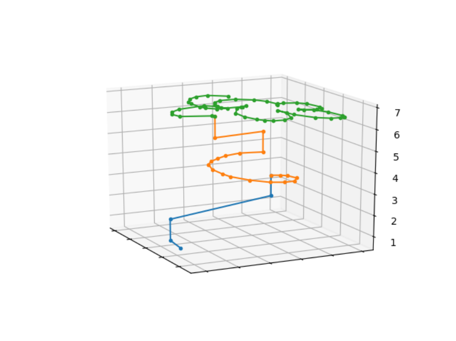


Stroke path, version 2

Different embeddings

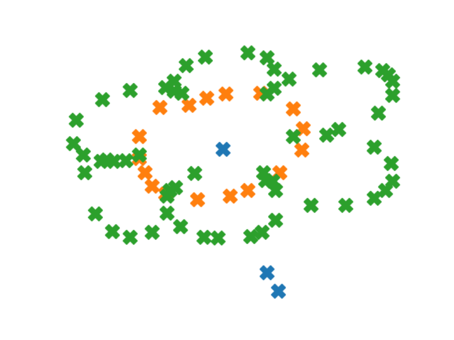


Original data



Stroke path, version 3

Different embeddings



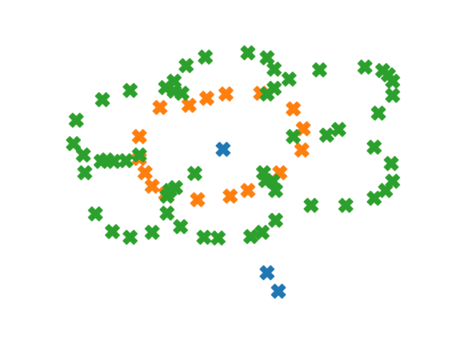
Original data

$t \to (X_t^1, X_t^2, t, X_t^3, X_t^4)$, where

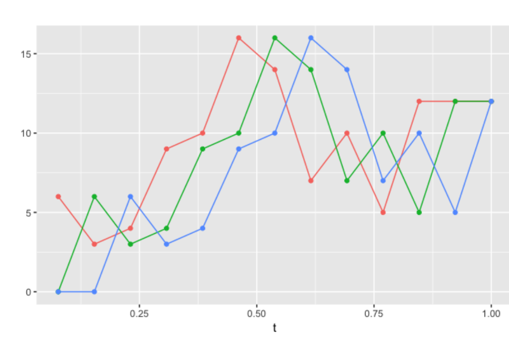
$$X_t^3 = \begin{cases} 0 & \text{if } t < t_1 \\ X_{t-t_1}^1 & \text{otherwise} \end{cases}$$

$$X_t^4 = \begin{cases} 0 & \text{if } t < t_1 \\ X_{t-t_1}^2 & \text{otherwise} \end{cases}$$

Different embeddings

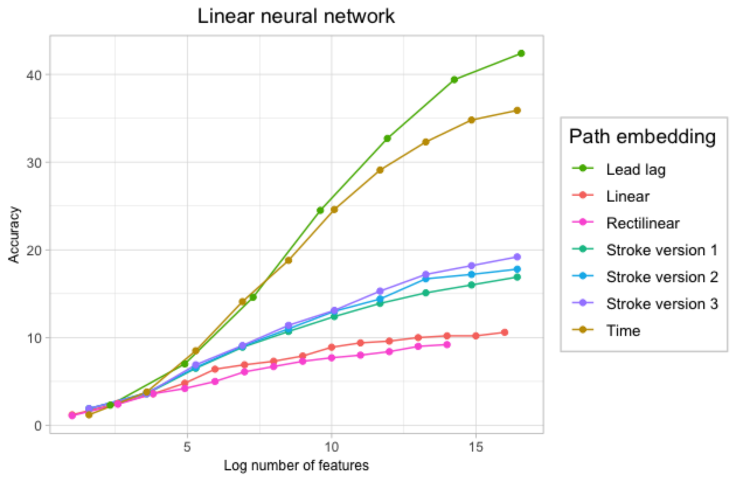


Original data



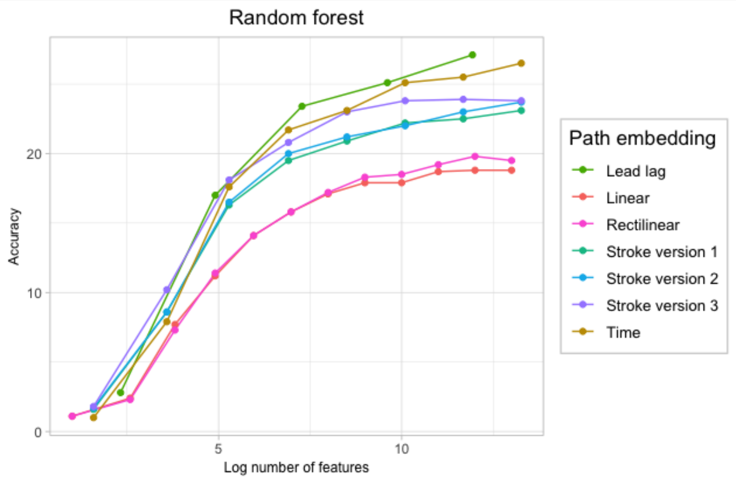
Lead-lag path

Quick, Draw! dataset results



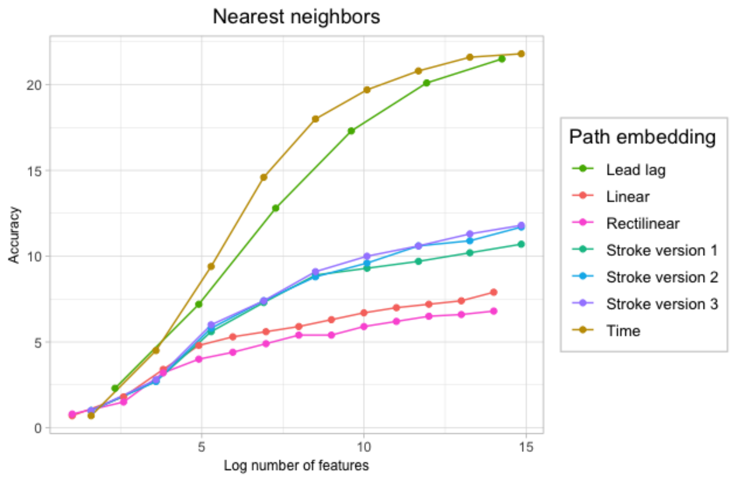
Prediction accuracy with a linear NN.

Quick, Draw! dataset results



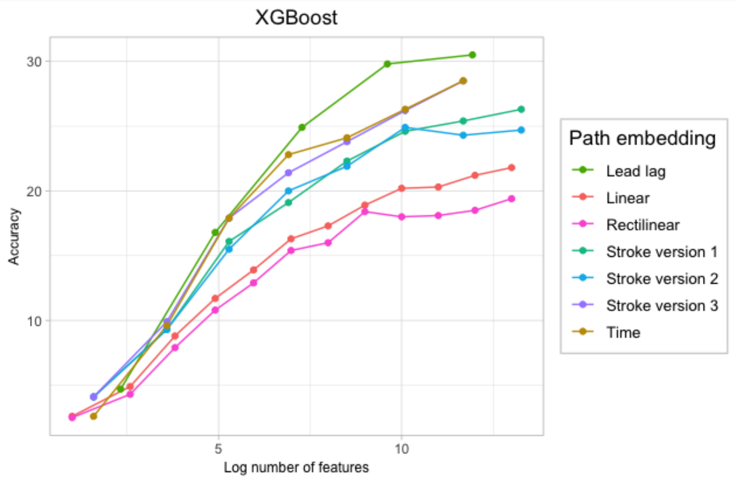
Prediction accuracy with a random forest.

Quick, Draw! dataset results



Prediction accuracy with 5 nearest neighbors

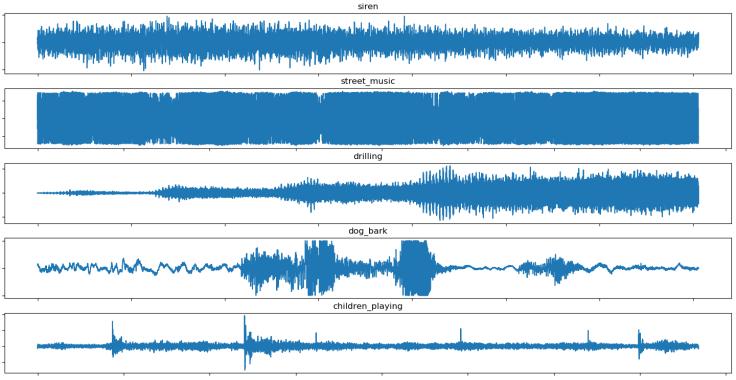
Quick, Draw! dataset results



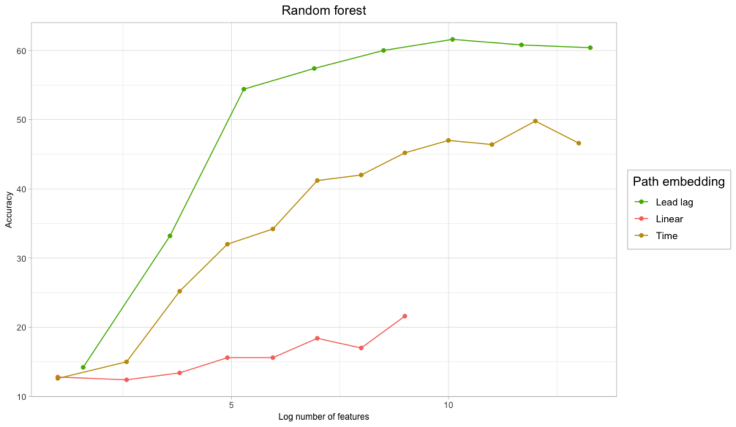
Prediction accuracy with XGBoost

Urban Sound dataset

10 different sounds: car horn, street music, dork barking...
5435 noisy 1-dimensional times series of average size 171 135



Urban Sound dataset results



Prediction accuracy with a random forest.

Motion Sense dataset

Smartphone sensory data recorded by accelerometer and gyroscope sensors

Goal: detect 6 activities (walking upstairs, jogging, sitting...)

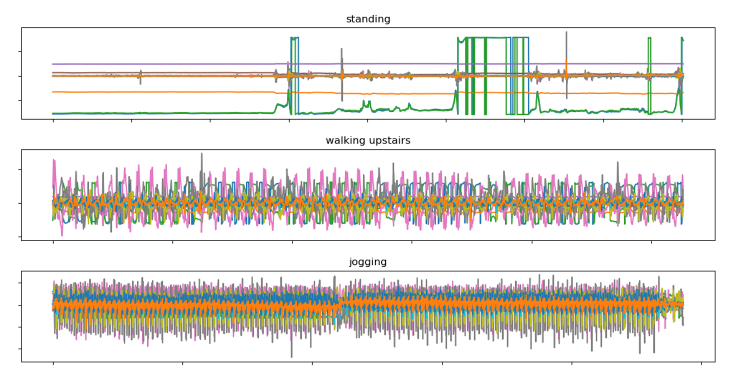
74 800 12-dimensional times series of average size 3934

Motion Sense dataset

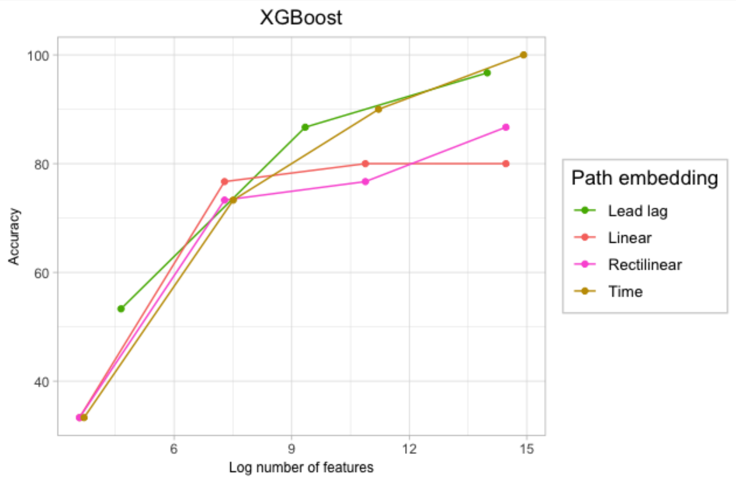
Smartphone sensory data recorded by accelerometer and gyroscope sensors

Goal: detect 6 activities (walking upstairs, jogging, sitting...)

74 800 12-dimensional times series of average size 3934



Motion Sense dataset results



Prediction accuracy with XGBoost.

Take-home message

- Striking fact: some embeddings seem consistently better.

Take-home message

- Striking fact: some embeddings seem consistently better.
- Good performance of the lead lag path.

Take-home message

- Striking fact: some embeddings seem consistently better.
- Good performance of the lead lag path.
- This is due to intrinsic properties of the signature and the embedding, not to domain-specific properties.

Take-home message

- Striking fact: some embeddings seem consistently better.
- Good performance of the lead lag path.
- This is due to intrinsic properties of the signature and the embedding, not to domain-specific properties.
- It is particularly remarkable as the dimension of the input stream is different from one dataset to another.

Take-home message

- Striking fact: some embeddings seem consistently better.
- Good performance of the lead lag path.
- This is due to intrinsic properties of the signature and the embedding, not to domain-specific properties.
- It is particularly remarkable as the dimension of the input stream is different from one dataset to another.
- Conclusion: the lead lag embedding seems to be the best choice, regardless of the data and algorithm used.

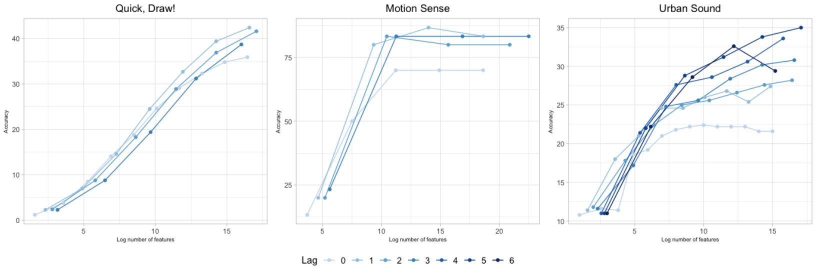
Take-home message

- Striking fact: some embeddings seem consistently better.
- Good performance of the lead lag path.
- This is due to intrinsic properties of the signature and the embedding, not to domain-specific properties.
- It is particularly remarkable as the dimension of the input stream is different from one dataset to another.
- Conclusion: the lead lag embedding seems to be the best choice, regardless of the data and algorithm used.
- Computationally cheap and drastically improves prediction accuracy.

Performance of signatures

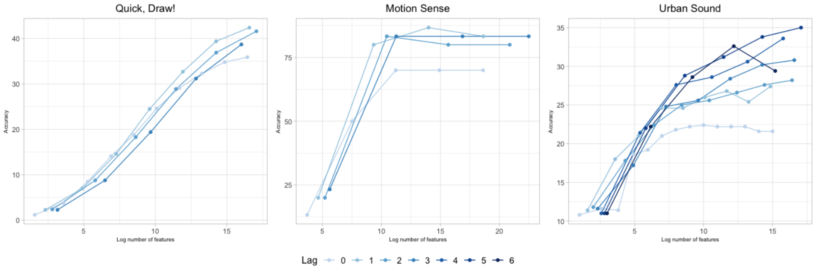
Our plan

- For each dataset: lead lag + lag selection.



Our plan

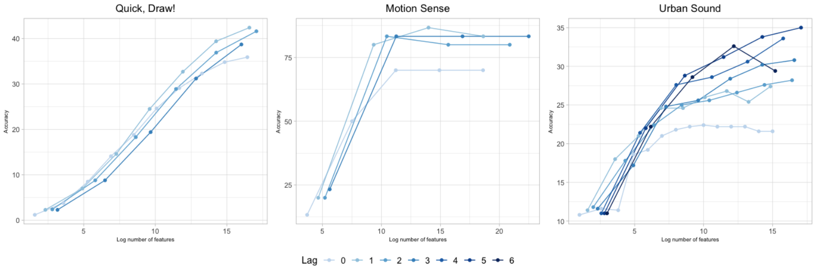
- For each dataset: lead lag + lag selection.



- Quick, Draw! and Motion Sense: 1.

Our plan

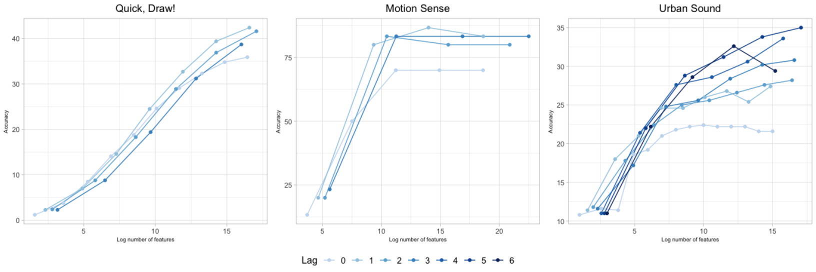
- For each dataset: lead lag + lag selection.



- Quick, Draw! and Motion Sense: 1. Urban Sound: 5.

Our plan

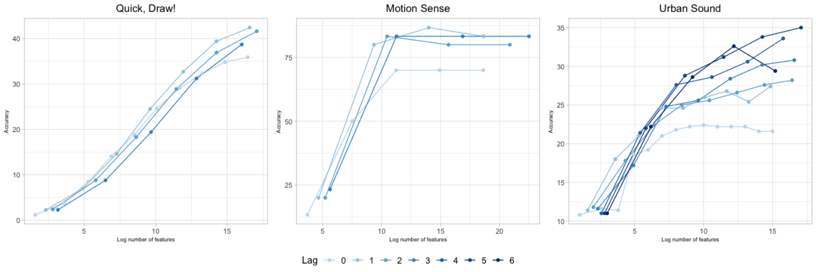
- For each dataset: lead lag + lag selection.



- Quick, Draw! and Motion Sense: 1. Urban Sound: 5.
- Quick, Draw!: dense NN with three hidden layers and ReLU activation (around 12 million samples for training and 80 000 for validation and test).

Our plan

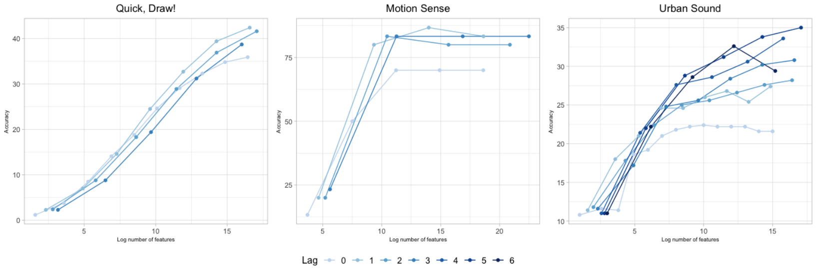
- For each dataset: lead lag + lag selection.



- Quick, Draw! and Motion Sense: 1. Urban Sound: 5.
- Quick, Draw!: dense NN with three hidden layers and ReLU activation (around 12 million samples for training and 80 000 for validation and test).
- Urban Sound: Random Forests.

Our plan

- For each dataset: lead lag + lag selection.



- Quick, Draw! and Motion Sense: 1. Urban Sound: 5.
- Quick, Draw!: dense NN with three hidden layers and ReLU activation (around 12 million samples for training and 80 000 for validation and test).
- Urban Sound: Random Forests.
- Motion Sense: XGBoost.

Performance of signature learning

- Quick, Draw!

Performance of signature learning

- Quick, Draw!
  - ▷ State of the art: deep CNN trained with several million of samples.

Performance of signature learning

- Quick, Draw!
  - ▷ State of the art: deep CNN trained with several million of samples.
  - ▷ Kaggle mean average precision at 3 = 95%.

Performance of signature learning

- Quick, Draw!
  - State of the art: deep CNN trained with several million of samples.
  - Kaggle mean average precision at $3 = 95\%$.
  - Our result: small NN + signature features at order $6 = 54\%$.

Performance of signature learning

- Quick, Draw!
  - ▷ State of the art: deep CNN trained with several million of samples.
  - ▷ Kaggle mean average precision at 3 $= 95\%$.
  - ▷ Our result: small NN + signature features at order 6 $= 54\%$.
- Urban Sound

Performance of signature learning

- Quick, Draw!
  - State of the art: deep CNN trained with several million of samples.
  - Kaggle mean average precision at $3 = 95\%$.
  - Our result: small NN + signature features at order $6 = 54\%$.
- Urban Sound
  - State of the art: feature extraction with mixture of experts models.

Performance of signature learning

- Quick, Draw!
  - State of the art: deep CNN trained with several million of samples.
  - Kaggle mean average precision at 3 $= 95\%$.
  - Our result: small NN + signature features at order 6 $= 54\%$.
- Urban Sound
  - State of the art: feature extraction with mixture of experts models.
  - Accuracy $= 77.36\%$.

Performance of signature learning

- Quick, Draw!
  - State of the art: deep CNN trained with several million of samples.
  - Kaggle mean average precision at $3 = 95\%$.
  - Our result: small NN + signature features at order $6 = 54\%$.
- Urban Sound
  - State of the art: feature extraction with mixture of experts models.
  - Accuracy $= 77.36\%$.
  - Our result: Random Forests + signature features at order $5 = 70\%$.

Performance of signature learning

- Quick, Draw!
  - State of the art: deep CNN trained with several million of samples.
  - Kaggle mean average precision at 3 = 95%.
  - Our result: small NN + signature features at order 6 = 54%.
- Urban Sound
  - State of the art: feature extraction with mixture of experts models.
  - Accuracy = 77.36%.
  - Our result: Random Forests + signature features at order 5 = 70%.
- Motion Sense

Performance of signature learning

- Quick, Draw!
  - ▷ State of the art: deep CNN trained with several million of samples.
  - ▷ Kaggle mean average precision at $3 = 95\%$.
  - ▷ Our result: small NN + signature features at order $6 = 54\%$.
- Urban Sound
  - ▷ State of the art: feature extraction with mixture of experts models.
  - ▷ Accuracy $= 77.36\%$.
  - ▷ Our result: Random Forests + signature features at order $5 = 70\%$.
- Motion Sense
  - ▷ State of the art: deep NN + autoencoders + multi-objective loss.

Performance of signature learning

- Quick, Draw!
  - State of the art: deep CNN trained with several million of samples.
  - Kaggle mean average precision at 3 = 95%.
  - Our result: small NN + signature features at order 6 = 54%.
- Urban Sound
  - State of the art: feature extraction with mixture of experts models.
  - Accuracy = 77.36%.
  - Our result: Random Forests + signature features at order 5 = 70%.
- Motion Sense
  - State of the art: deep NN + autoencoders + multi-objective loss.
  - F1 score = 92.91.

Performance of signature learning

- Quick, Draw!
  - State of the art: deep CNN trained with several million of samples.
  - Kaggle mean average precision at 3 = 95%.
  - Our result: small NN + signature features at order 6 = 54%.
- Urban Sound
  - State of the art: feature extraction with mixture of experts models.
  - Accuracy = 77.36%.
  - Our result: Random Forests + signature features at order 5 = 70%.
- Motion Sense
  - State of the art: deep NN + autoencoders + multi-objective loss.
  - F1 score = 92.91.
  - Our result: XGBoost + signature features at order 3 = 93.5.

Conclusion

- Our algorithms have not been tuned to a specific application.

Conclusion

- Our algorithms have not been tuned to a specific application.
- The combination "signature + generic algorithm" $\approx$ state-of-the-art.

Conclusion

- Our algorithms have not been tuned to a specific application.
- The combination "signature + generic algorithm" $\approx$ state-of-the-art.
- Few computing resources and no domain-specific knowledge.

Conclusion

- Our algorithms have not been tuned to a specific application.
- The combination "signature + generic algorithm" $\approx$ state-of-the-art.
- Few computing resources and no domain-specific knowledge.
- A lot of open questions

Thank you!