Bayesian Learning from Sequential Data using Gaussian Processes with Signature Covariances

Csaba Toth
Joint work with Harald Oberhauser
Mathematical Institute, University of Oxford

International Conference on Machine Learning, July 2020

University of Oxford
Mathematical Institute

Oxford Mathematics

Overview

Overview

Purpose of this work

1. Define a Gaussian process (GP) [6] over sequences/time series

Overview

Purpose of this work

1. Define a Gaussian process (GP) [6] over sequences/time series

   - To model of functions of sequences $\{\mathrm{Seq}(\mathbb{R}^d) \to \mathbb{R}\}$

$$(f_{\mathbf{x}})_{\mathbf{x} \in \mathrm{Seq}(\mathbb{R}^d)} \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot))$$

Overview

Purpose of this work

1. Define a Gaussian process (GP) [6] over sequences/time series

   - To model of functions of sequences $\{\mathrm{Seq}(\mathbb{R}^d) \to \mathbb{R}\}$

$$(f_{\mathbf{x}})_{\mathbf{x} \in \mathrm{Seq}(\mathbb{R}^d)} \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot))$$

   - Find a suitable covariance kernel

$$\mathrm{k} : \mathrm{Seq}(\mathbb{R}^d) \times \mathrm{Seq}(\mathbb{R}^d) \to \mathbb{R}$$

Overview

Purpose of this work

1. Define a Gaussian process (GP) [6] over sequences/time series

- To model of functions of sequences $\{\mathrm{Seq}(\mathbb{R}^d) \to \mathbb{R}\}$

$$(f_{\mathbf{x}})_{\mathbf{x} \in \mathrm{Seq}(\mathbb{R}^d)} \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot))$$

- Find a suitable covariance kernel

$$\mathrm{k} : \mathrm{Seq}(\mathbb{R}^d) \times \mathrm{Seq}(\mathbb{R}^d) \to \mathbb{R}$$

- $\mathrm{Seq}(\mathbb{R}^d) := \{(\mathbf{x}_{t_1}, \ldots, \mathbf{x}_{t_L}) \mid (t_i, \mathbf{x}_{t_i}) \in \mathbb{R}_+ \times \mathbb{R}^d, L \in \mathbb{N}\}$

Overview

Purpose of this work

1. Define a Gaussian process (GP) [6] over sequences/time series
   - To model of functions of sequences $\{\mathrm{Seq}(\mathbb{R}^d) \to \mathbb{R}\}$
     $$(f_{\mathbf{x}})_{\mathbf{x} \in \mathrm{Seq}(\mathbb{R}^d)} \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot))$$
   - Find a suitable covariance kernel
     $$\mathrm{k} : \mathrm{Seq}(\mathbb{R}^d) \times \mathrm{Seq}(\mathbb{R}^d) \to \mathbb{R}$$
   - $\mathrm{Seq}(\mathbb{R}^d) := \{(\mathbf{x}_{t_1}, \ldots, \mathbf{x}_{t_L}) \mid (t_i, \mathbf{x}_{t_i}) \in \mathbb{R}_+ \times \mathbb{R}^d, L \in \mathbb{N}\}$
2. Develop an efficient inference framework

Overview

Purpose of this work

1. Define a Gaussian process (GP) [6] over sequences/time series
   - To model of functions of sequences $\{\mathrm{Seq}(\mathbb{R}^d) \to \mathbb{R}\}$
     $$(f_{\mathbf{x}})_{\mathbf{x} \in \mathrm{Seq}(\mathbb{R}^d)} \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot))$$
   - Find a suitable covariance kernel
     $$\mathrm{k} : \mathrm{Seq}(\mathbb{R}^d) \times \mathrm{Seq}(\mathbb{R}^d) \to \mathbb{R}$$
   - $\mathrm{Seq}(\mathbb{R}^d) := \{(\mathbf{x}_{t_1}, \ldots, \mathbf{x}_{t_L}) \mid (t_i, \mathbf{x}_{t_i}) \in \mathbb{R}_+ \times \mathbb{R}^d, L \in \mathbb{N}\}$
2. Develop an efficient inference framework
   - Standard challenges: intractable posteriors, $O(N^3)$ scaling in training data

Overview

Purpose of this work

1. Define a Gaussian process (GP) [6] over sequences/time series
   - To model of functions of sequences $\{\mathrm{Seq}(\mathbb{R}^d) \to \mathbb{R}\}$
     $$(f_{\mathbf{x}})_{\mathbf{x} \in \mathrm{Seq}(\mathbb{R}^d)} \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot))$$
   - Find a suitable covariance kernel
     $$\mathrm{k} : \mathrm{Seq}(\mathbb{R}^d) \times \mathrm{Seq}(\mathbb{R}^d) \to \mathbb{R}$$
   - $\mathrm{Seq}(\mathbb{R}^d) := \{(\mathbf{x}_{t_1}, \dots, \mathbf{x}_{t_L}) \mid (t_i, \mathbf{x}_{t_i}) \in \mathbb{R}_+ \times \mathbb{R}^d, L \in \mathbb{N}\}$
2. Develop an efficient inference framework
   - Standard challenges: intractable posteriors, $O(N^3)$ scaling in training data
   - Additional challenge: potentially very high dimensional inputs (long sequences)

Overview

Suitable feature map? Signatures from stochastic analysis [2]!

Overview

Suitable feature map? Signatures from stochastic analysis [2]!

Can be used to transform vector-kernels into sequence-kernels

Overview

Suitable feature map? Signatures from stochastic analysis [2]!

Can be used to transform vector-kernels into sequence-kernels

- $\kappa : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ a kernel for vector-valued data

Overview

Suitable feature map? Signatures from stochastic analysis [2]!

Can be used to transform vector-kernels into sequence-kernels

- $\kappa : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ a kernel for vector-valued data
- [4] used signatures to define the kernel for $\mathbf{x}, \mathbf{y} \in \mathrm{Seq}(\mathbb{R}^d)$

$$\mathrm{k}(\mathbf{x}, \mathbf{y}) = \sum_{m=0}^{M} \sigma_m^2 \sum_{\substack{1 \le i_1 < \cdots < i_m \le L_{\mathbf{x}} \\ 1 \le j_1 < \cdots < j_m \le L_{\mathbf{y}}}} c(\mathbf{i}) c(\mathbf{j}) \prod_{l=1}^{m} \Delta_{i_l, j_l} \kappa(\mathbf{x}_{i_l}, \mathbf{y}_{j_l})$$

for some explicitly given constants $c(i_1, \ldots, i_m), c(j_1, \ldots, j_m)$
$\Delta_{i,j} \kappa(\mathbf{x}_i, \mathbf{y}_j) = \kappa(\mathbf{x}_{i+1}, \mathbf{y}_{j+1}) - \kappa(\mathbf{x}_i, \mathbf{y}_{j+1}) - \kappa(\mathbf{x}_{i+1}, \mathbf{y}_j) + \kappa(\mathbf{x}_i, \mathbf{y}_j)$

Overview

Suitable feature map? Signatures from stochastic analysis [2]!

Can be used to transform vector-kernels into sequence-kernels

- $\kappa : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ a kernel for vector-valued data
- [4] used signatures to define the kernel for $\mathbf{x}, \mathbf{y} \in \mathrm{Seq}(\mathbb{R}^d)$

$$\mathrm{k}(\mathbf{x}, \mathbf{y}) = \sum_{m=0}^{M} \sigma_m^2 \sum_{\substack{1 \le i_1 < \cdots < i_m \le L_{\mathbf{x}} \\ 1 \le j_1 < \cdots < j_m \le L_{\mathbf{y}}}} c(\mathbf{i}) c(\mathbf{j}) \prod_{l=1}^{m} \Delta_{i_l, j_l} \kappa(\mathbf{x}_{i_l}, \mathbf{y}_{j_l})$$

for some explicitly given constants $c(i_1, \ldots, i_m), c(j_1, \ldots, j_m)$
$\Delta_{i,j} \kappa(\mathbf{x}_i, \mathbf{y}_j) = \kappa(\mathbf{x}_{i+1}, \mathbf{y}_{j+1}) - \kappa(\mathbf{x}_i, \mathbf{y}_{j+1}) - \kappa(\mathbf{x}_{i+1}, \mathbf{y}_j) + \kappa(\mathbf{x}_i, \mathbf{y}_j)$

- Strong theoretical properties!

Overview

Our contributions

- Bringing GPs and signatures together (+analysis)

Overview

Our contributions

- Bringing GPs and signatures together (+analysis)
- Developing a tractable, efficient inference scheme

Overview

Our contributions

- Bringing GPs and signatures together (+analysis)
- Developing a tractable, efficient inference scheme
    1. Sparse VI [3]: non-conjugacy, large $N \in \mathbb{N}$

Overview

Our contributions

- Bringing GPs and signatures together (+analysis)
- Developing a tractable, efficient inference scheme
  1. Sparse VI [3]: non-conjugacy, large $N \in \mathbb{N}$
  2. Inter-domain inducing points: long sequences ($\sup_{\mathbf{x} \in \mathbf{X}} L_{\mathbf{x}}$ large)

Overview

Our contributions

- Bringing GPs and signatures together (+analysis)
- Developing a tractable, efficient inference scheme
  1. Sparse VI [3]: non-conjugacy, large $N \in \mathbb{N}$
  2. Inter-domain inducing points: long sequences ($\sup_{\mathbf{x} \in \mathbf{X}} L_{\mathbf{x}}$ large)
- GPflow implementation, thorough experimental evaluation

Signatures

Signatures

What are signatures?

Signatures

What are signatures?

Signatures are defined on continuous time objects, paths

- $\text{Paths}(\mathbb{R}^d) = \left\{ \mathbf{x} \in C([0,T], \mathbb{R}^d) \mid \mathbf{x}_0 = 0, \|\mathbf{x}\|_{bv} < +\infty \right\}$

Signatures

What are signatures?

Signatures are defined on continuous time objects, paths

- $\text{Paths}(\mathbb{R}^d) = \left\{ \mathbf{x} \in C([0,T], \mathbb{R}^d) \mid \mathbf{x}_0 = 0, \|\mathbf{x}\|_{bv} < +\infty \right\}$

$\Phi_m(\mathbf{x}) = \int_{0<t_1<\cdots<t_m<T} \dot{\mathbf{x}}_{t_1} \otimes \cdots \otimes \dot{\mathbf{x}}_{t_m} dt_1 \ldots dt_m$

Signatures

What are signatures?

Signatures are defined on continuous time objects, paths

- $\text{Paths}(\mathbb{R}^d) = \left\{ \mathbf{x} \in C([0, T], \mathbb{R}^d) \mid \mathbf{x}_0 = 0, \|\mathbf{x}\|_{bv} < +\infty \right\}$

$\Phi_m(\mathbf{x}) = \int_{0<t_1<\dots<t_m<T} \dot{\mathbf{x}}_{t_1} \otimes \cdots \otimes \dot{\mathbf{x}}_{t_m} dt_1 \dots dt_m$

$\Phi_m(\mathbf{x}) \in (\mathbb{R}^d)^{\otimes m}$ is what is known as a tensor of degree $m \in \mathbb{N}$

Signatures

What are signatures?

Signatures are defined on continuous time objects, paths

- $\text{Paths}(\mathbb{R}^d) = \left\{ \mathbf{x} \in C([0, T], \mathbb{R}^d) \mid \mathbf{x}_0 = 0, \|\mathbf{x}\|_{bv} < +\infty \right\}$

$\Phi_m(\mathbf{x}) = \int_{0<t_1<\cdots<t_m<T} \dot{\mathbf{x}}_{t_1} \otimes \cdots \otimes \dot{\mathbf{x}}_{t_m} dt_1 \dots dt_m$

$\Phi_m(\mathbf{x}) \in (\mathbb{R}^d)^{\otimes m}$ is what is known as a tensor of degree $m \in \mathbb{N}$

$\Phi(\mathbf{x}) = (\Phi_m(\mathbf{x}))_{m\geq 0}$ is an infinite collection of tensors with increasing degrees

Signatures

What are signatures?

Signatures are defined on continuous time objects, paths

- $\text{Paths}(\mathbb{R}^d) = \left\{ \mathbf{x} \in C([0,T], \mathbb{R}^d) \mid \mathbf{x}_0 = 0, \|\mathbf{x}\|_{bv} < +\infty \right\}$

$\Phi_m(\mathbf{x}) = \int_{0<t_1<\cdots<t_m<T} \dot{\mathbf{x}}_{t_1} \otimes \cdots \otimes \dot{\mathbf{x}}_{t_m} dt_1 \ldots dt_m$

$\Phi_m(\mathbf{x}) \in (\mathbb{R}^d)^{\otimes m}$ is what is known as a tensor of degree $m \in \mathbb{N}$

$\Phi(\mathbf{x}) = (\Phi_m(\mathbf{x}))_{m \geq 0}$ is an infinite collection of tensors with increasing degrees

A generalization of polynomials for vector-valued data to paths (and sequences!)

Signatures

Sequences as paths

$\mathbf{x} = (\mathbf{x}_{t_1}, \ldots, \mathbf{x}_{t_L}) \in \mathsf{Seq}(\mathbb{R}^d)$

Signatures

Sequences as paths

$\mathbf{x} = (\mathbf{x}_{t_1}, \ldots, \mathbf{x}_{t_L}) \in \mathrm{Seq}(\mathbb{R}^d)$

Define a mapping $\mathrm{Seq}(\mathbb{R}^d) \to \mathrm{Paths}(\mathbb{R}^d)$

Signatures

Sequences as paths

$\mathbf{x} = (\mathbf{x}_{t_1}, \ldots, \mathbf{x}_{t_L}) \in \mathrm{Seq}(\mathbb{R}^d)$

Define a mapping $\mathrm{Seq}(\mathbb{R}^d) \to \mathrm{Paths}(\mathbb{R}^d)$

Straightforward choice? Linear interpolation!

$$t \mapsto (t_{i+1} - t_i)^{-1}(\mathbf{x}_{t_i}(t_{i+1} - t) + \mathbf{x}_{t_{i+1}}(t - t_i) \text{ for } t \in [t_i, t_{i+1})$$

Signatures

Sequences as paths

$\mathbf{x} = (\mathbf{x}_{t_1}, \ldots, \mathbf{x}_{t_L}) \in \mathrm{Seq}(\mathbb{R}^d)$

Define a mapping $\mathrm{Seq}(\mathbb{R}^d) \to \mathrm{Paths}(\mathbb{R}^d)$

Straightforward choice? Linear interpolation!

$$t \mapsto (t_{i+1} - t_i)^{-1}(\mathbf{x}_{t_i}(t_{i+1} - t) + \mathbf{x}_{t_{i+1}}(t - t_i)) \text{ for } t \in [t_i, t_{i+1})$$

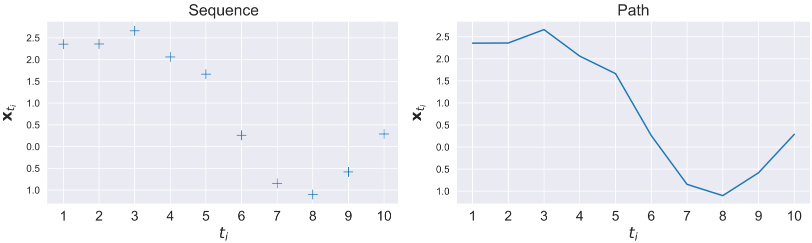


Figure: Linear interpolation of a sequence

Signatures

What makes the signature a good feature map

Signatures

What makes the signature a good feature map

- continuous time treatment of sequences

Signatures

What makes the signature a good feature map

- continuous time treatment of sequences
- same feature space for sequences of different length

Signatures

What makes the signature a good feature map

- continuous time treatment of sequences
- same feature space for sequences of different length
- universally approximates functions of sequences (paths)

Signatures

What makes the signature a good feature map

- continuous time treatment of sequences
- same feature space for sequences of different length
- universally approximates functions of sequences (paths)
- learn the extent of parametrization (in)variance

Signatures

More on parametrization invariance

Signatures

More on parametrization invariance

Paths can be broken down into two constituents:

More on parametrization invariance

Paths can be broken down into two constituents:

- trajectory

Signatures

More on parametrization invariance

Paths can be broken down into two constituents:

- trajectory
- parametrization

More on parametrization invariance

Paths can be broken down into two constituents:

- trajectory
- parametrization

Trajectory: an ordered collection of points the path crosses

Signatures

More on parametrization invariance

Paths can be broken down into two constituents:

- trajectory
- parametrization

Trajectory: an ordered collection of points the path crosses
Parametrization: the speed at which the trajectory is traversed

Signatures

More on parametrization invariance

Paths can be broken down into two constituents:

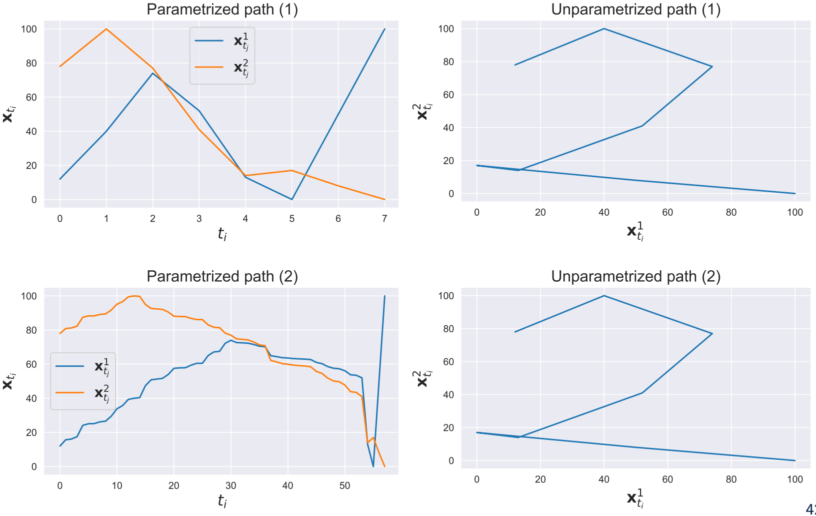
- trajectory
- parametrization

Trajectory: an ordered collection of points the path crosses
Parametrization: the speed at which the trajectory is traversed
Parametrization invariance: only takes the trajectory into account, but factors out the parametrization

Signatures

Parametrization invariance: an illustration

Parametrization invariance: an illustration



Signatures

What the signature can do for you

Signatures

What the signature can do for you

- Compare sequences of different length (same feature space)

Signatures

What the signature can do for you

- Compare sequences of different length (same feature space)
- Approximate functions of sequences (universality)

Signatures

What the signature can do for you

- Compare sequences of different length (same feature space)
- Approximate functions of sequences (universality)
- Learn functions of sequences that depend only on the trajectory (parametrization invariance)

Signatures

What the signature can do for you

- Compare sequences of different length (same feature space)
- Approximate functions of sequences (universality)
- Learn functions of sequences that depend only on the trajectory (parametrization invariance)
- Deal with irregularly sampled time series (parametrization invariance)

Signatures

What the signature can do for you

- Compare sequences of different length (same feature space)
- Approximate functions of sequences (universality)
- Learn functions of sequences that depend only on the trajectory (parametrization invariance)
- Deal with irregularly sampled time series (parametrization invariance)
- Deal with high-dimensional sequences (kernelization)

Signatures

What the signature can do for you

- Compare sequences of different length (same feature space)
- Approximate functions of sequences (universality)
- Learn functions of sequences that depend only on the trajectory (parametrization invariance)
- Deal with irregularly sampled time series (parametrization invariance)
- Deal with high-dimensional sequences (kernelization)
- $+1$: Learn degree of smoothness by choice of base kernel, e.g. RBF, Matérn (kernelization)

Signatures

Take away. signature features have many attractive properties for modelling sequences, and they can be kernelized to define Gaussian processes over sequences and paths

Experiments

Experiments

Compared GPs with signature covariances on 16 multivariate TSC datasets against baselines:

Experiments

Compared GPs with signature covariances on 16 multivariate TSC datasets against baselines:

- Recurrent deep kernels (LSTM, GRU) [1]

Experiments

Compared GPs with signature covariances on 16 multivariate TSC datasets against baselines:

- Recurrent deep kernels (LSTM, GRU) [1]
- Convolutional kernels [5]

Experiments

Compared GPs with signature covariances on 16 multivariate TSC datasets against baselines:

- Recurrent deep kernels (LSTM, GRU) [1]
- Convolutional kernels [5]

GPs with signatures consistently performed well, while alternatives did good on some datasets, but very poorly on others

Experiments

Compared GPs with signature covariances on 16 multivariate TSC datasets against baselines:

- Recurrent deep kernels (LSTM, GRU) [1]
- Convolutional kernels [5]

GPs with signatures consistently performed well, while alternatives did good on some datasets, but very poorly on others

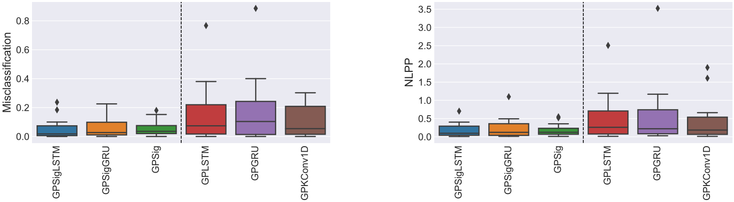


Figure: Box-plots of misclassification errors and negative log-predictive probabilities (NLPP) on 16 multivariate time series classification datasets

Further reading

Further reading

Signatures are an exciting new way of modelling sequential data

Further reading

Signatures are an exciting new way of modelling sequential data

Feature extraction

- Rough paths, Signatures and the modelling of functions on streams, arXiv:1405.4537, 2014.
- A Primer on the Signature Method in Machine Learning, arXiv:1603.03788, 2016.
- A Generalised Signature Method for Time Series, arXiv:2006.00873, 2020.

Further reading

Signatures are an exciting new way of modelling sequential data

Feature extraction

- Rough paths, Signatures and the modelling of functions on streams, arXiv:1405.4537, 2014.
- A Primer on the Signature Method in Machine Learning, arXiv:1603.03788, 2016.
- A Generalised Signature Method for Time Series, arXiv:2006.00873, 2020.

Nonparametric methods

- Kernels for sequentially ordered data, Journal of Machine Learning Research, 2019.
- Signature moments to characterize laws of stochastic processes, arXiv:1810.10971, 2018.
- Persistence paths and signature features in topological data analysis, arXiv:1806.00381, 2018.
- This work

Further reading

Deep learning

- Sparse arrays of signatures for online character recognition, arXiv:1308.0371, 2013.
- Learning stochastic differential equations using RNN with log signature features, arXiv:1908.08286, 2019.
- Deep Signature Transforms, 33rd Conferenceon Neural Information Processing Systems, NeurIPS, 2019.
- Seq2Tens: An Efficient Representation of Sequences by Low-Rank Tensor Projections, arXiv:2006.07027, 2020.

... and many more!

Thank you!

C. Toth, and H. Oberhauser,
"Bayesian Learning from Sequential Data using Gaussian Processes with Signature Covariances"

References

Maruan Al-Shedivat, Andrew Gordon Wilson, Yunus Saatchi, Zhiting Hu, and Eric P. Xing.
Learning scalable deep kernels with recurrent structure.
J. Mach. Learn. Res., 18(1):2850–2886, January 2017.

I. Chevyrev and A. Kormilitzin.
A primer on the signature method in machine learning.
arXiv preprint arXiv:1603.03788, 2016.

James Hensman, Alexander G. de G. Matthews, and Zoubin Ghahramani.
Scalable variational gaussian process classification.
JMLR Workshop and Conference Proceedings, 2015.

Franz J Király and Harald Oberhauser.
Kernels for sequentially ordered data.
Journal of Machine Learning Research, 2019.

References

Mark van der Wilk, Carl Edward Rasmussen, and James Hensman.
Convolutional gaussian processes.
In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2849–2858. Curran Associates, Inc., 2017.

Christopher KI Williams and Carl Edward Rasmussen.
Gaussian processes for machine learning, volume 2.
MIT press Cambridge, MA, 2006.