Learning from the order of events

Durham LMS meeting, July 2017

Harald Oberhauser

Mathematical Institute, University of Oxford

Two common learning tasks

$\mathcal{X}$ topological space in which data lives, e.g. $\mathbb{R}^n$, a manifold, space of graphs, space of paths, etc.

- make inference about a function $f \in \mathbb{R}^{\mathcal{X}}$
- make inference about a probability measure on $\mathcal{X}$

Two common learning tasks

$\mathcal{X}$ topological space in which data lives, e.g. $\mathbb{R}^n$, a manifold, space of graphs, space of paths, etc.

- make inference about a function $f \in \mathbb{R}^{\mathcal{X}}$
- make inference about a probability measure on $\mathcal{X}$

This talk:

- $\mathcal{X}$ space of paths
- Examples: text, evolution of a social network, rough paths/semimartingales, diffusions,...

Inference on pathspace studied by different communities:

- Statistics/stochastic analysis approach. Focus on parametrized models. Typically Ito diffusions and stochastic calculus. Very few truly nonparametric results.
- Machine learning: Focus on black box/non-parametric approaches and efficient algorithms. Most in discrete time

Mathematical difficulties if data is path-valued

- infinite dimensional and non-locally compact
- computational complexity

Learning

- **Stylized facts.**
  - data nonlinear
  - scaleable learning algorithms are linear
- **Feature map Φ**
  - map $\mathcal{X}$ into a linear space; run learning algorithm there
    - linearize functionals $f(x) \simeq \langle \Phi(x), \ell \rangle$
    - efficiently computable
    - robust

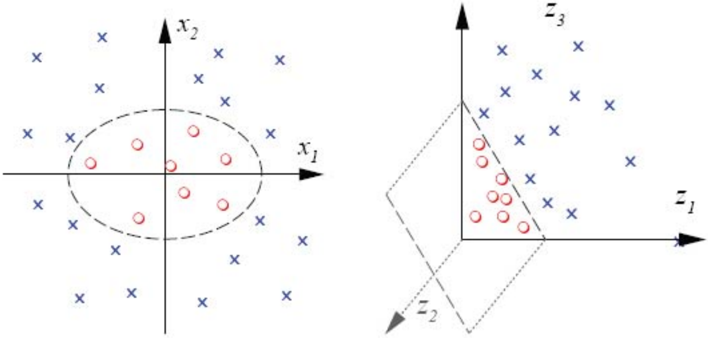


Figure: $\Phi : \mathbb{R}^2 \to \mathbb{R}^3$, $(x_1, x_2) \mapsto (x_1^2, \sqrt{2}x_1x_2, x_2^2)$

- Signature as a feature map?

$$\Phi(x) = \left(\int dx^{\otimes m}\right)_{m \geq 0}$$

- **Issues**
  1. Combinatorial explosion! $O\left(d^M\right)$ coordinates for d-dimensional path and up to m-iterated integrals
  2. Signature of paths in non-linear or infinite dimensional space? E.g. network evolution, SPDE, etc.

Rest of talk

1. Randomization (with Terry Lyons)
2. Kernelization (with Franz Kiraly)
3. Expected signatures (with Ilya Chevyrev)

Randomization (with Terry Lyons)

Example

- $\mathcal{X} = \{1, \ldots, 10^{38}\}$ IP addresses
- $\sigma = (\sigma_i)_{i=1}^{L} \in \mathcal{X}^L$ requests to a server from IP addresses
- Engineer: most active IP addresses over a month?
  i.e. compute $\Phi(\sigma) = \left(\sum_{i:\sigma_i = x} 1\right)_{x \in \mathcal{X}} \in \mathbb{R}^{|\mathcal{X}|}$

Example

- $\mathcal{X} = \{1, \ldots, 10^{38}\}$ IP addresses
- $\sigma = (\sigma_i)_{i=1}^{L} \in \mathcal{X}^L$ requests to a server from IP addresses
- Engineer: most active IP addresses over a month?
  i.e. compute $\Phi(\sigma) = \left(\sum_{i:\sigma_i = x} 1\right)_{x \in \mathcal{X}} \in \mathbb{R}^{|\mathcal{X}|}$
- **Naive algorithm** $|\mathcal{X}|$ counters and parse once over stream
  - needs $O(|\mathcal{X}|)$ space...infeasible

Example

- $\mathcal{X} = \{1, \ldots, 10^{38}\}$ IP addresses
- $\sigma = (\sigma_i)_{i=1}^{L} \in \mathcal{X}^L$ requests to a server from IP addresses
- Engineer: most active IP addresses over a month?
  i.e. compute $\Phi(\sigma) = \left(\sum_{i:\sigma_i = x} 1\right)_{x \in \mathcal{X}} \in \mathbb{R}^{|\mathcal{X}|}$
- **Naive algorithm** $|\mathcal{X}|$ counters and parse once over stream
  - needs $O(|\mathcal{X}|)$ space...infeasible
- **Randomized algorithm**: compute *random variable* $\hat{\Phi}$
  - $\hat{\Phi}(\sigma) \approx \Phi(\sigma)$ for big coordinates with high probability
  - sublinear space complexity & single pass over σ
  - Work of: Flajolet, Alon, Matias, Szegedy, Charikar, Chen, Colton, Cormode, Muthukrishnan,...

Massive data streams

- $\sigma \in \mathcal{X}^L$ for $\mathcal{X}$ large set
- Compute $\Phi(\sigma) = \left(\sum_{i:\sigma_i = x} 1\right)_{x \in \mathcal{X}}$
- **Randomized algorithm**
  - Fix "small set" $\mathcal{Y}$ with $|\mathcal{Y}| \ll |\mathcal{X}|$
  - sample random function $h : \mathcal{X} \to \mathcal{Y}$
  - Calculate $\Phi(h(\sigma))$
  - Define $\Phi^h(\sigma)$ as $\left\langle \Phi^h(\sigma), x \right\rangle := \langle \Phi(h(\sigma)), h(x) \rangle$
  - Sample several h, take $\left\langle \hat{\Phi}(\sigma), x \right\rangle := \min_h \Phi^h(x)$
- Easy to extend to $\sigma \in (\mathbb{R} \times \mathcal{X})^L$

Proof: elementary

$$
\begin{aligned}
\mathbb{E}\left[\langle\Phi(h(\sigma)), h(x)\rangle - \langle\Phi(\sigma), x\rangle\right] &= \mathbb{E}\left[\sum_{i:h(\sigma_i)=h(x)} 1\right] - \sum_{i:\sigma_i=x} 1 \\
&= \sum_{i:\sigma_i\neq x} \mathbb{E}\left[1_{h(\sigma_i)=h(x)}\right] \\
&= \sum_{i:\sigma_i\neq x} |\mathcal{Y}|^{-1} \leq |\sigma|\,|\mathcal{Y}|^{-1}
\end{aligned}
$$

- $\forall\epsilon > 0$, $\mathbb{P}\left(\langle\Phi(h(\sigma)), h(x)\rangle - \langle\Phi(\sigma), x\rangle > \epsilon\,|\sigma|\right) \leq \frac{1}{2}$ for $\mathcal{Y} := \left\{1, \ldots, \left\lceil\frac{2}{\epsilon}\right\rceil\right\}$
- repeat k times; then $\left\langle\hat{\Phi}(\sigma), x\right\rangle := \min_h \langle\Phi(h(\sigma)), h(x)\rangle$ gives $\mathbb{P}\left(\left\langle\hat{\Phi}(\sigma), x\right\rangle - \langle\Phi(\sigma), x\rangle > \epsilon\,|\sigma|\right) \leq 2^{-k}$

Massive data streams

- $\sigma \in \mathcal{X}^L$ for $\mathcal{X}$ large set
- Compute $\Phi(\sigma) = \left(\sum_{i:\sigma_i = x} 1\right)_{x \in \mathcal{X}}$
- **Sketch algorithm**:
  - Given ϵ, δ, compute random variable $\hat{\Phi}(\sigma)$

$$\mathbb{P}\left(\frac{\left\langle \hat{\Phi}(\sigma), x\right\rangle - \langle \Phi(\sigma), x\rangle}{|\Phi(\sigma)|_1} > \epsilon\right) \geq 1 - \delta$$

    where $|\Phi(\sigma)|_1 = \sum_{x \in \mathcal{X}} \left(\sum_{i:\sigma_i = x} 1\right)$
  - Complexity: single pass over σ, $O\left(\frac{1}{\epsilon} \log \frac{1}{\delta}\right)$ space and $\log \frac{1}{\delta} \log |\mathcal{X}|$ random bits
  - Compressed sensing: linear projection via hashes and ℓ_1-norm. Difference: projection more structure
- Much information about path lost
- Above is first level of the signature of a lattice path in $|\mathcal{X}| = 10^{38}$ dimensions...

Streams, paths, polynomials

- Fix "**event map**"

$$\gamma : \mathcal{X} \mapsto \mathbb{R}\langle\langle \mathcal{X} \rangle\rangle$$

  from $\mathcal{X} = \{x_1, \ldots, x_d\}$ into
  $\mathbb{R}\langle\langle \mathcal{X} \rangle\rangle = \left\{ \sum_{i_1,\ldots,i_m} c_{i_1 \ldots i_m} x_{i_1} \cdots x_{i_m} \right\}$
- Extend to $\mathcal{X}^L$ by multiplication

$$\Phi : \mathcal{X}^L \to \mathbb{R}\langle\langle \mathcal{X} \rangle\rangle, \ \sigma \mapsto \prod_{i=1}^{L} \gamma(\sigma_i)$$

Example: $\sigma = (a, b, b, a)$

- with $\gamma(x) = 1 + x$,

$$\begin{aligned}\Phi(\sigma) &= \prod_{i=1}^{L} \gamma(\sigma_i) = (1+a)(1+b)(1+b)(1+a) \\ &= 1 + 2a + 2b + a^2 + 2ab + b^2 + 2ba\end{aligned}$$

- with $\gamma(x) = 1 + x + \frac{x^2}{2!} + \cdots$,

$$\begin{aligned}\Phi(\sigma) &= \prod_{i=1}^{L} \gamma(\sigma_i) = \left(1 + a + \frac{a^2}{2!} + \cdots\right) \cdots \left(1 + a + \frac{a^2}{2!} + \cdots\right) \\ &= 1 + 2a + 2b + \left(1 + \frac{1}{2!} + \frac{1}{2!}\right) a^2 + \cdots\end{aligned}$$

- Latter is the standard rough paths; good scaling limit, rich mathematical structure (Hopf algebra of shuffles)
- First recovers standard ML features (string kernels). We will see that there's also Hopf algebra structure (with different coproduct)

Hopf algebras

- Consider an algebra (A, m), where $m : A \otimes A \to A$ denotes multiplication
- Define $\Delta : A^\star \otimes A^\star \to A^\star$ as $\langle \Delta(a), b \otimes c \rangle := \langle a, m(b \otimes c) \rangle$. Then $(A^\star, \Delta^\star)$ is a so-called **co-algebra**
- Applied to two "compatible" algebra structures (A, m) and $(A^\star, m^\star)$. Then

$$(A, m, \Delta_{m^\star})$$

  a so-called **bi-algebra.**
- If A is additionally graded **Hopf algebra.**
- $\mathcal{G}(A) = \{a \in A : \Delta(a) = a \otimes a\}$ is a group

Hopf algebras

- Consider an algebra (A, m), where $m : A \otimes A \to A$ denotes multiplication
- Define $\Delta : A^\star \otimes A^\star \to A^\star$ as $\langle \Delta(a), b \otimes c \rangle := \langle a, m(b \otimes c) \rangle$. Then $(A^\star, \Delta^\star)$ is a so-called **co-algebra**
- Applied to two "compatible" algebra structures (A, m) and $(A^\star, m^\star)$. Then

$$(A, m, \Delta_{m^\star})$$

  a so-called **bi-algebra.**
- If A is additionally graded **Hopf algebra.**
- $\mathcal{G}(A) = \{a \in A : \Delta(a) = a \otimes a\}$ is a group

- **Our setting:**
  - $A = \mathbb{R}\langle \mathcal{X} \rangle$, $A^\star = \mathbb{R}\langle\langle \mathcal{X} \rangle\rangle$
  - non-commutative multiplication in $\mathbb{R}\langle\langle \mathcal{X} \rangle\rangle$ concatenation
  - commutative multiplication in $\mathbb{R}\langle \mathcal{X} \rangle$ implies $f(\sigma) \simeq \langle \Phi(\sigma), \ell \rangle$

Back to "rough paths"

- Finite set $\mathcal{X}$, sequence $\sigma \in \mathcal{X}^L$
- Fix map $\gamma : \mathcal{X} \mapsto \mathbb{R}\langle\langle\mathcal{X}\rangle\rangle$ and define $\Phi : \mathcal{X}^L \to \mathbb{R}\langle\langle\mathcal{X}\rangle\rangle$ as $\Phi(\sigma) = \prod_{i=1}^{L} \gamma(\sigma_i)$
- Feature space $\Phi(\sigma) \in \mathbb{R}\langle\langle\mathcal{X}\rangle\rangle$. Algebra using concatention product m_{concat}
- Linear functionals $\mathbb{R}\langle\mathcal{X}\rangle$. Algebra using $m_{shuffle}$

Back to "rough paths"

- Finite set $\mathcal{X}$, sequence $\sigma \in \mathcal{X}^L$
- Fix map $\gamma : \mathcal{X} \mapsto \mathbb{R}\langle\langle\mathcal{X}\rangle\rangle$ and define $\Phi : \mathcal{X}^L \to \mathbb{R}\langle\langle\mathcal{X}\rangle\rangle$ as $\Phi(\sigma) = \prod_{i=1}^{L} \gamma(\sigma_i)$
- Feature space $\Phi(\sigma) \in \mathbb{R}\langle\langle\mathcal{X}\rangle\rangle$. Algebra using concatention product m_{concat}
- Linear functionals $\mathbb{R}\langle\mathcal{X}\rangle$. Algebra using $m_{shuffle}$

Theorem (Sweedler, Reutenauer, etc.)

With $\gamma(x) = \exp(x)$, $\Phi(\sigma) = \prod_{i=1}^{L} \gamma(\sigma_i)$

- $\langle \Phi(\sigma), w \rangle = \sum_{\boldsymbol{i} \in \Delta} \frac{1}{\boldsymbol{i}!} 1_{\sigma_{i_1} \cdots \sigma_{i_M} = w}$,
- $(\mathbb{R}\langle\mathcal{X}\rangle, m_{shuffle}, \Delta_{concat})$ *is a commutative Hopf algebra*

Back to "rough paths"

- Finite set $\mathcal{X}$, sequence $\sigma \in \mathcal{X}^L$
- Fix map $\gamma : \mathcal{X} \mapsto \mathbb{R}\langle\langle\mathcal{X}\rangle\rangle$ and define $\Phi : \mathcal{X}^L \to \mathbb{R}\langle\langle\mathcal{X}\rangle\rangle$ as $\Phi(\sigma) = \prod_{i=1}^{L} \gamma(\sigma_i)$
- Feature space $\Phi(\sigma) \in \mathbb{R}\langle\langle\mathcal{X}\rangle\rangle$. Algebra using concatention product m_{concat}
- Linear functionals $\mathbb{R}\langle\mathcal{X}\rangle$. Product that turns it into commutative algebra?

Back to "rough paths"

- Finite set $\mathcal{X}$, sequence $\sigma \in \mathcal{X}^L$
- Fix map $\gamma : \mathcal{X} \mapsto \mathbb{R}\langle\langle\mathcal{X}\rangle\rangle$ and define $\Phi : \mathcal{X}^L \to \mathbb{R}\langle\langle\mathcal{X}\rangle\rangle$ as $\Phi(\sigma) = \prod_{i=1}^{L} \gamma(\sigma_i)$
- Feature space $\Phi(\sigma) \in \mathbb{R}\langle\langle\mathcal{X}\rangle\rangle$. Algebra using concatention product m_{concat}
- Linear functionals $\mathbb{R}\langle\mathcal{X}\rangle$. Product that turns it into commutative algebra?

Theorem (Lyons&O)

With $\gamma(x) = 1 + x$, $\Phi(\sigma) = \prod_{i=1}^{L} \gamma(x)$

- $\langle \Phi(\sigma), w\rangle = \sum_{(i_1,\ldots,i_M)\in\Delta} 1_{\sigma_{i_1}\cdots\sigma_{i_M}=w}$,
- $(\mathbb{R}\langle\mathcal{X}\rangle, m_{inf}, \Delta_{concat})$ *is a commutative Hopf algebra*

Back to "rough paths"

- **Goal:** approximate

$$\Phi(\sigma) = \prod_{i=1}^{L} \gamma(\sigma_i) \in \mathbb{R}\langle\langle \mathcal{X} \rangle\rangle$$

with random variable $\hat{\Phi}(\sigma)$

Back to "rough paths"

- **Goal:** approximate

$$\Phi(\sigma) = \prod_{i=1}^{L} \gamma(\sigma_i) \in \mathbb{R}\langle\langle \mathcal{X} \rangle\rangle$$

  with random variable $\hat{\Phi}(\sigma)$

- **Step 1**. Fix $\mathcal{Y}$, $|\mathcal{Y}| \ll |\mathcal{X}|$, sample uniformly $h : \mathcal{X} \to \mathcal{Y}$
- **Step 2.** Calculate $\Phi(h(\sigma)) \in \mathbb{R}\langle\langle \mathcal{Y} \rangle\rangle$
- **Step 3.** Repeat steps 1&2 several times; combine $\Phi(h(\sigma)) \in \mathbb{R}\langle\langle \mathcal{Y} \rangle\rangle$ to one estimator for $\Phi(\sigma) \in \mathbb{R}\langle\langle \mathcal{X} \rangle\rangle$

Step 1. Universal hashing

- **Step 1**. Fix small set $\mathcal{Y}$, sample uniformly $h : \mathcal{X} \to \mathcal{Y}$
- Sampling uniformly from $\mathcal{Y}^{\mathcal{X}}$ is too expensive: $|\mathcal{Y}|^{|\mathcal{X}|}$ possible choices; specifying h costs $O(|\mathcal{X}| \log |\mathcal{Y}|)$
- If h drawn uniformly from $\mathcal{Y}^{\mathcal{X}}$, then $\mathbb{P}(h(x) = h(y)) = |\mathcal{Y}|^{-1}$ for $x, y \in \mathcal{X}$, $x \neq y$

Step 1. Universal hashing

- **Step 1**. Fix small set $\mathcal{Y}$, sample uniformly $h : \mathcal{X} \to \mathcal{Y}$
- Sampling uniformly from $\mathcal{Y}^{\mathcal{X}}$ is too expensive: $|\mathcal{Y}|^{|\mathcal{X}|}$ possible choices; specifying h costs $O(|\mathcal{X}| \log |\mathcal{Y}|)$
- If h drawn uniformly from $\mathcal{Y}^{\mathcal{X}}$, then $\mathbb{P}(h(x) = h(y)) = |\mathcal{Y}|^{-1}$ for $x, y \in \mathcal{X}$, $x \neq y$

Definition

$\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ is called 2-universal if h is drawn uniformly from $\mathcal{H}$

$$\mathbb{P}(h(a) = h(b)) = |\mathcal{Y}|^{-1} \text{ for } a, b \in \mathcal{X}, a \neq b$$

Step 1. Universal hashing

- **Step 1**. Fix small set $\mathcal{Y}$, sample uniformly $h : \mathcal{X} \to \mathcal{Y}$
- Sampling uniformly from $\mathcal{Y}^{\mathcal{X}}$ is too expensive: $|\mathcal{Y}|^{|\mathcal{X}|}$ possible choices; specifying h costs $O(|\mathcal{X}| \log |\mathcal{Y}|)$
- If h drawn uniformly from $\mathcal{Y}^{\mathcal{X}}$, then $\mathbb{P}(h(x) = h(y)) = |\mathcal{Y}|^{-1}$ for $x, y \in \mathcal{X}$, $x \neq y$

Definition

$\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ is called 2-universal if h is drawn uniformly from $\mathcal{H}$

$$\mathbb{P}(h(a) = h(b)) = |\mathcal{Y}|^{-1} \text{ for } a, b \in \mathcal{X}, a \neq b$$

Example. Fix prime $p \geq |\mathcal{X}|$.

$$\mathcal{H} = \{h_{a,b} | h_{a,b}(x) = (((ax + b) \mod p) \mod m), 1 \leq a \leq p - 1, 0 \leq$$

is 2-universal. Choosing a random element of $\mathcal{H}$ requires $2 \log p$ random bits.

Step 2

Step 2. Calculate $\Phi(h(\sigma)) \in \mathbb{R}\langle\langle\mathcal{Y}\rangle\rangle$, estimate $\Phi(\sigma)$

Proposition

Let $h \in \mathcal{Y}^{\mathcal{X}}$ and $\sigma \in \mathcal{X}^L$. Define $\Phi_h(\sigma)$ as $\langle\Phi_h(\sigma), w\rangle := \langle\Phi(h(\sigma)), h(w)\rangle$. Then

$$\Phi_h(\sigma) = \Phi(\sigma) + b \text{ and } \langle b, w\rangle = \sum_{\substack{\boldsymbol{i}=(i_1,\dots,i_M)\\ i_1<\dots<i_M}} 1_{\sigma(\boldsymbol{i})\neq w}$$

Corollary

Let h be choosen uniformly from a universal hash family $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$, then

$$\mathbb{P}\left(\langle\Phi(\sigma), w\rangle \in \left[\langle\Phi_h(\sigma), w\rangle - \frac{2\left\|\Phi^{|w|}(\sigma)\right\|_1}{|\mathcal{Y}|}, \langle\Phi_h(\sigma), w\rangle\right]\right) \geq \frac{1}{2}$$

Randomized algorithms

Theorem (Lyons&O 16)

$\mathcal{X}$ finite set, $\Phi(\sigma) \in \mathbb{R}\langle\langle\mathcal{X}\rangle\rangle$ signature of $\sigma \in \mathcal{X}^L$. For any $\epsilon, \delta > 0$ there exists a random variable $\hat{\Phi}(\sigma)$ such that

1. $\mathbb{P}\left(\frac{\left|\left\langle\hat{\Phi}(\sigma), w\right\rangle - \langle\Phi(\sigma), w\rangle\right|}{\sum_{|v|=|w|}|\langle\Phi(\sigma), v\rangle|} > \epsilon\right) < \delta$
2. *for $M \geq 1$ the set of coordinates*

$$\left\{\left\langle\hat{\Phi}(\sigma), w\right\rangle : |w| \leq M\right\}$$

can be calculated using $O\left(\epsilon^{-M} \log\frac{1}{\delta}\right)$ memory units, $\lceil -\log\delta\rceil \log|\mathcal{X}|$ random bits and a single pass over σ.

Remark

Extends to $\sigma \in (\mathbb{R} \times \mathcal{X})^L$. Good estimate if few "heavy hitter patterns"

$\lvert\mathcal{Y}\rvert$	Nr. of hashes	letters/second	$\frac{\text{memory for } \Phi(\sigma)}{\text{memory for } \hat{\Phi}(\sigma)}$	$\ell\left(\Phi\left(\sigma\right), \hat{\Phi}\left(\sigma\right)\right)$
4	8	17651.8	1503.13	2927.01
4	16	9120.63	751.56	2086.38
4	32	4620.79	375.78	2061.50
8	8	3411.47	216.20	293.34
8	16	1712.27	108	268.00
8	32	850.85	54.05	230.30
16	8	390.48	28.91	38.66
16	16	194.98	14.45	33.14
16	32	97.213	7.23	26.29
32	8	195.25	3.73	5.01
32	16	97.93	1.87	4.41
32	32	49.21	0.99	3.60

Table: 10 letters appear 10 percent of the time, the rest of the events is uniformly distributed among the remaining 90 letters.

II. Kernelization (with Franz Kiraly)

- feature map $x \mapsto \Phi(x)$ typically computationally expensive.
- Kernel learning (Aizerman'64, Wahba'90, Vapnik'95, Smale'00,...)
    - often an inner product $\langle \Phi(x), \Phi(y) \rangle$ makes sense & computationally cheap
    - many learning algorithms depend only on $\langle \Phi(x), \Phi(y) \rangle$
    - with
$$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}, \quad (x, y) \mapsto \langle \Phi(x), \Phi(y) \rangle$$

      our features take value in reproducing kernel Hilbert space $(\mathcal{H}, k)$

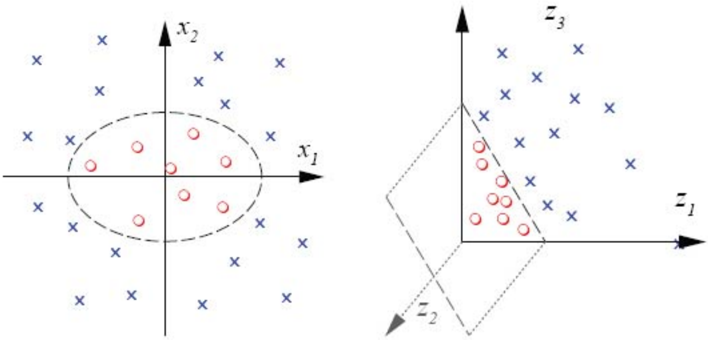


Figure: $\Phi : \mathbb{R}^2 \to \mathbb{R}^3$, $(x_1, x_2) \mapsto \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right)$ costs $O\left(d^2\right)$. But $k\left(x, y\right) = \left\langle \Phi\left(x\right), \Phi\left(y\right)\right\rangle = \left\langle x, y\right\rangle^2$ costs $O\left(d\right)$. **Exponential saving!**

Kernel learning

- **(+)** rich literature of kernels for *static* non-linear data
  - e.g. kernels for graphs, images, molecules,... (constructed using expert domain knowledge)
- **(+)** modularity:
  - evaluate kernel matrix $(k(x,y))_{x,y\in\mathcal{X}}$
  - plug into kernelized algorithm
- **(+)** quantified Occam's razor: PAC/VC/Rademacher bounds (Vapnik, Smale, ...)
- **(-)** possible issues: huge matrix $(k(x,y))_{x,y\in\mathcal{X}}$, Hilbert norm as regularizer, ...
- **(-)** not so much literature for sequences of observations (**BUT:** string kernels)

Kernelized signatures

- **Key remark**: How to evaluate univariate polynomial $P \in \mathbb{R}[X]$?
  - Horner scheme! $P(x) = c_0 + X(c_1 + X(c_2 \cdots))$
  - already non-trivial for $\mathbb{R}[X, Y]$; truncated signature is "non-commutative polynomial" $\mathbb{R}\langle \mathcal{X} \rangle$
- **Signature Horner type scheme:** Let $\sigma, \tau \in C^1([0,1], \mathcal{H})$ and $\Phi(\sigma) = \left(\int d\sigma^{\otimes m}\right)_m$

$$
\begin{aligned}
& k(\sigma, \tau) := \langle \Phi(\sigma), \Phi(\tau) \rangle \\
:= \quad & 1 + \left\langle \int d\sigma, \int d\tau \right\rangle_{\mathcal{H}} + \cdots + \left\langle \int d\sigma^{\otimes M}, \int d\tau^{\otimes M} \right\rangle_{\mathcal{H}^{\otimes M}} \\
= \quad & \sum_{m=0}^{M} \int_{s_1, t_1} \left\langle \int d\sigma^{\otimes(m-1)}, \int d\tau^{\otimes(m-1)} \right\rangle_{\mathcal{H}^{\otimes(m-1)}} d\langle \sigma_{s_1}, \tau_{t_1} \rangle_{\mathcal{H}} \\
= \quad & 1 + \int_{s_1, t_1} \left(1 + \int_{s_2, t_2} \left(1 + \cdots \int_{s_M, t_M} d\langle \sigma_{s_M}, \tau_{t_M} \rangle_{\mathcal{H}}\right) \cdots d\langle \sigma_{s_1}, \tau
\end{aligned}
$$

  - only evalutate $\langle \sigma_s, \tau_t \rangle_{\mathcal{H}}$ for $s, t \in [0,1]$...can be cheap, even if $\mathcal{H}$ is infinite dimenensional & recursive evalution!

Theorem (Kiraly&O '16)

Let $\sigma, \tau \in C^1([0,1], \mathcal{H})$ *and*

$$k : C^1 \times C^1 \to \mathbb{R}$$

defined as inner product of their signatures. Then there exists a positive definite kernel

$$k_\oplus : \bigcup_L \mathcal{H}^L \times \bigcup_L \mathcal{H}^L \to \mathbb{R}$$

such that

1. $|k_\oplus(\sigma^\pi, \tau^\pi) - k(\sigma, \tau)| \leq O(mesh(\pi))$ *for any partition* $\pi = (t_i) \subset [0,1]$,
2. $k_\oplus(\sigma^\pi, \tau^\pi)$ *can be evaluated with...*

Complexity

algorithm	steps	storage
A	$O\left(c \cdot M \cdot L^2\right)$	$O\left(L^2\right)$
B	$O\left(c \cdot M \cdot \rho \cdot L\right)$	$O\left(L \cdot \rho\right)$

where
c cost of evaluating $\langle \cdot, \cdot \rangle_{\mathcal{H}}$
L number of time points
M truncation level of tensor algebra
ρ low rank approximation meta parameter

Remark

For paths in $\mathcal{H} = \mathbb{R}^d$

$$k_{\oplus}\left(\sigma, \tau\right) = \left\langle \Phi\left(\sigma\right), \Phi\left(\tau\right) \right\rangle = \sum_{m=0}^{M} \left\langle \int d\sigma^{\otimes m}, \int d\tau^{\otimes m} \right\rangle_{\left(\mathbb{R}^d\right)^{\otimes m}}$$

needs $O\left(d \cdot M \cdot \rho \cdot L\right)$. Compare to $O\left(d^M L\right)$ for direct feature evaluation.

Black box to produce features for paths/sequences

- Data in some space $\mathcal{X}$ (e.g. networks) and we are given a feature map

$$\varphi : \mathcal{X} \to \mathcal{H}$$

- Now observe data in $\mathcal{X}$ over time (e.g. network evolution)
- Kernelization allows to use the signature of this infinite dimensional path for learning!
- Canonical method to transform from static to dynamic features
- **Fun fact**: already powerful with $\mathcal{X} = \mathbb{R}^d$ low dimensional and φ a nonlinearity

toy example: pendigts

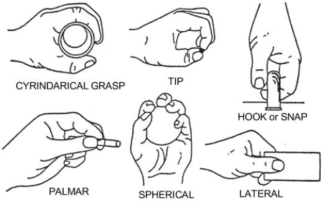
$$\mathcal{D} = \left\{ (x_i, y_i) \in \left(\mathbb{R}^2\right)^7 \times \{0, \ldots, 9\}, i = 1, \ldots, 7494 \right\}$$

label	precision	recall	f1-score	support
0.0	0.96	1.00	0.98	363
1.0	0.88	0.45	0.59	364
2.0	0.73	1.00	0.85	364
3.0	0.85	0.99	0.92	336
4.0	1.00	0.99	0.99	364
5.0	0.94	0.88	0.91	335
6.0	0.96	0.97	0.96	336
7.0	0.91	0.85	0.88	364
8.0	0.98	0.97	0.98	336
9.0	0.88	0.94	0.91	336
average/sum	0.91	0.90	0.89	total 3498

label	precision	recall	f1-score	support
0.0	1.00	0.99	1.00	363
1.0	0.98	0.99	0.98	364
2.0	0.99	1.00	0.99	364
3.0	0.87	0.99	0.92	336
4.0	0.96	1.00	0.98	364
5.0	0.97	0.92	0.94	335
6.0	1.00	0.99	1.00	336
7.0	0.98	0.92	0.95	364
8.0	0.97	0.98	0.97	336
9.0	0.96	0.88	0.92	336
average/sum	0.97	0.97	0.97	3498

Gesture recognition

$$\mathcal{D} = \left\{(x_i, y_i) \in \left(\mathbb{R}^2\right)^{3000} \times \{1, \ldots, 6\}\right\}$$



label	precision	recall	f1-score	support
1.0	0.66	0.83	0.74	30
2.0	0.88	0.77	0.82	30
3.0	0.88	0.77	0.82	30
4.0	0.87	0.90	0.89	30
5.0	0.97	0.93	0.95	30
6.0	0.93	0.93	0.93	30
avg/ total	0.87	0.86	0.86	total 180

- no feature extraction & beats baseline

III. Expected signatures (with Ilya Chevyrev)

- Let X, Y be random variables taking values in a topological space $\mathcal{X}$
- Hypothesis test

$$H_0 : X =^{\text{Law}} Y \text{ versus } H_1 : X \neq^{\text{Law}} Y$$

given iid samples $X_1, \ldots, X_n \sim X$ and $Y_1, \ldots Y_n \sim Y$
- Our motivation X, Y path-valued random variables, i.e. stochastic processes

Metrics on measures

- Fix $\mathcal{F} \subset \mathbb{R}^{\mathcal{X}}$ and define

$$d(\mu,\nu) := \sup_{f\in\mathcal{F}} \left| \int_{\mathcal{X}} f(x)\,\mu(dx) - \int_{\mathcal{X}} f(x)\,\nu(dx) \right|$$
$$= \sup_{f\in\mathcal{F}} \left| \mathbb{E}_{X\sim\mu}[f(X)] - \mathbb{E}_{Y\sim\nu}[f(X)] \right|$$

- If $\mathcal{F}$ is big enough, this becomes a metric; e.g. $C_b(\mathcal{X})$, $\{f : \sup|f(x)| \leq 1\}$, $\left\{f : |f|_{Lip} \leq 1\right\}$,...
- Test if $d(\mu,\nu) = 0$ or > 0
- Bad news: computing d is typically hard due to supremum

Metrics from RKHS

- Let $\mathcal{F}$ be unit ball in a RKHS $(\mathcal{H}, k)$. Denote

$$\mu_k := \int k(x,\cdot)\,\mu(dx) \in \mathcal{H}$$

By reproducing property

$$\begin{aligned} d(\mu,\nu) &= \sup_{f\in\mathcal{F}} \left| \int f(x)\,\mu(dx) - \int f(x)\,\nu(dx) \right| \\ &= \sup_{f\in\mathcal{F}} \left| \langle f, \mu_k - \nu_k \rangle_{\mathcal{H}} \right| \\ &= |\mu_k - \nu_k|_{\mathcal{H}} = \int k(x,y)\,(\mu-\nu)^{\otimes 2}(dx \otimes dy) \\ &= \mathbb{E}_{X\sim\mu, X'\sim\mu}\left[k(X,X')\right] - 2\mathbb{E}_{X\sim\mu, Y\sim\nu}\left[k(X,Y)\right] + \mathbb{E}_{Y\sim\nu, Y'} \end{aligned}$$

- Easy to estimate from finite samples! Leads to uniformly most powerful tests (Gretton et. al)
- Put differently: if feature map $\Phi : \mathcal{X} \to \mathcal{H}$ can be kernelized, above gives optimal tests via expected features

Theorem (Chevyrev&O)

There exists a kernel

$$k : C^1 \times C^1 \to \mathbb{R}$$

such that

$$d(\mu, \nu) := \mathbb{E}_{X \sim \mu, X' \sim \mu}\left[k(X, X')\right] - 2\mathbb{E}_{X \sim \mu, Y \sim \nu}\left[k(X, Y)\right] + \mathbb{E}_{Y \sim \nu, Y' \sim \nu}\left[k\right.$$

is a metric on Borel probablity measures on C^1 and k is cheap to evaluate.

- Extends from C^1 to branched rough paths and to signed measures on paths
- Equivalent to "expected signature characterizes measures"
- Completely non-parametric testing in Neyman-Pearson setting $H_0 : d(\mu, \nu) = 0$ vs $H_1 : d(\mu, \nu) \neq 0$.

Summary: from stochastic analysis to ML and back

- **Randomization**
  - signatures often computable in high dimensions ($d \sim 10^6$ on a standard desktop)
- **Kernelization**
  - Special cases of signatures classic in ML literature (e.g. string/alignment/Anova kernels)
  - Black box to turn static into dynamic features:
    - canonical: input is kernel, output is kernel for sequences in data
    - general PAC learning guarantees apply
  - Easy to implement: algorithms vectorized
- **Hypothesis testing**
  - ML literature provides kernel based MMD
  - combined with signatures:
    - non-parametric(!) tests for pathvalued random variables
    - new results about expected signatures

THANKS FOR YOUR TIME!