What can kernel methods offer to HeP?

Lorenzo Rosasco

main coauthors Marco Letizia (MaLGa), Gaia Grosso (MIT), Marco Zanetti (UniPadova), Andrea Wulzer (ICREAS), Maurizio Pierini (CERN)

NP-Twins 2024 Genova

December 18th, 2024

UniGe | DIBRIS | MaLGa MACHINE LEARNING GENOA CENTER | European Research Council Established by the European Commission

The way of ML

$$(x_i, y_i)_{i=1}^n \rightsquigarrow f : \mathcal{X} \to \mathbb{R}$$

a) $f_w, \quad w \in \mathbb{R}^p$ **model**

b) $\hat{w} = \arg\min_w \sum_{i=1}^{n/2} (y_i - f_w(x_i))^2$ **fit**

c) $\sum_{i=n/2+1}^{n} (y_i - f_{\hat{w}}(x_i))^2$ **test**

The way of ML

$$(x_i, y_i)_{i=1}^n \rightsquigarrow f : \mathcal{X} \to \mathbb{R}$$

a) $f_w, \quad w \in \mathbb{R}^p, \qquad n << p$ **model**

b) $\hat{w} = \arg\min_w \sum_{i=1}^{n/2} (y_i - f_w(x_i))^2 \approx 0$ **fit**

c) $\sum_{i=n/2+1}^{n} (y_i - f_{\hat{w}}(x_i))^2$ **test**

With four parameters I can fit an elephant, and with five I can make him wiggle his trunk?

Outline

Machine learning with kernels

Large scale machine learning with kernels

Discovering anomalies with kernels

Models

- Linear models

$$f_w(x) = \langle w, x \rangle.$$

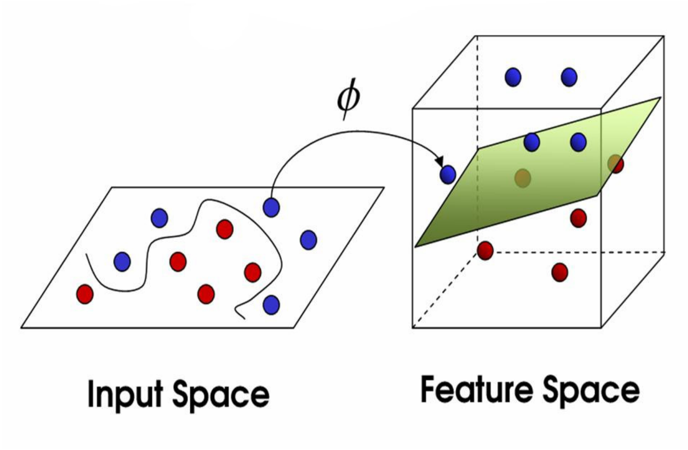
- Perceptron and neural nets

$$f_w(x) = \sigma(\langle w, x \rangle), \qquad f_w(x) = \sum_{j=1}^{u} c_i \sigma(\langle a_j, x \rangle).$$

- Kernel methods

$$f_w(x) = \langle w, \Phi(x) \rangle.$$

Just a trick!



Kernel methods for adults

Reproducing kernel Hilbert space (RKHS) [Aronzajn '50]

$\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ Hilbert space with a reproducing kernel $\exists\, k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that

- for all $x \in \mathcal{X}$,
$$k_x = k(x, \cdot) \in \mathcal{H},$$
- for all $x \in \mathcal{X}, f \in \mathcal{H}$,
$$f(x) = \langle f, k_x \rangle_{\mathcal{H}}$$

.

Examples with $\mathcal{X} \subset \mathbb{R}^d$

- Band limited functions, $\rightarrow\ k(x, x') = \text{sinc}(x - x')$
- Analytic functions, $\rightarrow\ k(x, x') = e^{-\|x - x'\|^2}$
- Sobolev spaces $W^{s,2}(\mathbb{R}^d)$, $s = 2d\ \rightarrow\ k(x, x') = e^{-\|x - x'\|}$

Fitting with kernels

$$\hat{f}_\lambda = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}}^2$$

Theorem **[Kimeldorf, Wahba, '70]**

$$\hat{f}_\lambda(x) = \sum_{i=1}^{n} k(x, x_i)\hat{c}_i, \quad \hat{c}_i \in \mathbb{R}$$

$$\hat{c} = (\hat{K} + \lambda I)^{-1}\hat{y} \qquad \hat{K}_{ij} = k(x_i, x_j) \qquad \hat{y} = [y_1, \dots, y_n]$$

Time complexity: $O(n^3)$ Space complexity: $O(n^2)$

Testing with kernels

$$L(f) = \int (y - f(x))^2 dP(x, y)$$

P probability on $(\mathcal{X} \times \mathbb{R})$ s.t. $(x_i, y_i,)_{i=1}^n \sim P^n$.

Theorem **[Caponnetto, De Vito, '07]**

If $k(x, x') \leq 1$, $y \leq M$ *a.s. and* $\exists f_{\mathcal{H}} \in \mathcal{H}$ *s.t.* $L(f_{\mathcal{H}}) = \min_{f \in \mathcal{H}} L(f)$.

Then, choosing $\lambda = \frac{1}{\sqrt{n}}$

$$\mathbb{E}[L(\hat{f}_\lambda) - L(f_{\mathcal{H}})] \lesssim \frac{1}{\sqrt{n}}.$$

Remarks

- **History**. 1970. 2000. Now.
- Issue 1: No **feature learning** .
- Issue 2: **Scaling** issues.

Outline

Machine learning with kernels

Large scale machine learning with kernels

Discovering anomalies with kernels

Models for large scale kernel methods

- Random Features [Rahimi, Recht '08]

$$z_i \in \mathbb{R}^m \quad \text{such that} \quad \langle z_i, z_j \rangle_{\mathbb{R}^m} \approx k(x_i, x_j)$$

- **Random subspaces** (aka Nyström method/inducing points) [Williams, Seeger '00]

$$\mathcal{H}_m = \overline{\operatorname{span}\{k_{\tilde{x}_1}, \dots, k_{\tilde{x}_m}\}} \subset \mathcal{H} \qquad \{\tilde{x}_1, \dots, \tilde{x}_m\} \subset \{x_1, \dots, x_n\}$$

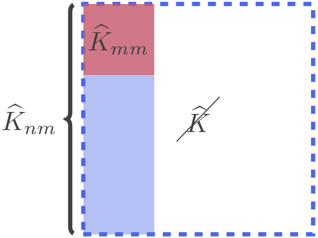
Fitting large scale kernel methods

$$\hat{f}_{\lambda,m} = \arg\min_{f \in \mathcal{H}_m} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|f\|^2_{\mathcal{H}_m}$$

Theorem [Williams, Seeger '00]

$$\hat{f}_{\lambda,m} = \sum_{i=1}^{m} k(\cdot, \tilde{x}_i)\hat{c}_i, \quad \hat{c}_i \in \mathbb{R}$$

$$\hat{c} = (\widehat{K}_{nm}^\top \widehat{K}_{nm} + \lambda \widehat{K}_{mm})^{-1} \widehat{K}_{nm}^\top \hat{y}$$



Time complexity: $O(n^2 + m^3)$ Space complexity: $O(nm)$

Testing large scale kernel methods

$$L(f) = \int (y - f(x))^2 dP(x, y)$$

Theorem [Rudi, Camoriano, Rosasco, '16]

If $k(x, x') \leq 1$, $y \leq M$ *a.s. and* $\exists f_{\mathcal{H}} \in \mathcal{H}$ *s.t.* $L(f_{\mathcal{H}}) = \min_{f \in \mathcal{H}} L(f)$.

Then, with $\lambda = \frac{1}{\sqrt{n}}$ *and* $m \gtrsim \sqrt{n}$

$$\mathbb{E}[L(\hat{f}_{\lambda,m}) - L(f_{\mathcal{H}})] \lesssim \frac{1}{\sqrt{n}}.$$

Going faster with randomized linear algebra

$$\beta_t = \beta_{t-1} + \frac{\tau}{n} B^\top [\widehat{K}_{nm}^\top (\widehat{K}_{nm} B \beta_{t-1} - \hat{y}) + n\lambda \widehat{K}_{mm} B \beta_{t-1}] \qquad c_t = B\beta_t$$

a) Iterative solvers (e.g. Gradient descent, bonjugate gradient)

b) Condition number and preconditioning

$$\kappa = \frac{\sigma_{\max}(\widehat{K}_{nm}^\top \widehat{K}_{nm} + \lambda \widehat{K}_{mm})}{\sigma_{\min}(\widehat{K}_{nm}^\top \widehat{K}_{nm} + \lambda \widehat{K}_{mm})}$$

c) Compressed preconditioningt

$$BB^\top = (\frac{n}{m} \widehat{K}_{mm}^2 + \lambda \widehat{K}_{mm})^{-1}$$

["FALKON: An Optimal Large Scale Kernel Method", Rudi, Carratino, Rosasco, 2018]

Falkon Software

GPU

- out-of-core operations
- multi-GPU support

Memory

- preconditioner
- matrix-vector products
- low precision floats

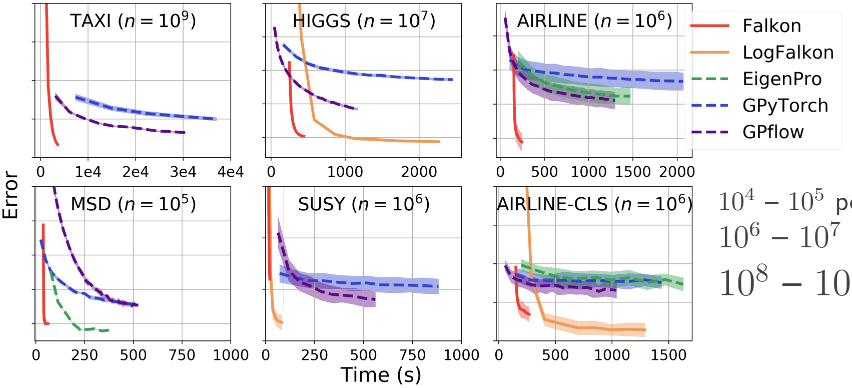
Communications

- Optimized memory transfers

$20\times$ Improvement over strong baseline

["Kernel methods through the roof", M., Carratino, Rosasco, Rudi, 2020]

Falkon Experiments



$10^4 - 10^5$ points in seconds

$10^6 - 10^7$ points in minutes

$10^8 - 10^9$ points in hours

[“Kernel methods through the roof”, M., Carratino, Rosasco, Rudi, 2020]

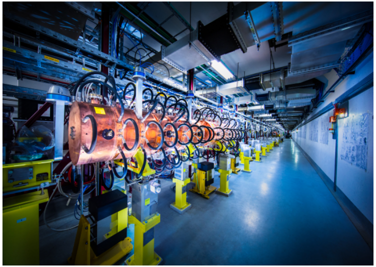
Outline

Machine learning with kernels

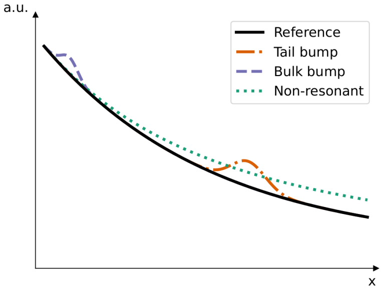
Large scale machine learning with kernels

Discovering anomalies with kernels

Anomalies aka new physics



© Andrew Hara



A ML approach to anomalies I

- Data

$$x_1, \ldots, x_M \sim P_{\text{mother nature}}.$$

- Model

$$x_1, \ldots, x_N \sim P_{\text{model}}.$$

Idea: binary classification

NATURE vs MODEL

But the model is good $\Longrightarrow$ "*Accuracy*$= 50.5\%$".

A ML approach to anomalies II

Is *Accuracy*$= \approx 50,5\%$ significant?

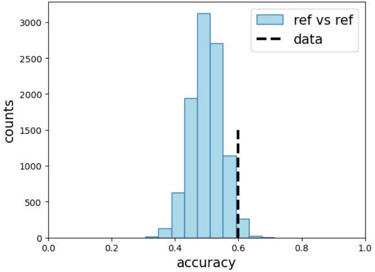
- Permutation test.
- ...
- Exploit physics

$$x_1, \ldots, x_M \sim P_{\text{model}}.$$

$$x_1, \ldots, x_N \sim P_{\text{model}}.$$

Get null distribution classifying

MODEL vs MODEL



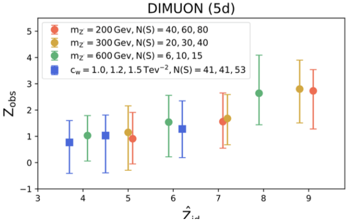
Some results

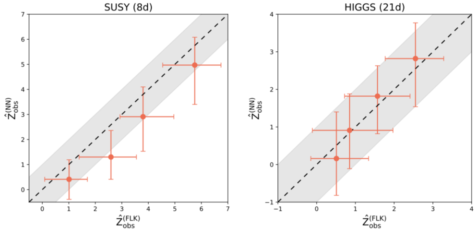
$pp \to \mu^+\mu^-$ $[p_{T1}, p_{T2}, \eta_1, \eta_2, \Delta\phi]$,

$N(R) = 2\times10^4$, $\mathcal{N}_{\mathcal{R}} = 5\times N(R)$.

SUSY (8d), HIGGS (21d)

$N(R) = 10^5$, $\mathcal{N}_{\mathcal{R}} = 5\times N(R)$





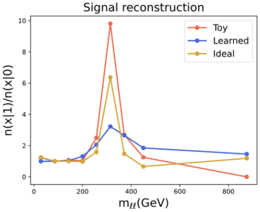


Table 1 Average training times per single run with standard deviations (low level features and reference toys). Note that time measured in hours (for NN) and seconds (for Falkon)

Model	DIMUON	SUSY	HIGGS
FLK	**(44.9 ± 3.4) s**	**(18.2 ±1.2) s**	**(22.7 ± 0.4) s**
NN	(4.23 ± 0.73) h	(73.1 ± 10) h	(112 ± 9) h

Bold values indicate the lowest for each column (lower is better)

Data: https://zenodo.org/records/4442665

Wrap up

- Kernel method can run on **millions/billions** points.
- Great model for **intermediate** dimensions (or pretrained features!).
- **HeP a natural test bed?** New physics, data quality monitoring, generative modeling quality. . . (**SEE Marco Letizia's TALK**)

Ongoing

- Not just supervised learning: physics informed ML, dynamical systems.
- Kernel design/learning?

(Come work @MaLGa– DM for info)

European Research Council
Established by the European Commission