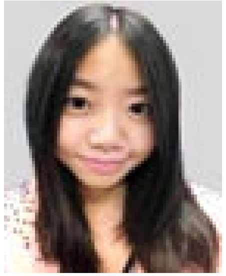
*Bellairs Research Institute*
Barbados, February 11, 2019

# Online Scalable Learning Adaptive to Unknown Dynamics and Graphs

*Y. Shen*



***Georgios B. Giannakis***

*T. Chen*

Dept. of ECE and Digital Tech. Center, University of Minnesota

**Acknowledgments:** NSF 1500713,1711471, NIH 1R01GM104975-01
Huawei Inc, gift 2018; and Prof. Geert Leus

UNIVERSITY OF MINNESOTA
Driven to Discover℠

# Roadmap

❑ Motivation and prior art

❑ Multi-kernel learning (MKL) via random feature (RF) approximation

❑ Online MKL with RF in environments with unknown dynamics

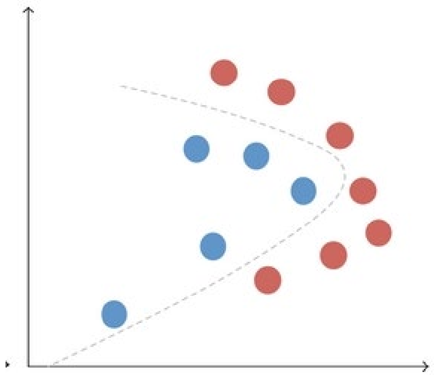
❑ Performance via regret analysis and real data tests

❑ Online MKL over graphs

# Motivation

- ❑ Nonlinear function models widespread in real-world applications

**Nonlinear dimension reduction**



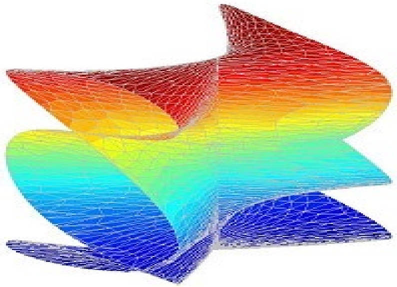
**Nonlinear classification**

Tesla Closing Price
close
dale_bucks
Stock Price
$400
$350
$300
$250
$200
$150
$100
$50
$0
2011
2012
2013
2014
2015
2016
2017
2018
Year

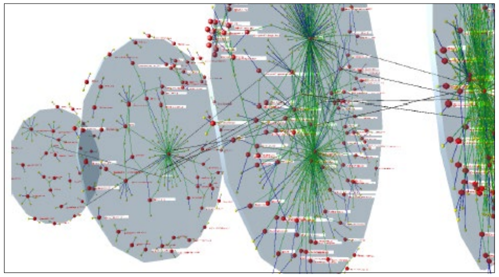
**Nonlinear regression**

- ❑ Challenges and opportunities

**Massive scale**



**Unknown nonlinearity**



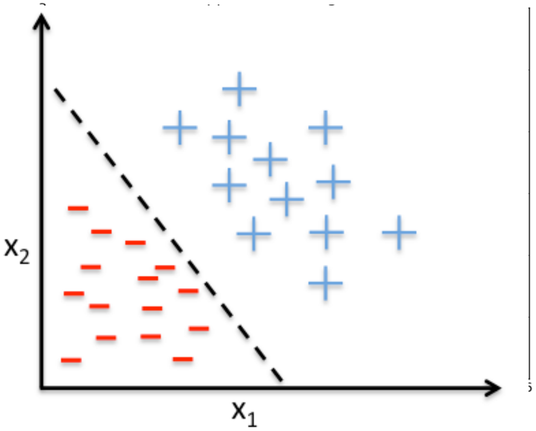
**Unknown dynamics**

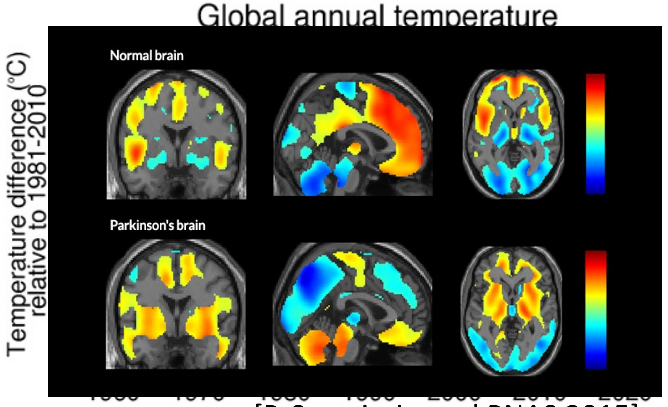
# Learning functions from data

**Goal:** Given data $\{(\mathbf{x}_t, y_t)\}_{t=1}^{T}$ , find *f* to model $y_t = f(\mathbf{x}_t) + e_t$

**Ex1**. Regression: $y_t = \boldsymbol{\theta}^\top \mathbf{x}_t + e_t$ Curve fitting for e.g. temperature forecasting

**Ex2**. Classification: $y_t = \mathrm{sign}(\boldsymbol{\theta}^\top \mathbf{x}_t + \mathbf{b})$ For e.g., disease diagnosis





Global annual temperature
Temperature difference relative to 1981-2010 (°C)
Normal brain
Parkinson's brain

[P. Spetsieris et al PNAS 2015]

- ❑ Even unsupervised tasks boil down to function learning
  - ➢ E.g., dimensionality reduction, clustering, anomaly detection …

# Learning functions with kernels

**Goal:** Given data $\{(\mathbf{x}_t, y_t)\}_{t=1}^{T}$ , find $f$ to model $y_t = f(\mathbf{x}_t) + e_t$

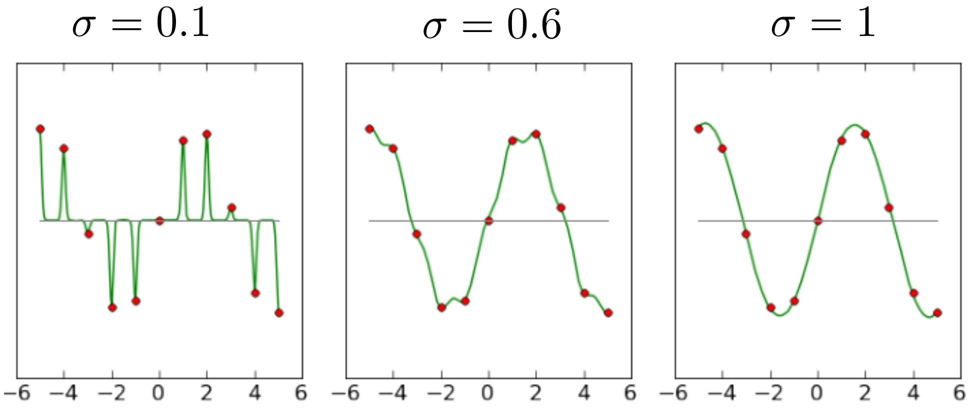
- ❑ Reproducing kernel Hilbert space (RKHS) $\mathcal{H} := \{f \,|\, f(\mathbf{x}) = \sum_{t=1}^{\infty} \alpha_t \kappa(\mathbf{x}, \mathbf{x}_t)\}$

kernel

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{T} \sum_{t=1}^{T} \mathcal{C}(f(\mathbf{x}_t), y_t) + \lambda \Omega\left(\|f\|_{\mathcal{H}}^2\right)$$

cost regularizer

**Ex.** Gaussian (RBF) kernel $\kappa(\mathbf{x}, \mathbf{x}_t) = \kappa(\mathbf{x} - \mathbf{x}_t) = \exp(-\|\mathbf{x} - \mathbf{x}_t\|_2^2/\sigma^2)$


$\sigma = 0.1$ $\sigma = 0.6$ $\sigma = 1$

How can we choose the appropriate kernel?

# The curse of dimensionality

$$[\mathbf{k}(\mathbf{x})]_t = \kappa(\mathbf{x}, \mathbf{x}_t)$$

$$[\mathbf{K}]_{t,t'} = \kappa(\mathbf{x}_t, \mathbf{x}_{t'})$$

- ❑ Representer Thm. $\hat{f}(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t \kappa(\mathbf{x}, \mathbf{x}_t) := \boldsymbol{\alpha}^\top \mathbf{k}(\mathbf{x})$

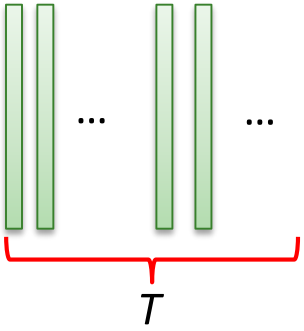
$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^T} \frac{1}{T} \sum_{t=1}^{T} \mathcal{C}(\boldsymbol{\alpha}^\top \mathbf{k}(\mathbf{x}_t), y_t) + \lambda \Omega \left( \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} \right)$$

➢ $\boldsymbol{\alpha} \in \mathbb{R}^T$ , complexity grows with *T*

Curse of Dimensionality (CoD)!

**Ex.** L2-norm cost and L2-norm regularizer: ridge regression $\mathcal{O}(T^3)$

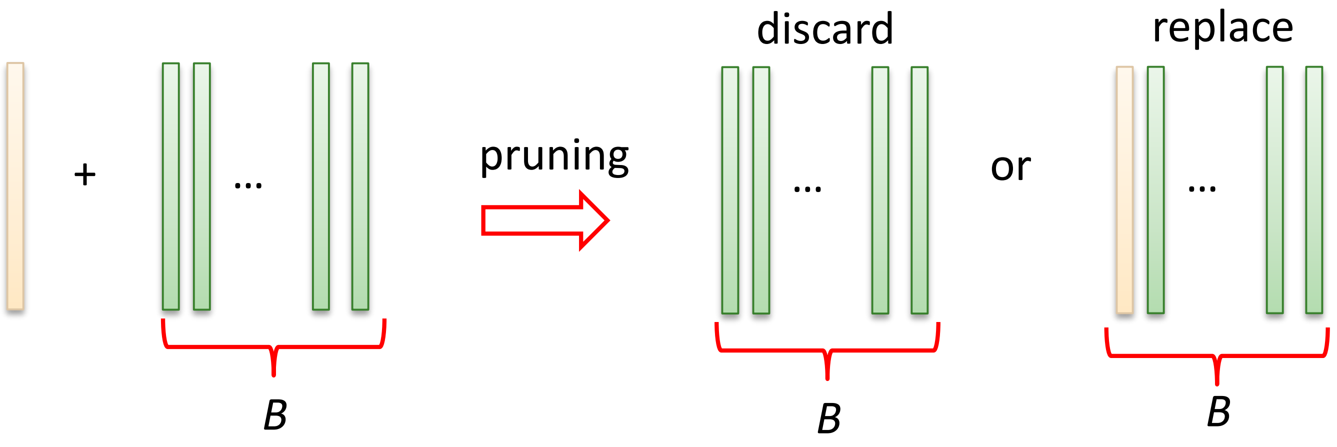
- ❑ Keep all data samples in memory



- ❑ Not scalable; and not suitable for streaming data

# Budget-constrained approaches

- Budget-constrained kernel-based learning (KL-B) [Kivinen et al' 04], [Dekel et al' 08]
  - Keep *B* data samples  in memory



**Challenges**: choice of *B*? Adaptivity to unknown dynamics?

# Random features for kernel-based learning

**Key idea:** View normalized shift-invariant kernels as characteristic functions

$$\kappa(\mathbf{x}_t, \mathbf{x}_{t'}) = \kappa(\mathbf{x}_t - \mathbf{x}_{t'}) = \int \pi_\kappa(\mathbf{v}) e^{j\mathbf{v}^\top(\mathbf{x}_t - \mathbf{x}_{t'})} d\mathbf{v} := \mathbb{E}_{\mathbf{v}}\left[e^{j\mathbf{v}^\top(\mathbf{x}_t - \mathbf{x}_{t'})}\right]$$

- ❑ Draw *D* random vectors from pdf $\pi_\kappa(\mathbf{v})$ to find kernel estimate

$$\hat{\kappa}_c(\mathbf{x}_t, \mathbf{x}_{t'}) := \frac{1}{D}\sum_{i=1}^{D} e^{j\mathbf{v}_i^\top(\mathbf{x}_t - \mathbf{x}_{t'})} \qquad e^{j\mathbf{v}_i^\top \mathbf{x}} = \cos(\mathbf{v}_i^\top \mathbf{x}) + j\sin(\mathbf{v}_i^\top \mathbf{x})$$

- ❑ Unbiased estimator $\hat{\kappa}(\mathbf{x}_t, \mathbf{x}_{t'}) = \mathbf{z}_{\mathbf{V}}^\top(\mathbf{x}_t)\mathbf{z}_{\mathbf{V}}(\mathbf{x}_{t'})$ via *2D*x*1* **random feature** (RF) vector

$$\mathbf{z}_{\mathbf{V}}(\mathbf{x}) = \frac{1}{\sqrt{D}}\left[\sin(\mathbf{v}_1^\top \mathbf{x}), \ldots, \sin(\mathbf{v}_D^\top \mathbf{x}), \cos(\mathbf{v}_1^\top \mathbf{x}), \ldots, \cos(\mathbf{v}_D^\top \mathbf{x})\right]^\top$$

- ❑ Function estimate $\hat{f}^{\mathrm{RF}}(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t \hat{\kappa}(\mathbf{x}_t, \mathbf{x}) = \sum_{t=1}^{T} \alpha_t \mathbf{z}_{\mathbf{V}}^\top(\mathbf{x}_t)\mathbf{z}_{\mathbf{V}}(\mathbf{x}) := \boldsymbol{\theta}^\top \mathbf{z}_{\mathbf{V}}(\mathbf{x})$

$\kappa \Updownarrow \pi_\kappa(\mathbf{v}) \Rightarrow$

$\{\mathbf{v}_i\}_{i=1}^{D} \Rightarrow$ RFs

$\frac{1}{D}\sum \Rightarrow \hat{\kappa} \Updownarrow \hat{f}^{\mathrm{RF}}$

$\boldsymbol{\theta} \in \mathbb{R}^{2D}$

Dimensionality not growing with *T*

A. Rahimi and B. Recht, “Random features for large scale kernel machines,” *Proc. Advances in Neural Info. Process. Syst.*, pp. 117-1184, Canada, Dec. 2008.

# Multi-kernel learning

- Given dictionary of kernels $\{\kappa_p\}_{p=1}^P$ , let $f(\mathbf{x}) := \sum_{p=1}^{P} \bar{w}_p f_p(\mathbf{x})$

$$\min_{\{\bar{w}_p\},\{f_p \in \mathcal{H}_p\}} \frac{1}{T} \sum_{t=1}^{T} \mathcal{C}\left(\sum_{p=1}^{P} \bar{w}_p f_p(\mathbf{x}_t), y_t\right) + \lambda \Omega \left(\left\| \sum_{p=1}^{P} \bar{w}_p f_p \right\|_{\bar{\mathcal{H}}}^2\right)$$

$$\text{s. to} \quad \sum_{p=1}^{P} \bar{w}_p = 1, \;\; \bar{w}_p \geq 0$$

  - Richer space of functions, but batch MKL also challenged by the CoD

- **Idea:** RFs to the rescue $\hat{f}_p(\mathbf{x}) = \boldsymbol{\theta}_p^\top \mathbf{z}_{\mathbf{V}_p}(\mathbf{x})$

$$\min_{\{\bar{w}_p\},\{\boldsymbol{\theta}_p\}} \frac{1}{T} \sum_{t=1}^{T} \sum_{p=1}^{P} \bar{w}_p \, \mathcal{C}\left(\boldsymbol{\theta}_p^\top \mathbf{z}_{\mathbf{V}_p}(\mathbf{x}), y_t\right) + \lambda \sum_{p=1}^{P} \bar{w}_p \, \Omega\left(\|\theta_p\|^2\right)$$

  - Online loss per kernel-based learner $\hat{f}_p(\mathbf{x}_t)$

$$\mathcal{L}_t(f_p(\mathbf{x}_t)) := \mathcal{C}(\boldsymbol{\theta}_p^\top \mathbf{z}_p(\mathbf{x}_t), y_t) + \lambda \Omega\left(\|\boldsymbol{\theta}_p\|^2\right)$$

# Random feature based multi-kernel learning

- **Raker**: Acquire data vector $\boldsymbol{x}_t$ per slot $t$, and run

**S1.** Parameter update

$$\boldsymbol{\theta}_{p,t+1} = \boldsymbol{\theta}_{p,t} - \eta \nabla \mathcal{L}_t(\boldsymbol{\theta}_{p,t}^\top \mathbf{z}_p(\mathbf{x}_t), y_t)$$

**S2.** Weight update

KL-divergence

$$w_{p,t+1} = \arg\min_{w_p} \eta\, \mathcal{L}_t\left(\hat{f}_{p,t}^{\mathrm{RF}}(\mathbf{x}_t)\right)(w_p - w_{p,t}) + w_p \log(w_p / w_{p,t})$$

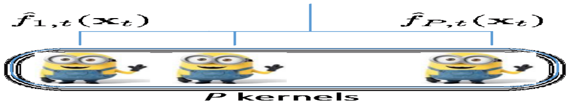
$$w_{p,t+1} = w_{p,t} e^{-\eta \mathcal{L}_t\left(\hat{f}_{p,t}^{\mathrm{RF}}(\mathbf{x}_t)\right)} \qquad \bar{w}_{p,t+1} = w_{p,t+1} / \sum_p w_{p,t+1}$$

**S3.** Function update

$$\hat{f}_{p,t+1}^{\mathrm{RF}}(\mathbf{x}_{t+1}) = \boldsymbol{\theta}_{p,t+1}^\top \mathbf{z}_p(\mathbf{x}_{t+1}) \qquad \hat{f}_{t+1}^{\mathrm{RF}}(\mathbf{x}_{t+1}) := \sum_{p=1}^{P} \bar{w}_{p,t+1} \hat{f}_{p,t+1}^{\mathrm{RF}}(\mathbf{x}_{t+1})$$

Y. Shen, T. Chen, and G. B. Giannakis, "Online Ensemble Multi-kernel Learning Adaptive to Non-stationary and Adversarial Environments," *Proc. of Intl. Conf. on Artificial Intelligence and Statistics*, Lanzarote, April 9-11, 2018.

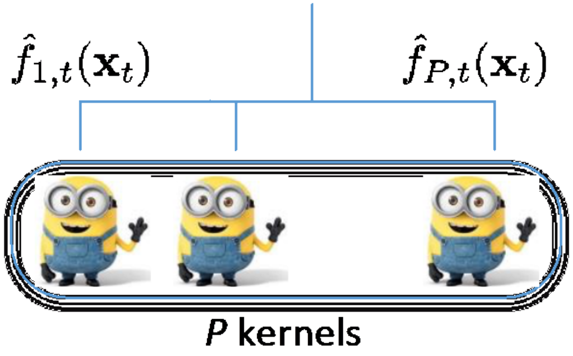
# Intuition and complexity of Raker



- ❑ function update

$$\hat{f}^{\mathrm{RF}}_{t+1}(\mathbf{x}_{t+1}) := \sum_{p=1}^{P} \bar{w}_{p,t} \hat{f}^{\mathrm{RF}}_{p,t+1}(\mathbf{x}_{t+1})$$

$$\hat{f}^{\mathrm{RF}}_{p,t+1}(\mathbf{x}_{t+1}) = \boldsymbol{\theta}^{\top}_{p,t+1} \mathbf{z}_p(\mathbf{x}_{t+1})$$



- ❑ Online (ensemble) learning with expert advice
  - ➢ **Self-**improvement of each expert (by updating $\boldsymbol{\theta}_{p,t}$ per RF kernel estimator)

- ❑ Per iteration complexity comparison with online (O) MKL and budgeted (B) MKL

| MKL | OMKL | OMKL-B | Raker |
|---|---|---|---|
| $\mathcal{O}(t^3 P)$ | $\mathcal{O}(tP)$ | $\mathcal{O}(BP)$ | $\mathcal{O}(DP)$ |

# Adaptive Raker for unknown dynamics

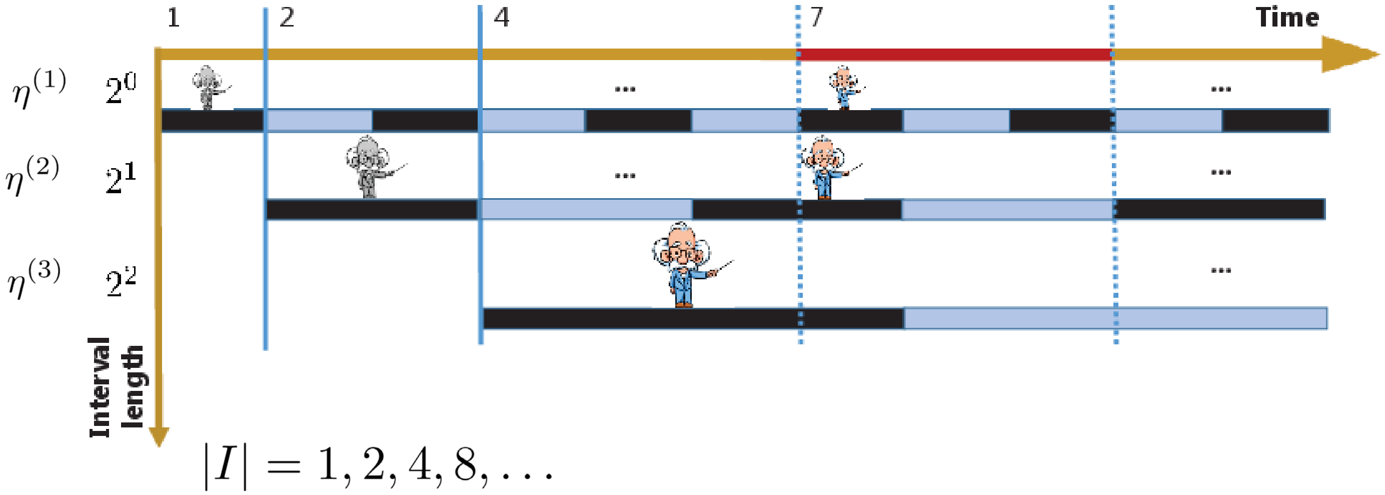
**Q.** What if the function changes over time?

- **Challenge**: Optimal stepsize depends on the dynamics – what if unknown?
- **Idea**: Combine weighted Raker learners with different step sizes

**AdaRaker** steps: A multiresolution design

**s1.** Add new Rakers at the beginning of intervals with progressively larger lengths

**s2.** $\hat{f}_t^{(I)}$: Raker active at interval *I*, with stepsize $\eta^{(I)} := \min\{1/2, \eta_0/\sqrt{|I|}\}$



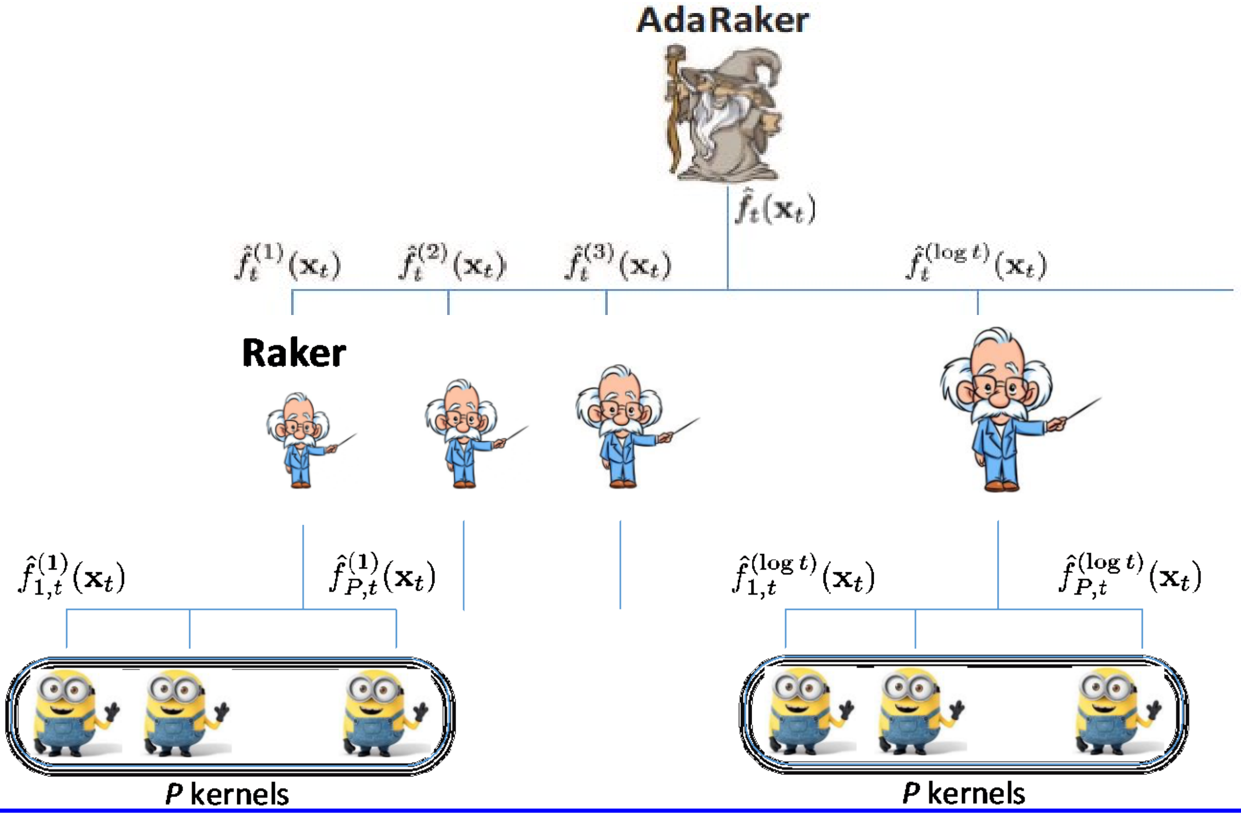
Y. Shen , T. Chen and G. B. Giannakis, "Random Feature-based Online Multi-kernel Learning in Environments with Unknown Dynamics," *Journal of Machine Learning Research*, to appear 2019.

# AdaRaker in action

**S1.** Obtain $\hat{f}_t^{(I)}(\mathbf{x}_t)$ from active Raker learners, and incur loss $\mathcal{L}_t(\hat{f}_t^{(I)}(\mathbf{x}_t))$

**S2.** Use relative loss $r_t^{(I)} := \mathcal{L}_t(\hat{f}_t(\mathbf{x}_t)) - \mathcal{L}_t(\hat{f}_t^{(I)}(\mathbf{x}_t))$ to update $\gamma_{t+1}^{(I)} = \gamma_t^{(I)} e^{-\eta^{(I)} r_t^{(I)}}$

**S3.** Update Raker learners $\{\hat{f}_{t+1}^{(I)}\}$ , to obtain $\hat{f}_{t+1}(\mathbf{x}_{t+1}) = \sum_{I=1}^{I_{\max}} \bar{\gamma}_{t+1}^{(I)} \hat{f}_{t+1}^{(I)}(\mathbf{x}_{t+1})$


AdaRaker
$\hat{f}_t(\mathbf{x}_t)$
$\hat{f}_t^{(1)}(\mathbf{x}_t)$
$\hat{f}_t^{(2)}(\mathbf{x}_t)$
$\hat{f}_t^{(3)}(\mathbf{x}_t)$
$\hat{f}_t^{(\log t)}(\mathbf{x}_t)$
Raker
$\hat{f}_{1,t}^{(1)}(\mathbf{x}_t)$
$\hat{f}_{P,t}^{(1)}(\mathbf{x}_t)$
$\hat{f}_{1,t}^{(\log t)}(\mathbf{x}_t)$
$\hat{f}_{P,t}^{(\log t)}(\mathbf{x}_t)$
P kernels
P kernels

# Performance analysis: Static regret

$$\mathrm{Reg}^{\mathrm{s}}_{\mathcal{A}}(T) := \sum_{t=1}^{T} \mathcal{L}_t(\hat{f}_t(\mathbf{x}_t)) - \min_{f \in \bigcup_{p=1}^{P} \mathcal{H}_p} \sum_{t=1}^{T} \mathcal{L}_t(f(\mathbf{x}_t))$$

- Online decisions benchmarked by best fixed strategy in hindsight
- Sublinear $\mathrm{Reg}_T = \mathbf{o}(T)$ implies algorithm $\mathcal{A}$ incurs no regret "on average”

**(a1)** Per slot loss $\mathcal{L}(\boldsymbol{\theta}^\top \mathbf{z}_{\mathbf{V}}(\mathbf{x}_t), y_t))$ is convex and bounded

**(a2)** Gradient $\nabla\mathcal{L}(\boldsymbol{\theta}^\top \mathbf{z}_{\mathbf{V}}(\mathbf{x}_t), y_t))$ is bounded

**(a3)** Kernels $\{\kappa_p\}_{p=1}^{P}$ are shift-invariant, and bounded

❑ Static regret of Raker

**Theorem 1**. Under (a1)-(a3), Raker attains $\mathrm{Reg}^{\mathrm{s}}_{\mathrm{Raker}}(T) = \mathcal{O}(\sqrt{T})$ w.h.p.

S. Shalev-Shwartz, “Online learning and online convex optimization,” *Foundations and Trends in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2011.

# Switching regret

- Best switching solution $\left\{\{\check{f}_t^*\}_{t=1}^T \in \bigcup_{p\in\mathcal{P}} \mathcal{H}_p \,\middle|\, \sum_{t=1}^T \mathbb{1}(\check{f}_t^* \neq \check{f}_{t-1}^*) \leq m\right\}$

$$\mathrm{Reg}_{\mathcal{A}}^{m}(T) := \sum_{t=1}^{T} \mathcal{L}_t(\hat{f}_t(\mathbf{x}_t)) - \sum_{t=1}^{T} \mathcal{L}_t(\check{f}_t^*(\mathbf{x}_t))$$

max. number of switches

- Switching regret of AdaRaker

**Theorem 2.** AdaRaker achieves $\mathrm{Reg}_{\mathrm{AdaRaker}}^{m}(T) \leq \mathcal{O}(\sqrt{Tm})$ w.h.p.

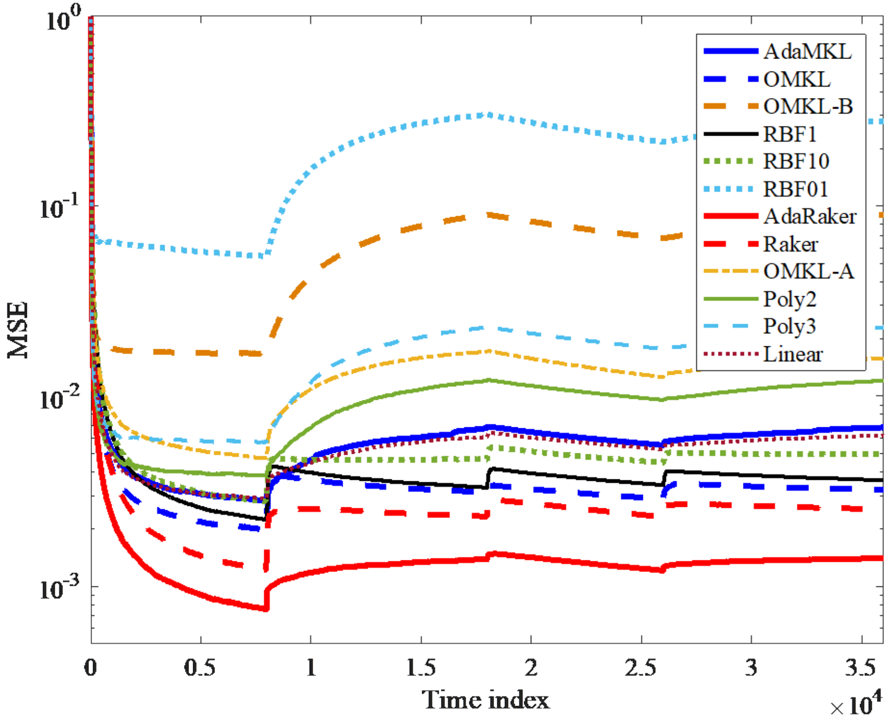
- If $m = \mathbf{o}(T) \Rightarrow \mathrm{Reg}_{\mathrm{AdaRaker}}^{m}(T) = \mathbf{o}(T)$

**Take home**: AdaRaker incurs on average no regret relative to the optimal switching solutions in unknown dynamics

Y. Shen , T. Chen and G. B. Giannakis, “Random Feature-based Online Multi-kernel Learning in Environments with Unknown Dynamics,” *Journal of Machine Learning Research*, to appear 2019.

# Synthetic test

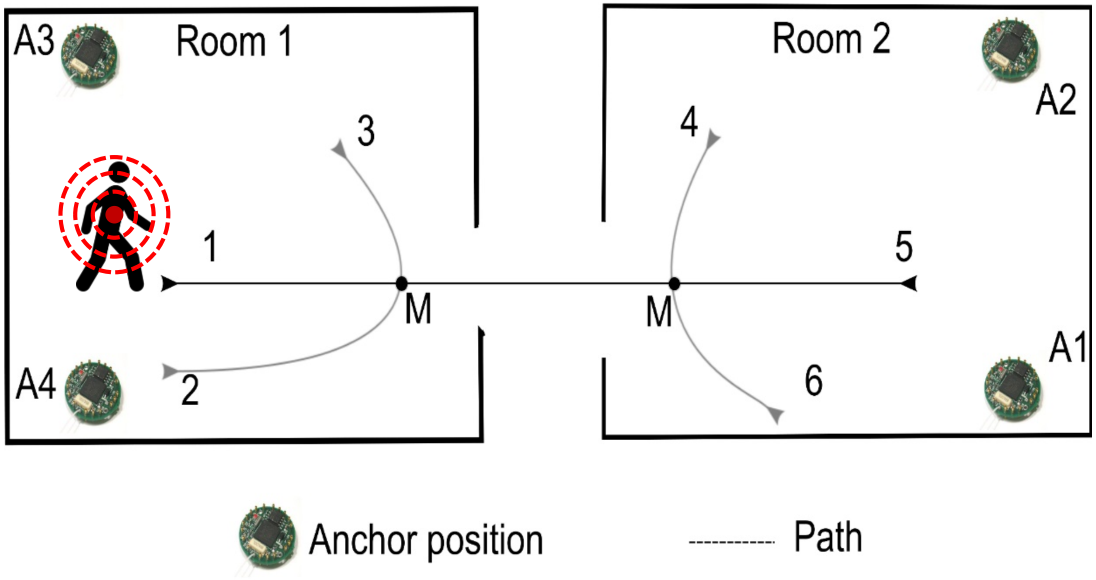
- Switching points: *t* = {8,000, 18,000, 26,000}
- RBF kernels with $\sigma^2 = \{0.1, 1, 10\}$, *B=D=50*

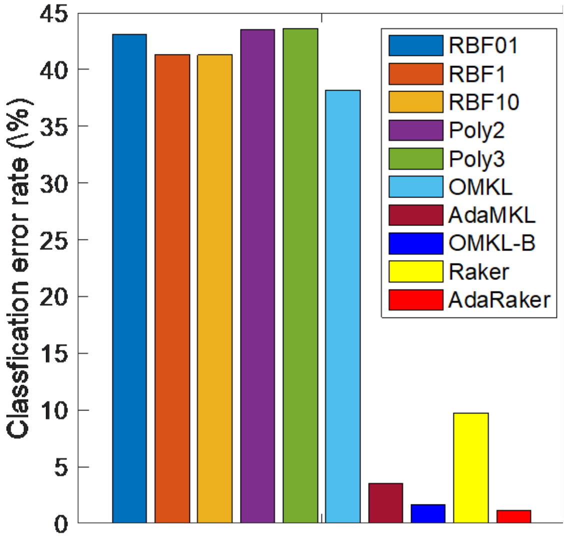


| | Runtime (sec) |
|---|---|
| AdaMKL | 318.52 |
| OMKL | 157.10 |
| RBF | 47.83 |
| Polynomial | 28. 27 |
| OMKL-B | 4.02 |
| **Raker** | **1.53** |
| AdaRaker | 24.2 |

- AdaRaker adapts fastest, Raker runs fastest

# In-home safety monitoring of elderly





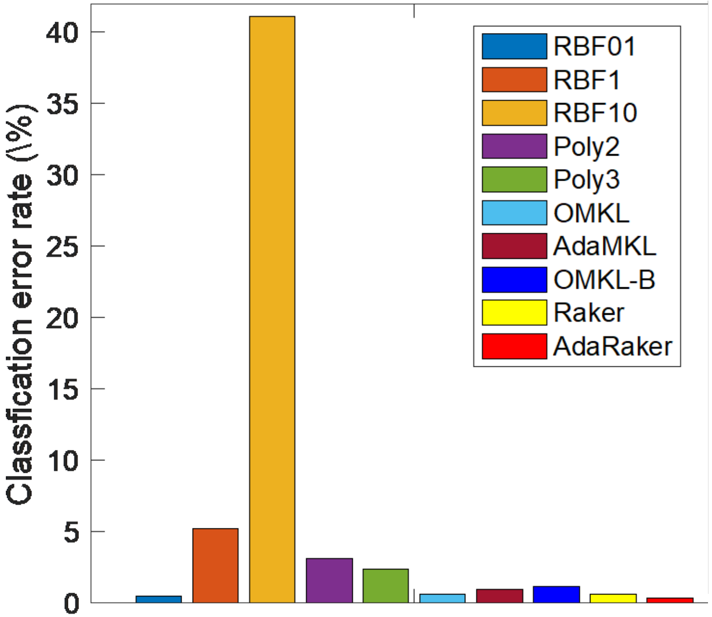
- □ $\mathbf{x}_t$: received signal strength (RSS) measurements from 4 anchor nodes

- □ $y_t$ : Does trajectory lead to a change of rooms?

Moshe Lichman. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

# Activity monitoring for health and fitness

sitting
standing
walking
running
climbing stairs

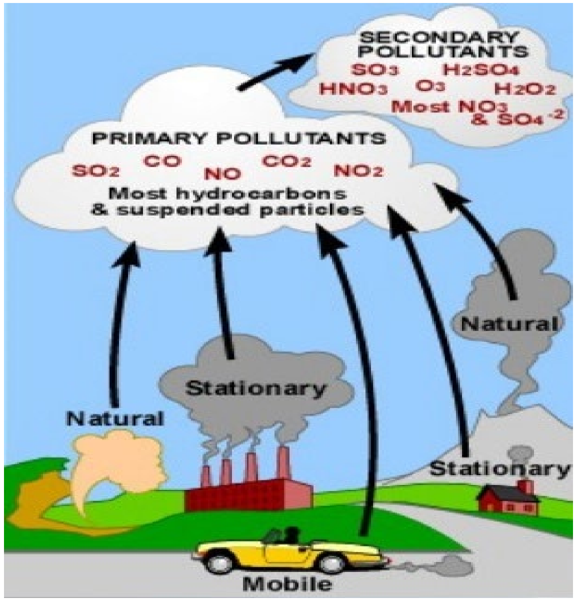


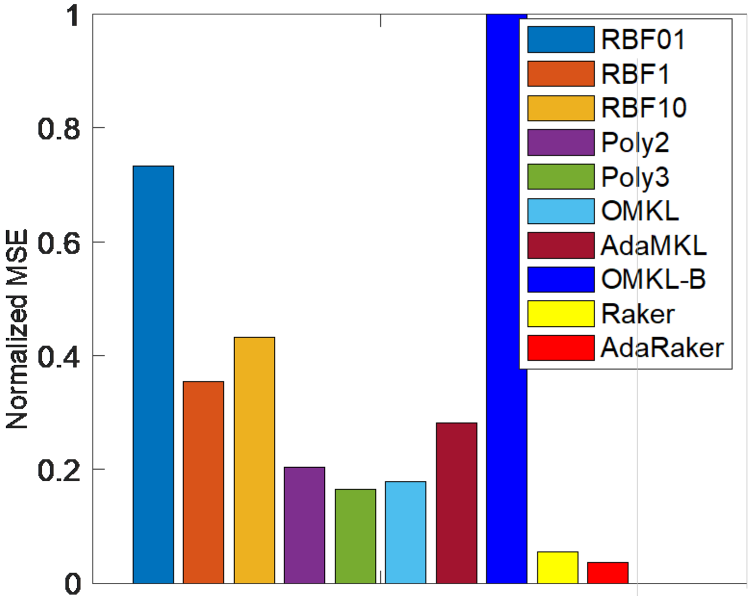
- $\mathbf{x}_t$ : triaxial acceleration and angular velocity

- $y_t$ : type of activity

Moshe Lichman. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

# Forecasting air pollution in smart cities





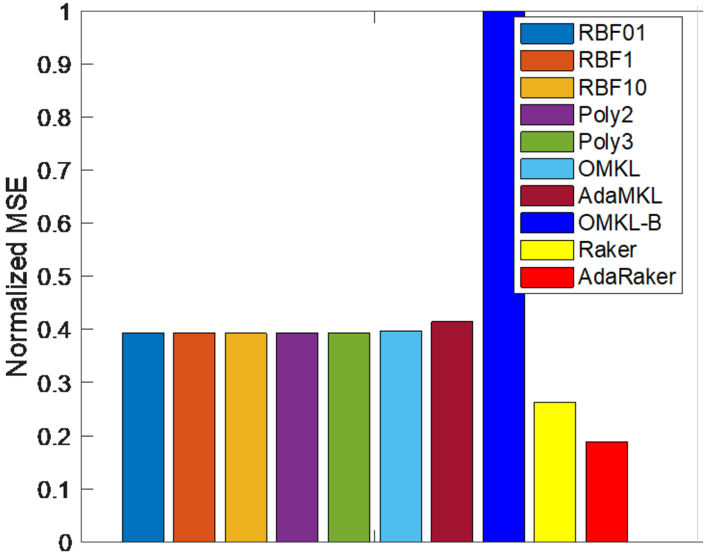
- ❑ $\mathbf{x}_t$: amount of different chemicals in the air
- ❑ $y_t$: amount of PM2.5 in the air

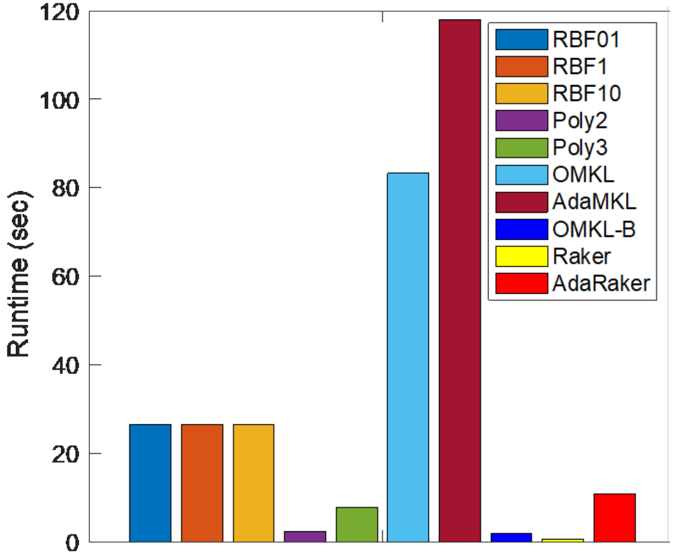
Moshe Lichman. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

# Energy consumption in smart homes

- $\mathbf{x}_t$: humidity and temperature
  outside and in different rooms

- $y_t$ : energy consumption





Moshe Lichman. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

# Contributions in context

- Batch function learning using kernels
  - Single kernel-based approach

    [Williams et al' 01], [Sheikholeslami et al' 17], [Rahimi-Recht' 07], [Felix et al' 16]
  - MKL approaches [Lanckriet et al' 04], [Bach' 08], [Cortes et al' 09], [Gonen-Alpaydin' 11]

- Online function learning using kernels
  - Budget-constrained approaches, e.g., [Kivinen et al' 04], [Dekel et al' 08]
  - RF-based single kernel learning [Lu et al'16], [Bouboulis et al'17]

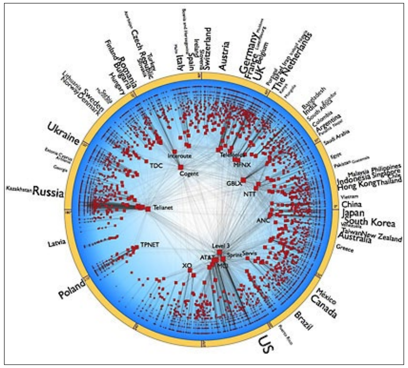
- **Our contributions**
  - Online **scalable** learning adaptive to **unknown dynamics** and **graphs**
  - **Data-driven** multi-kernel selection
  - Static and dynamic **regret bounds**

Y. Shen , T. Chen and G. B. Giannakis, "Random Feature-based Online Multi-kernel Learning in Environments with Unknown Dynamics," *Journal of Machine Learning Research*, to appear 2019.

# Learning over graphs

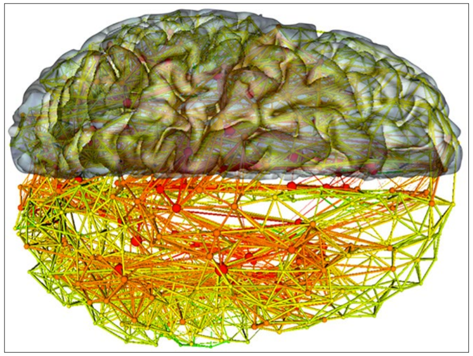
Social networks

Internet

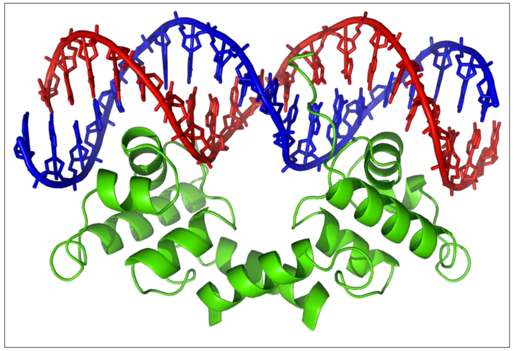


Energy grids

Financial markets



Brain networks



Gene/protein-regulatory nets

- ❑ **Challenges:** unavailable nodal attributes, privacy concern, growing networks
- ❑ **Desiderata:** Online graph- adaptive learning with **scalability** and **privacy**

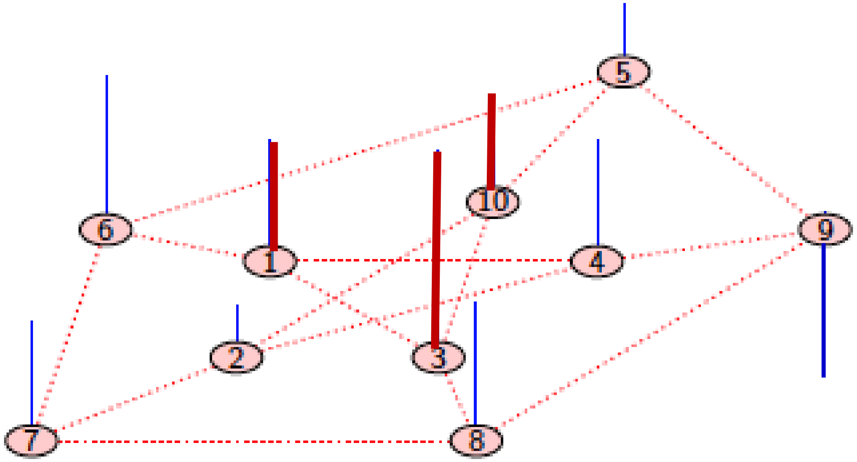
G. B. Giannakis, Y. Shen, and G. V. Karanikolas, "Topology Identification and Learning over Graphs: Accounting for Nonlinearities and Dynamics," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 787-807, May 2018.

# Learning graph signals

**Q1.** What if data are samples on vertices of a graph?

$$y_m = s_{v_m} + e_m \;, \quad m = 1, \ldots, M$$

- Adjacency matrix : $[\mathbf{A}]_{ij} \neq 0$ if $v_i$ is connected with $v_j$



**Goal.** Given adjacency matrix $\mathbf{A}$, and $\{y_m\}_{m=1}^{M}$, find $\{s_{v_n} = f(v_n)\}_{n=1}^{N}$ $M < N$

**Q2.** How are the graph signals related to the graph topology?

# Kernel-based learning over graphs

- Graph-induced RKHS $\mathcal{H}_{\mathcal{G}} := \{f | f(v) = \sum_{n=1}^{N} \alpha_n \kappa(v, v_n)\}$

$$\min_{f \in \mathcal{H}_{\mathcal{G}}} \frac{1}{M} \sum_{i=1}^{M} \mathcal{C}(f(v_i), y_i) + \lambda \Omega\left(\|f\|_{\mathcal{H}}^2\right)$$

  - Representer Thm. $\hat{f}(v) = \sum_{m=1}^{M} \alpha_m \kappa(v, v_m) := \boldsymbol{\alpha}^\top \mathbf{k}(v)$

$\mathbf{k}(v_i)$ : *i* th row of

$[\mathbf{K}]_{i,j} := \kappa(v_i, v_j)$

- Graph kernels : e.g. $\mathbf{K} = \mathbf{L}^\dagger$ , with Laplacian $\mathbf{L} := \mathrm{diag}(\mathbf{A1}) - \mathbf{A}$
  - Functions of $\mathbf{L}^\dagger$ can capture diffusion (DF) or bandlimited (BL) kernels
  - Rely on the entire **A**, and lead to complexity $\mathcal{O}(N^3)$

**Q3.** What if new nodes join? Scalability and adaptivity? Privacy concerns?

# RF-based learning over graphs

**Our idea**: treat *n*th column/row of adjacency ($\mathbf{a}_n$) as feature of node *n*

$$y_n = f(\mathbf{a}_n) + e_n$$

- ❑ MKL with RF-approximation

$$\hat{f}(v_n) = \hat{f}(\mathbf{a}_n) = \sum_{p=1}^{P} \bar{w}_p \hat{f}_p^{\mathrm{RF}}(\mathbf{a}_n)$$

$$\hat{f}_p^{\mathrm{RF}}(\mathbf{a}_n) = \sum_{m=1}^{M} \alpha_m \hat{\kappa}_p(\mathbf{a}_m, \mathbf{a}_n) := \boldsymbol{\theta}_p^\top \mathbf{z}_p(\mathbf{a}_n)$$

$$\mathbf{z}_{\mathbf{V}}(\mathbf{a}_n) := \frac{1}{\sqrt{D}} \left[\sin(\mathbf{v}_1^\top \mathbf{a}_n), \ldots, \sin(\mathbf{v}_D^\top \mathbf{a}_n), \cos(\mathbf{v}_1^\top \mathbf{a}_n), \ldots, \cos(\mathbf{v}_D^\top \mathbf{a}_n)\right]^\top$$

# Graph-adaptive Raker

- **GradRaker**: Acquire *N* x*1* adjacency vector $\mathbf{a}_t$ per slot *t* , and run

**S1.** Parameter update for each kernel-based learner

$$\boldsymbol{\theta}_{p,t+1} = \boldsymbol{\theta}_{p,t} - \eta \nabla \mathcal{L}_t(\boldsymbol{\theta}_{p,t}^\top \mathbf{z}_p(\mathbf{a}_t), y_t)$$

**S2.** Weight update

$$w_{p,t+1} = w_{p,t} e^{-\eta \mathcal{L}_t \left( \hat{f}_{p,t}^{\mathrm{RF}}(\mathbf{a}_t) \right)} \qquad \bar{w}_{p,t+1} = w_{p,t+1} / \sum_p w_{p,t+1}$$

**S3.** Function update

$$\hat{f}_{t+1}^{\mathrm{RF}}(\mathbf{a}_{t+1}) := \sum_{p=1}^{P} \bar{w}_{p,t+1} \hat{f}_{p,t+1}^{\mathrm{RF}}(\mathbf{a}_{t+1}) \qquad \hat{f}_{p,t+1}^{\mathrm{RF}}(\mathbf{a}_{t+1}) = \boldsymbol{\theta}_{p,t+1}^\top \mathbf{z}_p(\mathbf{a}_{t+1})$$

Y. Shen, G. Leus, and G. B. Giannakis, “Online Graph-Adaptive Learning with Scalability and Privacy,” *IEEE Transactions on Signal Processing*, 2019.

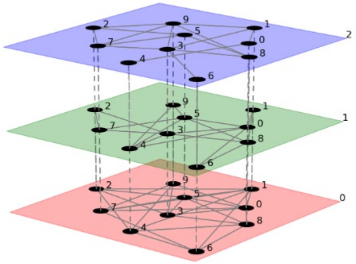
# Merits of GradRaker

- ❑ Sequential and scalable sampling and updates with theoretical guarantees
  - ➢ Sublinear regret

- ❑ Privacy-preserving scheme for each node with encrypted nodal information
  - ➢ $\mathbf{z}_{\mathbf{V}}(\mathbf{a}_n) := \frac{1}{\sqrt{D}} \left[\sin(\mathbf{v}_1^\top \mathbf{a}_n), \ldots, \sin(\mathbf{v}_D^\top \mathbf{a}_n), \cos(\mathbf{v}_1^\top \mathbf{a}_n), \ldots, \cos(\mathbf{v}_D^\top \mathbf{a}_n)\right]^\top$

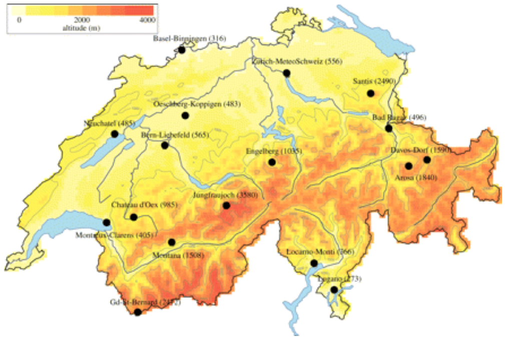
- ❑ Real-time prediction for newly joining nodes
  - ➢ $\hat{f}_p^{\mathrm{RF}}(v_{\mathrm{new}}) = \hat{\boldsymbol{\theta}}_p^\top \mathbf{z}_p(\mathbf{a}_{\mathrm{new}})$

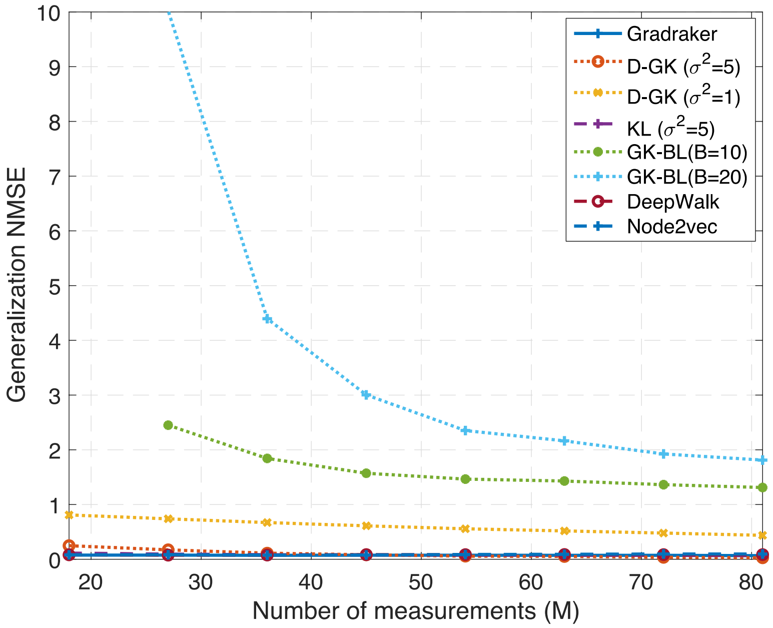
- ❑ Generalization to multi-layer networks or multi-hop neighbors
  - ➢ Adaptively combine layer-based learners

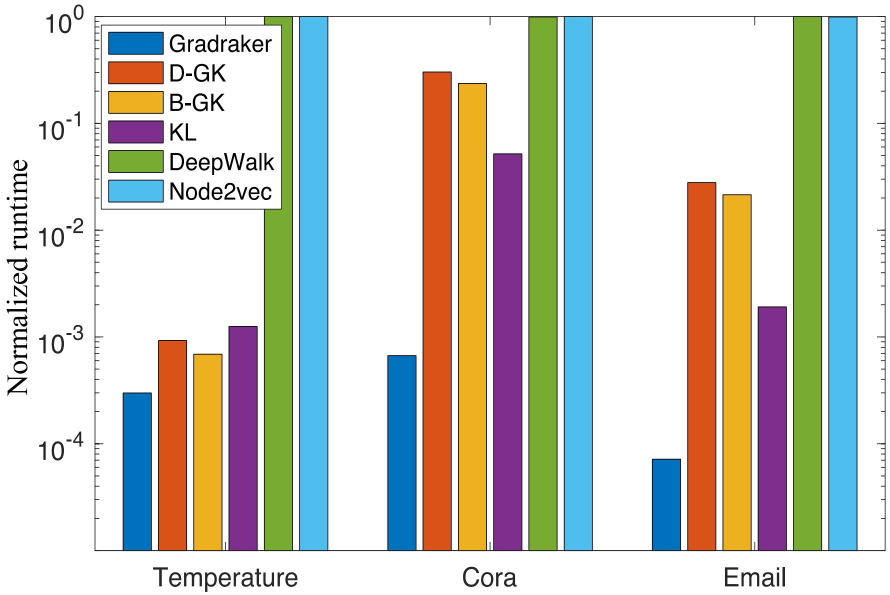


# Temperature forecasting

- Nodes: 89 measurement stations in Switzerland
- Edge weights obtained as in [Dong et al'14]
- Signals: temperatures between 1981 and 2010







Y. Shen, G. Leus, and G. B. Giannakis, "Online Graph-Adaptive Learning with Scalability and Privacy," *IEEE Transactions on Signal Processing*, 2019.

# Contributions in context

- ❑ Graph-kernel/filter based learning
  - ➢ Single kernel-based approach

    e.g., [Kondor et al 02], [Zhu et al 04], [Chen et al' 14] [Merkurjev et al' 16], [Segarra et al' 17]
  - ➢ MKL approaches [Romero et al' 17], [Ioannidis et al' 18]

- ❑ Graph based semi-supervised learning e.g., [Cortes et al' 06], [Berberidis et al' 18]

- ❑ Deep learning e.g., [Perozzi et al 14], [Kipf et al' 16], [Grover et al' 16]

- ❑ **Our contributions**
  - ➢ Sequential **scalable** function learning for **growing** networks
  - ➢ **Privacy-preserving** scheme based on encrypted nodal information
  - ➢ Analysis in terms of **regret bounds**

Y. Shen, G. Leus, and G. B. Giannakis, "Online Graph-Adaptive Learning with Scalability and Privacy," *IEEE Transactions on Signal Processing*, 2019.

# Conclusions

- **(Ada)Raker**
  - Adaptivity, scalability, and robustness to unknown dynamics
  - Sublinear regret relative to the best time-varying function approximant
- **GradRaker**
  - Sequential sampling and evaluation of nodal attributes
  - Adaptivity, scalability, privacy, and theoretical guarantee
- **Representative applications**
  - **Elderly safety monitoring**: Movement prediction, activity recognition
  - **Smart cities**: Air pollution, energy consumption, temperature prediction
  - **E-commerce, financial, social, and brain networks**

*Thank You!*

# Data science meets network science

- Scalable learning adaptive to (unknown) dynamics
  - Online subspace learning for **streaming** categorical data with **misses** [TSP17]
  - Online **function** learning adaptive to **unknown dynamics** [AISTATS18] [JMLR 19]

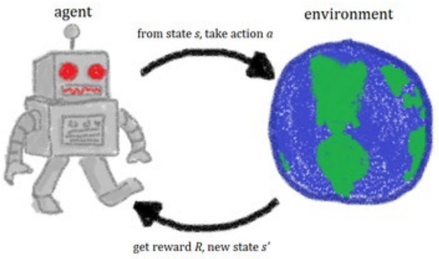
- Graph topology inference and tracking [PIEEE18]
  - **Data-driven** kernel based **nonlinear** topology inference [TSP17] [TSP18]
  - Tensor-based topology inference and **tracking** with **missing** observations [TSP17]

- Scalable learning adaptive to graphs
  - Graph-aware dimensionality reduction and learning [TSP17] [TSP18]
  - Privacy preserving graph-adaptive learning [TSP19]

# Outlook on algorithms and fundamental limits

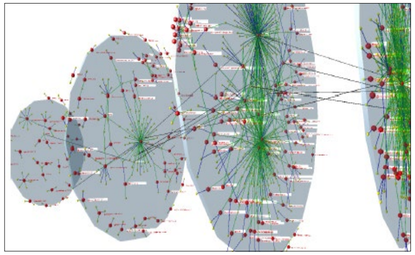
## Broadening the scope of function learning

- Reinforcement and deep learning
- (Non)parametric and semi-parametric learning
- Performance analysis


agent
environment
from state $s$, take action $a$
get reward $R$, new state $s'$

## Function learning over graphs

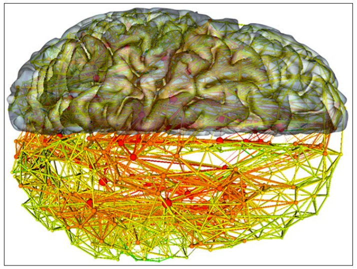
- Identifying time-varying topologies
- Adaptive sampling over graphs
- Scalable learning over growing networks
- Graph convolutional neural networks
- Performance and stability analysis



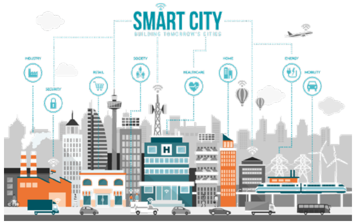
***Scalable, resilient, intelligent learning from big (network) data!***

# Outlook on “ML/DS+X”

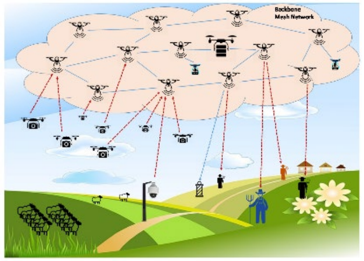
- **X = Biomedical engineering or Neuroscience**
  - Brain and gene regulatory networks
  - Medical imaging
  - Patient satisfaction evaluation



- **X = Smart cities**
  - Traffic, power, communication networks, IoT
  - Environmental data analytics



- **X = Multi-agent systems**
  - Robotics
  - Computer vision



- **X = Financial and Social networks**
  - Price discrimination
  - Recommender systems

*Thank You!*