Consistent Online Gaussian Process Regression Without the Sample Complexity Bottleneck

Alec Koppel
U.S. Army Research Laboratory

Statistical Learning
IEEE American Control Conference

July 11, 2019

Bayesian Methods

Supervised learning, map features to targets $\mathbf{x} \mapsto \hat{y} = f(\mathbf{x})$

$\Rightarrow$ found by minimizing loss $\ell(\hat{y}, y)$ averaged over data $(\mathbf{x}, y)$

$\rightarrow$ Bayesian methods ask: given $\{(\mathbf{x}_u, y_u)\})_{u<t}$, observe $\mathbf{x}_t$

$\Rightarrow$ how to form posterior distribution $\mathbb{P}(y_t \,\big|\, \{\mathbf{x}_u, y_u\}_{u<t} \cup \{\mathbf{x}_t\})$

$\rightarrow$ Needed for computing confidence intervals, quantiles, etc.

$\Rightarrow$ robustness/safety gaurentees, uncertainty-aware planning

$\Rightarrow$ foundation of climate forecasting, SLAM, robust MPC

Bayesian Methods

Can easily predict mean when dynamics are linear with AWGN

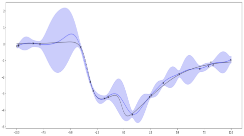
$\Rightarrow$ Kalman filter

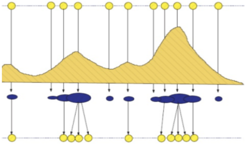
$\rightarrow$ In many modern applications, dynamics inherently nonlinear

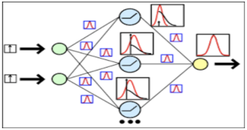
$\Rightarrow$ legged robotics, indoor localization, meterology

$\rightarrow$ How to estimate arbitrary posterior $\mathbb{P}(y \,\big|\, \{\mathbf{x}_u, y_u\}_{u \leq t} \cup \{\mathbf{x}_t\})$?

$\Rightarrow$ GPs, particle filters, “Bayesian deep networks”







Bayesian Methods

Can easily predict mean when dynamics are linear with AWGN

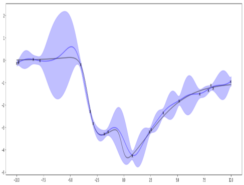
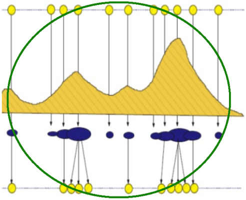
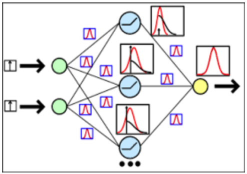
$\Rightarrow$ Kalman filter

$\rightarrow$ In many modern applications, dynamics inherently nonlinear

$\Rightarrow$ legged robotics, indoor localization, meterology

$\rightarrow$ How to estimate arbitrary posterior $\mathbb{P}(y \,\big|\, \{\mathbf{x}_u, y_u\}_{u \leq t} \cup \{\mathbf{x}_t\})$?

$\Rightarrow$ GPs, particle filters, “Bayesian deep networks”

Gaussian Processes

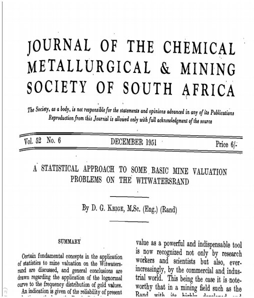
GPs $\Rightarrow$ nonparametric Bayesian method ($\mathcal{X} \subset \mathbb{R}^p$, $\mathcal{Y} \subset \mathbb{R}$)

$\Rightarrow$ $\hat{y} = f(\mathbf{x})$ $\Rightarrow$ capture relationship of $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$

$\Rightarrow$ estimate f via N training examples $\mathcal{S} = \{\mathbf{x}_n, y_n\}_{n=1}^N$.

$\rightarrow$ Unlike ERM, assume $f(\mathbf{x})$ follows parameterized distribution

$\Rightarrow$ then seek to estimate those parameters.


JOURNAL OF THE CHEMICAL METALLURGICAL & MINING SOCIETY OF SOUTH AFRICA

The Society, as a body, is not responsible for the statements and opinions advanced in any of its Publications
Reproduction from this Journal is allowed only with full acknowledgment of the source

Vol. 52 No. 6 DECEMBER 1951 Price 6/-

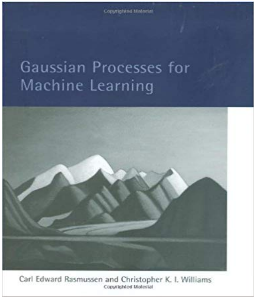
A STATISTICAL APPROACH TO SOME BASIC MINE VALUATION PROBLEMS ON THE WITWATERSRAND

By D. G. Krige, M.Sc. (Eng.) (Rand)

SUMMARY

Certain fundamental concepts in the application of statistics to mine valuation on the Witwatersrand are discussed, and general conclusions are drawn regarding the application of the lognormal curve to the frequency distribution of gold values. An indication is given of the reliability of present

value as a powerful and indispensable tool is now recognized not only by research workers and scientists but also, ever-increasingly, by the commercial and industrial world. This being the case it is noteworthy that in a mining field such as the



Gaussian Processes

GPs $\Rightarrow$ nonparametric Bayesian method ($\mathcal{X} \subset \mathbb{R}^p$, $\mathcal{Y} \subset \mathbb{R}$)

$\Rightarrow$ $\hat{y} = f(\mathbf{x})$ $\Rightarrow$ capture relationship of $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$

$\Rightarrow$ estimate f via N training examples $\mathcal{S} = \{\mathbf{x}_n, y_n\}_{n=1}^N$.

$\rightarrow$ Unlike ERM, assume $f(\mathbf{x})$ follows parameterized distribution

$\Rightarrow$ then seek to estimate those parameters.

$\rightarrow$ *Prior* on $\mathbf{f}_\mathcal{S} = [f(\mathbf{x}_n), \cdots, f(\mathbf{x}_N)]$ $\Rightarrow$ Gaussian: $\mathbf{f}_\mathcal{S} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_N)$

$\Rightarrow$ Covariance $\mathbf{K}_N = [\kappa(\mathbf{x}_m, \mathbf{x}_n)]_{m,n=1}^{N,N}$ via kernel $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$

$\Rightarrow$ Kernel $\Rightarrow$ prior about distance between points

$\Rightarrow$ e.g., Gaussian $\kappa(\mathbf{x}_m, \mathbf{x}_n) = \exp\{-\|\mathbf{x}_m - \mathbf{x}_n\|^2/c^2\}$

Gaussian Processes

GPs $\Rightarrow$ nonparametric Bayesian method ($\mathcal{X} \subset \mathbb{R}^p$, $\mathcal{Y} \subset \mathbb{R}$)

$\Rightarrow$ $\hat{y} = f(\mathbf{x})$ $\Rightarrow$ capture relationship of $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$

$\Rightarrow$ estimate f via N training examples $\mathcal{S} = \{\mathbf{x}_n, y_n\}_{n=1}^N$.

$\rightarrow$ Unlike ERM, assume $f(\mathbf{x})$ follows parameterized distribution

$\Rightarrow$ then seek to estimate those parameters.

$\rightarrow$ Standard GPs $\Rightarrow$ Gaussian noise corrupts $\mathbf{f}_\mathcal{S}$ to form obs.

$\rightarrow$ Observations have prior dist. $\mathbb{P}(\mathbf{y} \,|\, \mathbf{f}_\mathcal{S}) = \mathcal{N}(\mathbf{f}_\mathcal{S}, \sigma^2\mathbf{I})$

$\Rightarrow$ where σ^2 is some variance parameter.

$\rightarrow$ Integrate prior $\Rightarrow$ marginal prob. $\mathbb{P}(\mathbf{y} \,|\, \mathcal{S}) = \mathcal{N}(\mathbf{0}, \mathbf{K}_N + \sigma^2\mathbf{I})$

Gaussian Processes

GPs $\Rightarrow$ nonparametric Bayesian method ($\mathcal{X} \subset \mathbb{R}^p$, $\mathcal{Y} \subset \mathbb{R}$)

$\Rightarrow$ $\hat{y} = f(\mathbf{x})$ $\Rightarrow$ capture relationship of $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$

$\Rightarrow$ estimate f via N training examples $\mathcal{S} = \{\mathbf{x}_n, y_n\}_{n=1}^N$.

$\rightarrow$ Unlike ERM, assume $f(\mathbf{x})$ follows parameterized distribution

$\Rightarrow$ then seek to estimate those parameters.

$\rightarrow$ Upon receiving new sample $\mathbf{x}_{N+1}$, form posterior for $\hat{y}_{N+1}$ as

$$\mathbb{P}(y_{N+1} \,\big|\, \mathcal{S} \cup \mathbf{x}_{N+1}) = \mathcal{N}\Big(\boldsymbol{\mu}_{N+1\,|\,\mathcal{S}}, \boldsymbol{\Sigma}_{N+1\,|\,\mathcal{S}}\Big)$$

$\Rightarrow$ where the mean and covariance are given by

$$\boldsymbol{\mu}_{N+1\,|\,\mathcal{S}} = \mathbf{k}_{\mathcal{S}}(\mathbf{x}_{N+1})[\mathbf{K}_N + \sigma^2\mathbf{I}]^{-1}\mathbf{y}_N$$

$$\boldsymbol{\Sigma}_{N+1\,|\,\mathcal{S}} = \kappa(\mathbf{x}_{N+1}, \mathbf{x}_{N+1}) - \mathbf{k}_{\mathcal{S}}^T(\mathbf{x}_{N+1})[\mathbf{K}_N + \sigma^2\mathbf{I}]^{-1}\mathbf{k}_{\mathcal{S}}(\mathbf{x}_{N+1})$$

$\Rightarrow$ $\mathbf{k}_{\mathcal{S}}(\mathbf{x}) = [\kappa(\mathbf{x}_1, \mathbf{x}); \cdots \kappa(\mathbf{x}_N, \mathbf{x})]$ $\Rightarrow$ empirical kernel map

Curse of Dimensionality

Computing posterior mean requires:

$\Rightarrow$ computing empirical kernel map $\mathbf{k}_{\mathcal{S}}(\mathbf{x})$

$\Rightarrow$ inverting kernel matrix $\mathbf{K}_N$

$\rightarrow$ Complexity of former computation is $\mathcal{O}(N)$; the later is $\mathcal{O}(N^3)$

$\rightarrow$ In era of big data and streaming applications: $N \rightarrow \infty$

$\Rightarrow$ this causes GPs to require infinite complexity in the limit!

$\rightarrow$ Question: as $N \rightarrow \infty$, how to find close-to-optimal GP?

$\Rightarrow$ with finite memory that's flexible, problem-dependent

$\Rightarrow$ suitable for *online/streaming* settings

Some Context

Memory-reduced GPs $\Rightarrow$ two categories [Rasmussen Ch. 8]

$\Rightarrow$ greedy forward selection (Seeger, Csato & Opper, etc.)

$\Rightarrow$ variational approx. GP likelihood (Tsitsias, Snelson, etc.)

$\rightarrow$ Overarching theme: fix some memory budget M

$\Rightarrow$ "Project" likelihood of additional points onto "subspace"

$\Rightarrow$ since M unknown a priori, fixing it may cause divergence

$\rightarrow$ Goal: memory under control & approximate convergence

$\Rightarrow$ most existing approaches lack consistency guarantees

$\rightarrow$ Approach: compress current posterior w.r.t. metric

$\Rightarrow$ allows complexity to grow/shrink via data importance

Online Gaussian Processes

Define time-series of observations as $\mathcal{S}_t = \{\mathbf{x}_u, y_u\}_{u \le t}$,

$\Rightarrow$ Rewrite posterior in terms of $\mathcal{S}_t \cup \{\mathbf{x}_{t+1}\}$ as

$$\boldsymbol{\mu}_{t+1 \,|\, \mathcal{S}_t} = \mathbf{k}_{\mathcal{S}_t}(\mathbf{x}_{t+1})[\mathbf{K}_t + \sigma^2 \mathbf{I}]^{-1} \mathbf{y}_t$$

$$\boldsymbol{\Sigma}_{t+1 \,|\, \mathcal{S}_t} = \kappa(\mathbf{x}_{t+1}, \mathbf{x}_{t+1}) - \mathbf{k}_{\mathcal{S}_t}^T(\mathbf{x}_{t+1})[\mathbf{K}_t + \sigma^2 \mathbf{I}]^{-1} \mathbf{k}_{\mathcal{S}_t}(\mathbf{x}_{t+1}).$$

$\rightarrow$ Kernel dictionary $\mathbf{X}_t := [\mathbf{x}_1; \cdots; \mathbf{x}_t] \in \mathbb{R}^{p \times t}$

$\Rightarrow$ grows i.e. $\mathbf{X}_{t+1} = [\mathbf{X}_t; \mathbf{x}_{t+1}] \in \mathbb{R}^{p \times t}$, storing *full past* $\{\mathbf{x}_u\}_{u \le t}$.

$\Rightarrow$ Define no. of columns in dictionary as *model order* M_t.

$\Rightarrow$ GP posterior has model order $M_t = t$.

$\rightarrow$ Denote posterior of y_t as $\rho_t = \mathbb{P}(y_t \,|\, \mathcal{S}_{t-1} \cup \mathbf{x}_t)$

Online Gaussian Processes

Suppose posterior is defined by some kernel dict. $\mathbf{D} \in \mathbb{R}^{p \times M}$

$\Rightarrow$ Rather than $\mathbf{X}_t$ which stacks all past points

$\rightarrow$ Then the posterior parameters may be computed as

$$\boldsymbol{\mu}_{t+1 \mid \mathbf{D}} = \mathbf{k}_{\mathbf{D}}(\mathbf{x}_{t+1})[\mathbf{K}_{\mathbf{D},\mathbf{D}} + \sigma^2 \mathbf{I}]^{-1} \mathbf{y}_t$$

$$\boldsymbol{\Sigma}_{t+1 \mid \mathbf{D}} = \kappa(\mathbf{x}_{t+1}, \mathbf{x}_{t+1}) - \mathbf{k}_{\mathbf{D}}^T(\mathbf{x}_{t+1})[\mathbf{K}_{\mathbf{D},\mathbf{D}} + \sigma^2 \mathbf{I}]^{-1} \mathbf{k}_{\mathbf{D}}(\mathbf{x}_{t+1}).$$

$\rightarrow$ kernel matrix $\mathbf{K}_t \Rightarrow \mathbf{K}_{\mathbf{DD}}$; empirical kernel map $\mathbf{k}_{\mathcal{S}}(\cdot) \Rightarrow \mathbf{k}_{\mathbf{D}}(\cdot)$,

$\Rightarrow$ $[\mathbf{K}_{\mathbf{D},\mathbf{D}}]_{mn} = \kappa(\mathbf{d}_m, \mathbf{d}_n)$, $\mathbf{k}_{\mathbf{D}} = [\kappa(\mathbf{d}_1, \cdot); \cdots; \kappa(\mathbf{d}_M, \cdot)]$

$\Rightarrow$ dictionary pts. = subset of past obs. $\{\mathbf{d}_m\}_{m=1}^M \subset \{\mathbf{x}_u\}_{u \le t}$

Compressing the Posterior

Given dictionary $\mathbf{D}_t \in \mathbb{R}^{p\times(M_t)}$ at time t and obs. $\mathbf{x}_{t+1}$

$\Rightarrow$ Compute posterior distribution $\rho_{\mathbf{D}_t} := \mathcal{N}\left(\boldsymbol{\mu}_{t+1 \mid \mathbf{D}_t}, \boldsymbol{\Sigma}_{t+1 \mid \mathbf{D}_t}\right)$

$\rightarrow$ Compress by fixing error nbhd. at $\mathcal{N}\left(\boldsymbol{\mu}_{t+1 \mid \mathbf{D}_t}, \boldsymbol{\Sigma}_{t+1 \mid \mathbf{D}_t}\right)$

$\Rightarrow$ w.r.t. Hellinger metric: easily computable for Gaussians

$\rightarrow$ for distributions $\nu = \mathcal{N}(\boldsymbol{\mu}_1, \Sigma_1)$, $\lambda = \mathcal{N}(\boldsymbol{\mu}_2, \Sigma_2)$, given as

$$d_H(\nu, \lambda) = \sqrt{1 - \frac{|\Sigma_1|^{1/4}|\Sigma_2|^{1/4}}{|\bar{\Sigma}|}\exp\left\{-\frac{1}{8}(\mu_1 - \mu_2)\,\bar{\Sigma}^{-1}\,(\mu_1 - \mu_2)\right\}}$$

where $\bar{\Sigma} = (\Sigma_1 + \Sigma_2)/2$.

Compressing the Posterior

Given dictionary $\mathbf{D}_t \in \mathbb{R}^{p\times(M_t)}$ at time t and obs. $\mathbf{x}_{t+1}$

$\Rightarrow$ Compute posterior distribution $\rho_{\mathbf{D}_t} := \mathcal{N}(\boldsymbol{\mu}_{t+1 \mid \mathbf{D}_t}, \boldsymbol{\Sigma}_{t+1 \mid \mathbf{D}_t})$

$\rightarrow$ Compress by fixing error nbhd. at $\mathcal{N}(\boldsymbol{\mu}_{t+1 \mid \mathbf{D}_t}, \boldsymbol{\Sigma}_{t+1 \mid \mathbf{D}_t})$

$\Rightarrow$ w.r.t. Hellinger metric: easily computable for Gaussians

$\rightarrow$ Greedily prune w.r.t. Hellinger metric while inside nbhd.

$\Rightarrow$ Accomplished via destructive variant of matching pursuit

$\Rightarrow$ Customized to operate with the Hellinger distance

$$(\boldsymbol{\mu}_{\tilde{\mathbf{D}}_{t+1}}, \boldsymbol{\Sigma}_{\tilde{\mathbf{D}}_{t+1}}, \tilde{\mathbf{D}}_{t+1}) = \mathbf{DHMP}(\boldsymbol{\mu}_{t+1 \mid \mathbf{D}_t}, \boldsymbol{\Sigma}_{t+1 \mid \mathbf{D}_t}, \tilde{\mathbf{D}}_{t+1}, \epsilon_t)$$

$\rightarrow$ Then append latest point: $\mathbf{D}_{t+1} = [\tilde{\mathbf{D}}_{t+1}, \mathbf{x}_{t+1}]$

$\Rightarrow$ details of matching pursuit are messy, left to the paper

Parsimonious Online GPs (POG)

A geometric view

→ Learning update rule

$$\mu_{t+1\,|\,\mathbf{D}} = \mathbf{k}_{\mathbf{D}}(\mathbf{x}_{t+1})[\mathbf{K}_{\mathbf{D},\mathbf{D}} + \sigma^2 \mathbf{I}]^{-1}\mathbf{y}_t$$

$$\Sigma_{t+1\,|\,\mathbf{D}} = \kappa(\mathbf{x}_{t+1}, \mathbf{x}_{t+1}) - \mathbf{k}_{\mathbf{D}}(\mathbf{x}_{t+1})[\mathbf{K}_{\mathbf{D},\mathbf{D}} + \sigma^2 \mathbf{I}]^{-1}\mathbf{k}_{\mathbf{D}}(\mathbf{x}_{t+1})$$

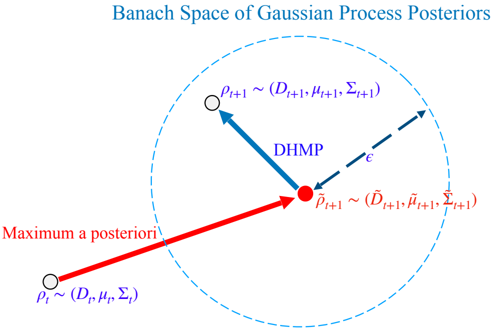
→ Compress w.r.t. *Hellinger* metric

⇒ causing ϵ error

⇒ add latest pt: $\tilde{\mathbf{D}}_{t+1} = [\mathbf{D}_t; \mathbf{x}_t]$

→ Linked to projected gradient

⇒ with hard-thresholding



Balancing Consistency and Memory

Theorem

POG attains the following posterior consistency results almost surely:

(i) *for decreasing budget* $\epsilon_t \to 0$, $\mathbb{P}_\Pi\{d_H(\rho_{\mathbf{D}_t}, \rho_{\mathbf{D}_{t-1}}) < \alpha \,|\, \mathcal{S}_t\} \to 1$

(ii) *for fixed budget* $\epsilon_t = \epsilon > 0$, $\mathbb{P}_\Pi\{d_H(\rho_{\mathbf{D}_t}, \rho_{\mathbf{D}_{t-1}}) < \gamma + \epsilon \,|\, \mathcal{S}_t\} \to 1$

For compression decreasing to null $\Rightarrow$ exact convergence

$\Rightarrow$ for constant compression budget, converge to nbhd.

Balancing Consistency and Memory

Theorem

Suppose POG is run with constant budget $\epsilon > 0$. Then the model order M_t of the posterior distributions $\rho_{\mathbf{D}_t}$ remains finite for all t, and the limiting distribution ρ_∞ has finite model complexity M^∞

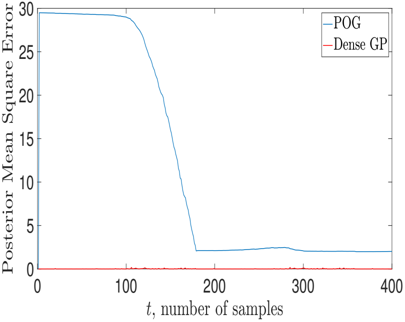
Merit of constant compression budget: provable finite memory

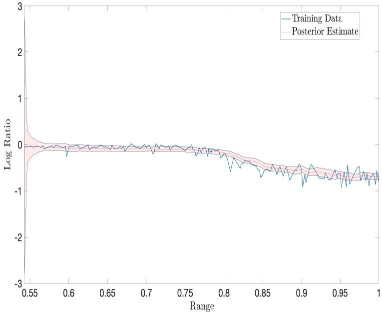
$\Rightarrow$ characterizing tradeoff of memory/consistency is difficult

$\Rightarrow$ depends on kernel hyperparameters, feature space radius

$\rightarrow$ Remaining open problem: how to establish this dependence

Real Data: LIDAR



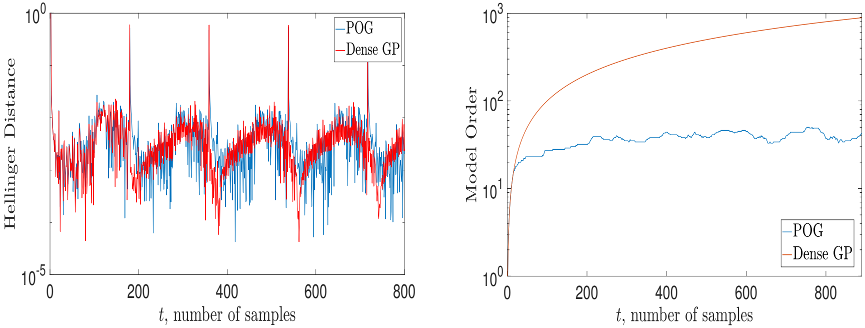


Posterior mean square error & actual interpolation

$\Rightarrow$ on LIDAR data set (nonlinear regression problem)

$\Rightarrow$ POG attains performance comparable to dense GP

Real Data: LIDAR



Evolution of Hellinger metric over time between sparse/dense GP

$\Rightarrow$ nearly identical

$\rightarrow$ but POG reduces complexity by orders of magnitude

$\Rightarrow$ by kicking out information extraneous to the posterior

$\Rightarrow$ has flexible complexity via convergence criterion

Conclusion

Gaussian processes $\Rightarrow$ often used in autonomy/robotics

$\Rightarrow$ curse of dimensionality: complexity $\approx$ sample size

$\Rightarrow$ a challenge common to nonparametric/Bayesian methods

$\rightarrow$ Precludes use in online settings

$\rightarrow$ Existing memory-reduction: proj. pts. to fixed size subspace

$\Rightarrow$ lack convergence guarantees, in contrast to POG

$\rightarrow$ POG trades off consistency and memory

$\rightarrow$ Experiments $\Rightarrow$ POG and dense GP exhibit similar behavior

$\rightarrow$ Future directions: GP bandits, safe low-latency MPC

References

$\Rightarrow$ A. Koppel, "Consistent Online Gaussian Process Regression Without the Sample Complexity Bottleneck," IEEE American Control Conference, July. 2019.

$\rightarrow$ A. Koppel, A.B. Singh, K. Rajawat, and B.M. Sadler, "Optimally Compressed Online Nonparametric Learning," in IEEE Signal Processing Magazine (submitted), 2019.