

A Dynamical Systems Perspective on Online Bayesian Nonparametric Estimators with Adaptive Hyperparameters

Alec Koppel, Amrit S. Bedi, and Vikram Krishnamurthy



Cornell University.

IEEE Int. Conf. on Acoustics, Speech, & Signal Process. (ICASSP),
Toronto, Ontario, Canada, June 6-11, 2021

Acknowledgment



Amrit Singh Bedi (ARL)



Vikram Krishnamurthy (Cornell University)

Supervised Learning

- ▶ Data $\mathcal{D} := \{\mathbf{x}_i, y_i\}_{i=1}^N$ drawn from some (unknown) dist. $\mathcal{P}(X, Y)$



- ▶ The goal is to learn a function $f(\mathbf{x}) = y$
- ▶ Select the appropriate function class for f say $f \in \mathcal{H}$
- ▶ Find the optimal f^* within the selected class \mathcal{H}
- ▶ **How to perform that?** — \rightarrow via minimizing a **loss function**

Supervised Learning

- ▶ $\ell : \mathcal{H} \times \mathcal{Y} \rightarrow \mathbb{R} \Rightarrow$ defines merit of statistical model

$$f^* = \operatorname{argmin}_{f \in \mathcal{H}} R(f) := \mathbb{E}_{\mathbf{x}, \mathbf{y}}[\ell(f(\mathbf{x}), y)] + \frac{\lambda}{2} \|f\|_{\mathcal{H}}^2$$

\Rightarrow Examples: Squared loss, absolute loss

- ▶ Applications:

\Rightarrow spam detection, image classification, speech recognition etc.



Spam detection

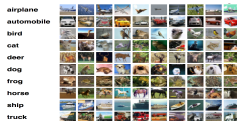


Image classification



Speech recognition

- ▶ **Remark:** \Rightarrow interested in streaming settings \Rightarrow sequential samples

On Hyperparameter Tuning

- ▶ Classical offline approaches: backtesting/cross-validation
- ▶ Modern online approaches formulated as
 - ⇒ Bayesian inference¹
 - ⇒ Multi-armed bandits²
- ▶ Bayesian inference
 - ⇒ requires likelihoods from a log-concave family
 - ⇒ beyond which it devolves into non-convex stochastic search³
- ▶ Multi-armed bandits
 - ⇒ Focuses only on the evolution of hyperparameters
 - ⇒ updates capturable by a black box reward⁴
- ▶ **Proposed: Evolving hyperparameters during training**
 - ⇒ originally as heuristic random search in genetic algorithms⁵

¹ T. D. Bui et al., "A unifying framework for GPs pseudo-point approx. using power expectation propagation," in JMLR, 2017

² G. Ghiasi et al., "Dropblock: A regularization method for convolutional networks," in Advances in Neural Information Processing Systems, 2018

³ D. M. Blei et al., "Variational inference: A review for statisticians," Journal of the American statistical Association, vol. 112, no. 518, 2017

⁴ A. S. Bedi et al., "Efficient gaussian process bandits by believing only informative actions," arXiv preprint arXiv:2003.10550, 2020

⁵ M. Mitchell, An introduction to genetic algorithms. MIT press, 1998.

Function class \mathcal{H}

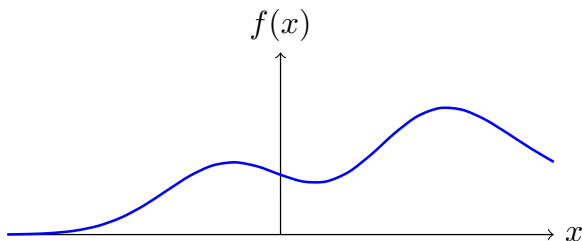
- ▶ \mathcal{H} is **Reproducing Kernel Hilbert Space (RKHS)**
- ▶ Equipped with a unique *kernel function*, $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, such that:

$$(i) f(\mathbf{x}) = \langle f, \kappa(\mathbf{x}, \cdot) \rangle_{\mathcal{H}} \quad \text{for all } \mathbf{x} \in \mathcal{X},$$

$$(ii) \mathcal{H} = \overline{\text{span}\{\kappa(\mathbf{x}, \cdot)\}} \quad \text{for all } \mathbf{x} \in \mathcal{X}.$$

- ▶ Property (i) \Rightarrow Will allow us to compute derivatives
- ▶ Kernel examples:

\Rightarrow Gaussian $\kappa(\mathbf{x}, \mathbf{x}') = \exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{2u^2}\right\}$, polynomial $(\mathbf{x}^T \mathbf{x}' + b)^c$



Function class \mathcal{H}

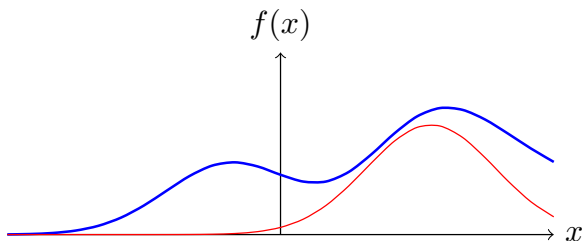
- ▶ \mathcal{H} is Reproducing Kernel Hilbert Space (RKHS)
- ▶ Equipped with a unique *kernel function*, $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, such that:

$$(i) f(\mathbf{x}) = \langle f, \kappa(\mathbf{x}, \cdot) \rangle_{\mathcal{H}} \quad \text{for all } \mathbf{x} \in \mathcal{X},$$

$$(ii) \mathcal{H} = \overline{\text{span}\{\kappa(\mathbf{x}, \cdot)\}} \quad \text{for all } \mathbf{x} \in \mathcal{X}.$$

- ▶ Property (i) \Rightarrow Will allow us to compute derivatives
- ▶ Kernel examples:

\Rightarrow Gaussian $\kappa(\mathbf{x}, \mathbf{x}') = \exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{2u^2}\right\}$, polynomial $(\mathbf{x}^T \mathbf{x}' + b)^c$



Function class \mathcal{H}

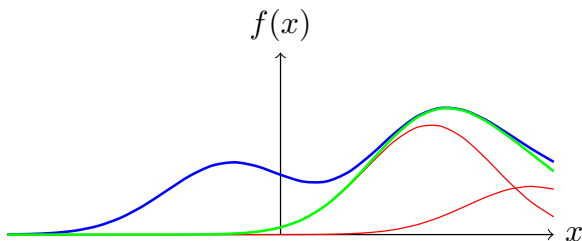
- ▶ \mathcal{H} is Reproducing Kernel Hilbert Space (RKHS)
- ▶ Equipped with a unique *kernel function*, $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, such that:

$$(i) f(\mathbf{x}) = \langle f, \kappa(\mathbf{x}, \cdot) \rangle_{\mathcal{H}} \quad \text{for all } \mathbf{x} \in \mathcal{X},$$

$$(ii) \mathcal{H} = \overline{\text{span}\{\kappa(\mathbf{x}, \cdot)\}} \quad \text{for all } \mathbf{x} \in \mathcal{X}.$$

- ▶ Property (i) \Rightarrow Will allow us to compute derivatives
- ▶ Kernel examples:

\Rightarrow Gaussian $\kappa(\mathbf{x}, \mathbf{x}') = \exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{2u^2}\right\}$, polynomial $(\mathbf{x}^T \mathbf{x}' + b)^c$



Function class \mathcal{H}

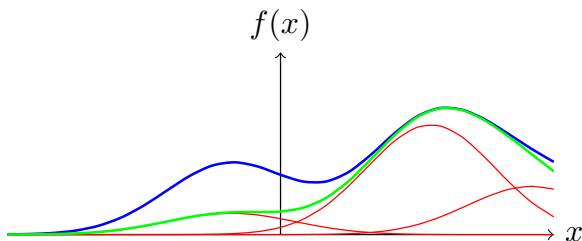
- ▶ \mathcal{H} is Reproducing Kernel Hilbert Space (RKHS)
- ▶ Equipped with a unique *kernel function*, $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, such that:

$$(i) f(\mathbf{x}) = \langle f, \kappa(\mathbf{x}, \cdot) \rangle_{\mathcal{H}} \quad \text{for all } \mathbf{x} \in \mathcal{X},$$

$$(ii) \mathcal{H} = \overline{\text{span}\{\kappa(\mathbf{x}, \cdot)\}} \quad \text{for all } \mathbf{x} \in \mathcal{X}.$$

- ▶ Property (i) \Rightarrow Will allow us to compute derivatives
- ▶ Kernel examples:

\Rightarrow Gaussian $\kappa(\mathbf{x}, \mathbf{x}') = \exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{2u^2}\right\}$, polynomial $(\mathbf{x}^T \mathbf{x}' + b)^c$



Function class \mathcal{H}

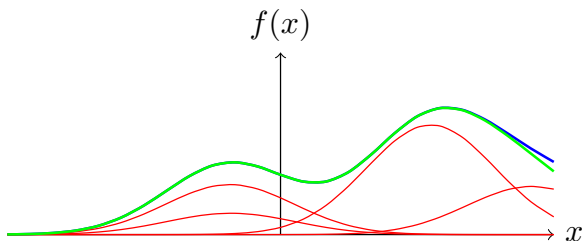
- ▶ \mathcal{H} is Reproducing Kernel Hilbert Space (RKHS)
- ▶ Equipped with a unique *kernel function*, $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, such that:

$$(i) f(\mathbf{x}) = \langle f, \kappa(\mathbf{x}, \cdot) \rangle_{\mathcal{H}} \quad \text{for all } \mathbf{x} \in \mathcal{X},$$

$$(ii) \mathcal{H} = \overline{\text{span}\{\kappa(\mathbf{x}, \cdot)\}} \quad \text{for all } \mathbf{x} \in \mathcal{X}.$$

- ▶ Property (i) \Rightarrow Will allow us to compute derivatives
- ▶ Kernel examples:

\Rightarrow Gaussian $\kappa(\mathbf{x}, \mathbf{x}') = \exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{2u^2}\right\}$, polynomial $(\mathbf{x}^T \mathbf{x}' + b)^c$



Function class \mathcal{H}

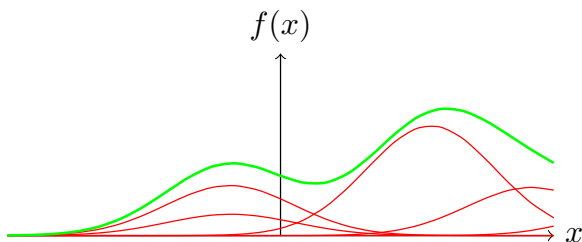
- ▶ \mathcal{H} is Reproducing Kernel Hilbert Space (RKHS)
- ▶ Equipped with a unique *kernel function*, $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, such that:

$$(i) f(\mathbf{x}) = \langle f, \kappa(\mathbf{x}, \cdot) \rangle_{\mathcal{H}} \quad \text{for all } \mathbf{x} \in \mathcal{X},$$

$$(ii) \mathcal{H} = \overline{\text{span}\{\kappa(\mathbf{x}, \cdot)\}} \quad \text{for all } \mathbf{x} \in \mathcal{X}.$$

- ▶ Property (i) \Rightarrow Will allow us to compute derivatives
- ▶ Kernel examples:

\Rightarrow Gaussian $\kappa(\mathbf{x}, \mathbf{x}') = \exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{2u^2}\right\}$, polynomial $(\mathbf{x}^T \mathbf{x}' + b)^c$



Train via Stochastic Gradient

- ▶ SGD on $R(f) := \mathbb{E}_{\mathbf{x}, \mathbf{y}}[\ell(f(\mathbf{x}), y)] + (\lambda/2)\|f\|_{\mathcal{H}}^2 \Rightarrow$ sample $(\mathbf{x}_k, \mathbf{y}_k)$:

$$\begin{aligned}f_{k+1} &= (1 - \eta_k \lambda) f_k - \eta_k \nabla_f \ell(f_k(\mathbf{x}_k), y_k) \\ &= (1 - \eta_k \lambda) f_k - \eta_k \ell'(f(\mathbf{x}_k), y_k) \kappa(\mathbf{x}_k, \cdot)\end{aligned}$$

- ▶ Newest feature vector \mathbf{x}_k enters kernel dictionary \mathbf{X}_k
 \Rightarrow with associated weight $\ell'(f(\mathbf{x}_k), y_k) := \partial \ell(f_k(\mathbf{x}_k), y_k) / \partial f_k(\mathbf{x}_k)$
- ▶ Representer Theorem $\Rightarrow f_k(\mathbf{x}) = \sum_{n=1}^{k-1} w_n \kappa(\mathbf{x}_n, \mathbf{x}) = \mathbf{w}_k^T \kappa_{\mathbf{X}_k}(\mathbf{x})$.
- ▶ SGD: parametric updates on weights and dictionary

$$\mathbf{X}_{k+1} = [\mathbf{X}_k, \mathbf{x}_k], \quad \mathbf{w}_{k+1} = [(1 - \eta_k \lambda) \mathbf{w}_k, -\eta_k \ell'(f_k(\mathbf{x}_k), y_k)]$$

Train via Stochastic Gradient

- ▶ SGD on $R(f) := \mathbb{E}_{\mathbf{x}, \mathbf{y}}[\ell(f(\mathbf{x}), y)] + (\lambda/2)\|f\|_{\mathcal{H}}^2 \Rightarrow$ sample $(\mathbf{x}_k, \mathbf{y}_k)$:

$$\begin{aligned}f_{k+1} &= (1 - \eta_k \lambda) f_k - \eta_k \nabla_f \ell(f_k(\mathbf{x}_k), y_k) \\ &= (1 - \eta_k \lambda) f_k - \eta_k \ell'(f(\mathbf{x}_k), y_k) \kappa(\mathbf{x}_k, \cdot)\end{aligned}$$

- ▶ Newest feature vector \mathbf{x}_k enters kernel dictionary \mathbf{X}_k
 \Rightarrow with associated weight $\ell'(f(\mathbf{x}_k), y_k) := \partial \ell(f_k(\mathbf{x}_k), y_k) / \partial f_k(\mathbf{x}_k)$
- ▶ Representer Theorem $\Rightarrow f_k(\mathbf{x}) = \sum_{n=1}^{k-1} w_n \kappa(\mathbf{x}_n, \mathbf{x}) = \mathbf{w}_k^T \kappa_{\mathbf{X}_k}(\mathbf{x})$.
- ▶ SGD: parametric updates on weights and dictionary

$$\mathbf{X}_{k+1} = [\mathbf{X}_k, \mathbf{x}_k], \quad \mathbf{w}_{k+1} = [(1 - \eta_k \lambda) \mathbf{w}_k, -\eta_k \ell'(f_k(\mathbf{x}_k), y_k)]$$

Train via Stochastic Gradient

- ▶ SGD on $R(f) := \mathbb{E}_{\mathbf{x}, \mathbf{y}}[\ell(f(\mathbf{x}), y)] + (\lambda/2)\|f\|_{\mathcal{H}}^2 \Rightarrow$ sample $(\mathbf{x}_k, \mathbf{y}_k)$:

$$\begin{aligned} f_{k+1} &= (1 - \eta_k \lambda) f_k - \eta_k \nabla_f \ell(f_k(\mathbf{x}_k), y_k) \\ &= (1 - \eta_k \lambda) f_k - \eta_k \ell'(f_k(\mathbf{x}_k), y_k) \underbrace{\kappa(\mathbf{x}_k, \cdot)} \end{aligned}$$

controlled by hyperparameters \mathbf{u}

- ▶ Newest feature vector \mathbf{x}_k enters kernel dictionary \mathbf{X}_k
 \Rightarrow with associated weight $\ell'(f_k(\mathbf{x}_k), y_k) := \partial \ell(f_k(\mathbf{x}_k), y_k) / \partial f_k(\mathbf{x}_k)$
- ▶ Representer Theorem $\Rightarrow f_k(\mathbf{x}) = \sum_{n=1}^{k-1} w_n \kappa(\mathbf{x}_n, \mathbf{x}) = \mathbf{w}_k^T \boldsymbol{\kappa}_{\mathbf{X}_k}(\mathbf{x})$.
- ▶ SGD: parametric updates on weights and dictionary

$$\mathbf{X}_{k+1} = [\mathbf{X}_k, \mathbf{x}_k], \quad \mathbf{w}_{k+1} = [(1 - \eta_k \lambda) \mathbf{w}_k, -\eta_k \ell'(f_k(\mathbf{x}_k), y_k)]$$

Approach

- ▶ $\ell : \mathcal{H} \times \mathcal{Y} \rightarrow \mathbb{R} \Rightarrow$ defines merit of statistical model

$$f^* = \operatorname{argmin}_{f \in \mathcal{H}} \mathbb{E}_{\mathbf{x}, y} [\ell(f(\mathbf{x}), y, \mathbf{u})] + \frac{\lambda}{2} \|f\|_{\mathcal{H}}^2$$

- \Rightarrow introduce parameters as **control variables** $\mathbf{u} \in \mathcal{U}$
 - \Rightarrow Evolve according to a distribution $\mathbb{P}(f(\mathbf{x}), y, \tilde{\mathbf{u}})$
 - \Rightarrow Let $r(f(\mathbf{x}), y, \tilde{\mathbf{u}})$ be a model fitness for $\tilde{\mathbf{u}}$
- ▶ Modified scheme \Rightarrow BARRETTE: **Bayesian Nonparametric Estimators with Adaptive Hyperparameters**

$$\mathbf{X}_{k+1} = [\mathbf{X}_k, \mathbf{x}_k], \quad \mathbf{w}_{k+1} = [(1 - \eta_k \lambda) \mathbf{w}_k, -\eta_k \ell'(f_k(\mathbf{x}_k), y_k, \mathbf{u}_k)]$$

- \Rightarrow Hyperpar. update $\mathbf{u}_{k+1} \sim \mathbb{P}(\bar{R}_{k+1})$ where long-run cost \bar{R}_{k+1} is


$$\bar{R}_{k+1} = \bar{R}_k + r(f_{k+1}(\mathbf{x}_k), y_k, \tilde{\mathbf{u}}_k)$$

Convergence Result

- ▶ Let us define the sequence $\phi_k = (f_k, \bar{R}_k)$, and $Z_k = (\mathbf{x}_k, y_k, \mathbf{u}_k)$
- ▶ Rewrite algorithm update succinctly as $\phi_{k+1} = \phi_k + \epsilon_k H(\phi_k, Z_k)$
- ▶ For $t \in [0, T]$, define interpolated process

$$\phi^\epsilon(t) = \phi_k, Z^\epsilon(t) = Z_k, \text{ for } t \in [k\epsilon, (k+1)\epsilon]$$

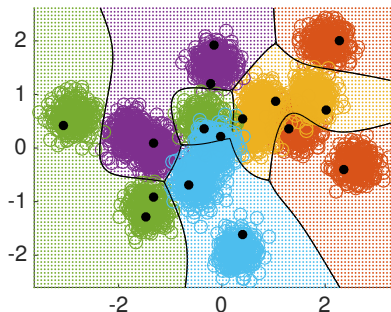
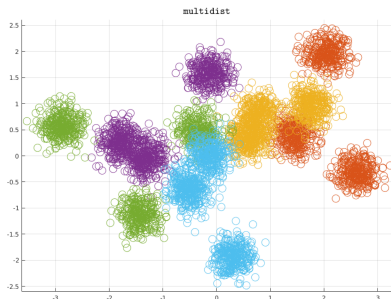
- ▶ **Theorem:** Under appropriate conditions, $\phi^\epsilon \rightarrow \phi$ weakly as $k \rightarrow \infty$, where ϕ is limiting functional that maps sample path to a real number.
- ▶ Limiting distribution is well defined equilibrium for $\min_{f \in \mathcal{H}} R(f)$
- ▶ $L(f)$ strongly cvx. \iff convergence in dist. to global optimizer
- ▶ **Related Work:**
 - \Rightarrow Convergence to SGD in Hilbert space⁶
 - \Rightarrow Weak convergence of tracking Markovian hyper-parameters⁷
- ▶ **This work:** study of their intertwined evolution

⁶H. J. Kushner et al., "Stoch. approx. in Hilbert space: Identification and opt. of linear continuous parameter sys.," SIAM J. Control Opt., 1985 

⁷M. Hamdi et al., "Tracking a markovmodulated stationary degree distribution of a dynamic random graph," IEEE Trans. Inf. Theory, 2014.

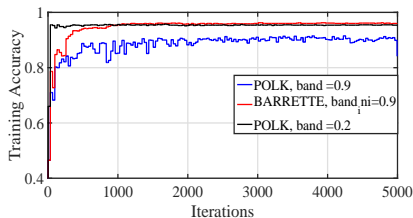
Experiments

- ▶ Case where training examples for a fixed class
⇒ drawn from a distinct Gaussian mixture
- ▶ 3 Gaussians per mixture, $C = 5$ classes total for this experiment
⇒ 15 total Gaussians generate data

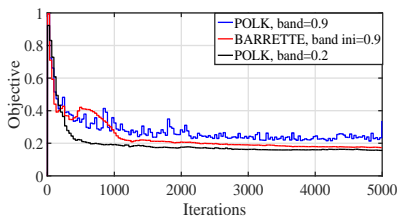


- ▶ **Grid colors** ⇒ decision, **bold black dots** ⇒ kernel dict. elements
- Zhu, Ji, and Trevor Hastie. "Kernel logistic regression and the import vector machine." NeurIPS, 2002.

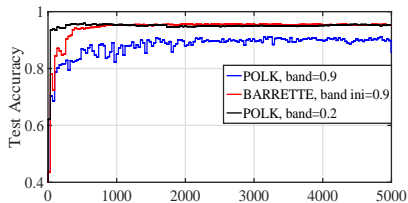
Experiments



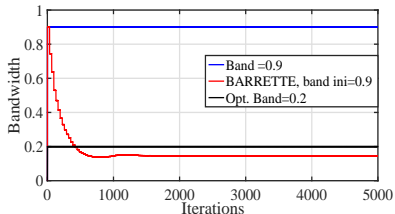
(a) Training accuracy



(b) Test loss



(c) Test accuracy



(d) Bandwidth

► We are able to *learn* the optimal bandwidth

Conclusion and Future Directions

- ▶ Proposed evolution of hyperparameters during training
- ▶ Established link to existing supervised learning in RKHS
- ▶ Established its global convergence in distribution
- ▶ Verified the approach via experiments
- ▶ Future Directions: form basis for neural architecture search
 - ⇒ hyperparameter tuning of convolutional kernels
 - ⇒ distributed algorithms with localized (“personalized”) models

Thank You