A DYNAMICAL SYSTEMS PERSPECTIVE ON ONLINE BAYESIAN NONPARAMETRIC ESTIMATORS WITH ADAPTIVE HYPERPARAMETERS

Alec Koppel[†]*, Amrit S. Bedi*[†]*, and Vikram Krishnamurthy*[*]

[†]CISD, U.S. Army Research Laboratory, Adelphi, Maryland, USA
[*]Department of Electrical and Computer Engineering, Cornell University, USA

ABSTRACT

This paper presents and analyzes constant step size stochastic gradient algorithms in reproducing kernel Hilbert Space (RKHS), which encapsulates various adaptive nonlinear interpolation schemes. The hyperparameters of the function iterates are updated via a distribution that depends on the estimates generated by the algorithm. Using stochastic averaging theory, we show that the estimates genertated by the algorithm converge weakly to an algebraically constrained ordinary differential equation. We illustrate this proposed algorithm in an online multi-class classification problem. Speficially, the proposed RKHS-valued stochastic gradient algorithm operating in concert with a Gaussian kernel whose bandwidth stably evolves during training, performs comparably to when the bandwidth is set according to oracle knowledge of its optimal value.

***Index Terms*—** Online learning, Bayesian learning, Stochastic optimization, averaging theory, algebraic-constrained ODE

1. INTRODUCTION

We consider convex expected-value minimization over an unknown distribution [1] where the feasible set is a reproducing kernel Hilbert Space (RKHS) [2]. Our main innovation is a variant of functional stochastic gradient method, where hyperparameters such as the kernel bandwidth [3] or basis points [4, 5] adapt *incrementally* during training. This introduces statistical dependence between parameters and hyperparameters, which under standard hypothesizes, via dynamical systems analysis of stochastic approximation [6], yields an algorithm that converges in distribution.

This paper considers stochastic optimization over a Hilbert space. What convex optimization over Euclidean space gains in terms of strength of conceptual guarantees is experimentally surpassed by nonlinear models across computer vision [7], natural language processing [8], and autonomous control [9]. We model nonlinearity using RKHS as doing so yields a convex program over a Hilbert space [10], gaining in representational power without the numerical challenges of non-convexity. Moreover, under suitable choice of kernel, they coincide with DNNs [11]. Unfortunately, the increased power of nonlinearity brings challenges of hyperparameter search. Specifically, hyperparameters significantly impact performance, and finding good selections in practice is nontrivial. Classical approaches such as cross validation [12] and grid/random search [13] fix them before training.

Evolving hyperparameters during training originally appeared as heuristic random search in genetic algorithms [14]. More modern approaches have instead considered formulations via Bayesian inference [15, 16] or multi-armed bandits (MAB) [17, 18]. The former typically necessitates prior/posterior conjugacy and likelihoods to be from a log-concave family, beyond which it devolves into nonconvex stochastic search [19]. The later focuses only on the evolution of hyperparameters, and treats the parametric updates as solely capturable by a black box reward [20, 21].

By contrast, here we propose to evolve both parameters and hyperparameters online *interdependently* via functional SGD in tandem with hyperparameter updates chosen via a distribution over the current iterate and samples (Sec. 3). We call this scheme **B**ayesian Nonp**ar**amet**r**ic **E**stimators wi**t**h Adap**t**ive Hyp**er**parameters, or BARRETTE. Notably, in Sec. 5, BARRETTE on a multi-class kernel SVM classification problem with a Gaussian kernel whose bandwidth evolves via a likelihood model [3] on a synthetic data [22] yields performance gains relative to fixed bandwidth approaches.

At a technical level, we emphasize that the stochastic gradient algorithm that we propose has an interesting structure: it converges weakly to an algebraically constrained differential equation for constant step-size selection (Theorem 1) under standard conditions (Assumption 1) in Sec. 4. Such dynamics are similar to that obtained in game-theoretic learning [23, 24]. This is quite different to the classical ODEs that arise from the stochastic averaging theory of gradient algorithms such as LMS that is widely studied in signal processing [25]. [6, 26].

2. PROBLEM DEFINITION

We consider the following stochastic optimization problem:

$$\theta^* = \operatorname*{argmin}_{\theta\in\Theta} L(\theta) := \operatorname*{argmin}_{\theta\in\Theta} \mathbb{E}_{\mathbf{x},y}[\ell(\theta(\mathbf{x}), y, \mathbf{u})] + \frac{\lambda}{2}\|\theta\|^2_{\mathcal{H}} \quad (1)$$

where $\lambda > 0$ is the regularization parameter. This corresponds to a nonlinear stochastic ridge regression or classification problem in Hilbert space; recall ridge regression adds a penalty l_2 norm squared term, namely $\frac{\lambda}{2}\|\theta\|^2_{\mathcal{H}}$ here, also called an Tikhonov regularizer. In (1), we interpret realizations $(\mathbf{x}_n, y_n)$ of the random pair $(\mathbf{x}, y)$ as training examples, i.e., feature vectors $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^p$ together with target variables $y_n \in \mathcal{Y}$ such as real values $\mathcal{Y} = \mathbb{R}$ or binary labels $\mathcal{Y} = \{0, 1\}$, in the respective cases of regression or classification, with $\mathbf{z} := (\mathbf{x}, y)$. Moreover, $\ell(\theta(\mathbf{x}), y, \mathbf{u})$ quantifies model fitness of estimator $\theta : \mathcal{X} \to \mathbb{R}$, which is small when $\theta(\mathbf{x})$ and y are close. Throughout, we assume that $L(\theta)$ is convex with respect to θ.

Moreover $\mathbf{u} \in \mathbb{R}^d$ is a vector of control variables, which may be interpreted as hyperparameters of a statistical model or decisions that influence which data is observed next. When $\mathbf{u}$ is fixed, then (1) reduces to the standard setting of supervised learning [27]. By contrast, here we focus on settings where we seek to *control* the choice of $\mathbf{u}$ when it belongs to a finite discrete set $\mathcal{U} = \{\mathbf{u}_{\min}, \ldots, \mathbf{u}_{\max}\}$ and it evolves according to a distribution

$\mathbb{P}(\theta, \mathbf{z}, \tilde{\mathbf{u}})$, i.e., $\mathbf{u} \sim \mathbb{P}(\theta, \mathbf{z}, \tilde{\mathbf{u}})$. In this way, $\{\mathbf{u}_t\}$ evolves as a Markov Chain over state space $\mathcal{U}$.

Included in this formulation is the case that $\mathbf{u} \sim \mathbb{P}(r(\theta(\mathbf{x}), y, \tilde{\mathbf{u}}))$ where $r(\theta(\mathbf{x}), y, \tilde{\mathbf{u}})$ is a model fitness criterion of $\theta(\mathbf{x})$ for previously chosen hyperparameters $\tilde{\mathbf{u}}$. Importantly, then, one may employ likelihood models that arise in bandwidth selection [28] or inducing inputs [4,5] for kernel/Gaussian Process regression, whose specific forms are deferred until later.

Reproducing Kernel Hilbert Space We now clarify the selection of function class Θ. We hypothesize Θ is a Hilbert space, denoted here as $\mathcal{H}$. We assume in this paper that $\mathcal{H}$ is a *separable* Hilbert space, i..e, $\mathcal{H}$ has a countable orthonormal basis. Elements of $\mathcal{H}$ are *functions*, $\theta : \mathcal{X} \to \mathcal{Y}$, that admit a representation in terms of elements of $\mathcal{X}$ when $\mathcal{H}$ has a special structure. In particular, equip $\mathcal{H}$ with a unique *kernel function*, $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, such that:

$$\begin{aligned} &(i)\ \langle \theta, \kappa(\mathbf{x}, \cdot)\rangle_{\mathcal{H}} = \theta(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \mathcal{X}\ , 5 \\ &(ii)\ \mathcal{H} = \overline{\text{span}\{\kappa(\mathbf{x}, \cdot)\}} \quad \text{for all } \mathbf{x} \in \mathcal{X}\ . \end{aligned} \tag{2}$$

where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ denotes the Hilbert inner product for $\mathcal{H}$. We further assume that the kernel is positive semidefinite, i.e. $\kappa(\mathbf{x}, \mathbf{x}') \geq 0$ for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$. This type of function space is called reproducing kernel Hilbert spaces (RKHS).

In (2), property (i) is called the reproducing property of the kernel, and is a consequence of the Riesz Representation Theorem [29]. Replacing θ by $\kappa(\mathbf{x}', \cdot)$ in (2) (i) yields the expression $\langle \kappa(\mathbf{x}', \cdot), \kappa(\mathbf{x}, \cdot)\rangle_{\mathcal{H}} = \kappa(\mathbf{x}, \mathbf{x}')$, which is the origin of the term "reproducing kernel." This property provides a practical means by which to access a nonlinear transformation of the input space $\mathcal{X}$. Specifically, denote by $\phi(\cdot)$ a nonlinear map of the feature space that assigns to each $\mathbf{x}$ the kernel function $\kappa(\cdot, \mathbf{x})$. Then the reproducing property of the kernel allows us to write the inner product of the image of distinct feature vectors $\mathbf{x}$ and $\mathbf{x}'$ under the map ϕ in terms of kernel evaluations only: $\langle \phi(\mathbf{x}), \phi(\mathbf{x}')\rangle_{\mathcal{H}} = \kappa(\mathbf{x}, \mathbf{x}')$. This is commonly referred to as the *kernel trick*, and yields an efficient way to estimate nonlinear functions.

Moreover, property (2) (ii) states that functions $\theta \in \mathcal{H}$ are given as a basis expansion over kernel evaluations. For the sample average approximations (SAA) of (1) with sample size N, the Representer Theorem [30, 31] yields that the optimal θ in the function class $\mathcal{H}$ is given as an expansion of kernel evaluated *only* training examples

$$\theta(\mathbf{x}) = \sum_{n=1}^{N} w_n \kappa(\mathbf{x}_n, \mathbf{x})\ . \tag{3}$$

where $\mathbf{w} = [w_1, \cdots, w_N]^T \in \mathbb{R}^N$ denotes a set of weights. The upper summand index N in (3) is henceforth referred to as the model order. Common choices κ include the polynomial kernel and the radial basis kernel (RBF), i.e., $\kappa(\mathbf{x}, \mathbf{x}') = \left(\mathbf{x}^T\mathbf{x}' + b\right)^c$ and $\kappa(\mathbf{x}, \mathbf{x}') = \exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{2d^2}\right\}$, respectively, where $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$. With the setting clarified, we shift to defining our algorithmic innovation in the following section.

3. STOCHASTIC GRADIENT ALGORITHM

We first present a stochastic gradient algorithm for the the kernelized λ-regularized expected risk minimization problem in (1) as

$$\theta_{k+1} = (1 - \eta_k\lambda)\theta_k - \eta_t \nabla_\theta \ell(\theta_k(\mathbf{x}_k), y_k, \mathbf{u}_k) \tag{4}$$

where $\eta_t > 0$ is an algorithm step-size either chosen as diminishing with $\mathcal{O}(1/t)$ or a small constant. We further require that, given $\lambda > 0$, the step-size satisfies $\eta_t < 1/\lambda$ and the sequence is initialized as $\theta_0 = 0 \in \mathcal{H}$. Given this initialization, by making use of the Representer Theorem (3), at step k, the function θ_k is given as an expansion in terms of feature vectors $\mathbf{x}_k$ observed thus far as

$$\theta_k(\mathbf{x}) = \sum_{n=1}^{k-1} w_n \kappa(\mathbf{x}_n, \mathbf{x}) = \mathbf{w}_k^T \boldsymbol{\kappa}_{\mathbf{X}_k}(\mathbf{x})\ . \tag{5}$$

On the right-hand side of (5) we have introduced the notation $\mathbf{X}_k = [\mathbf{x}_1, \ldots, \mathbf{x}_{k-1}] \in \mathbb{R}^{p\times(k-1)}$ and $\boldsymbol{\kappa}_{\mathbf{X}_k}(\cdot) = [\kappa(\mathbf{x}_1, \cdot), \ldots, \kappa(\mathbf{x}_{k-1}, \cdot)]^T$ Moreover, observe that the kernel expansion in (5), taken together with the functional update (4), yields the fact that performing the stochastic gradient method in $\mathcal{H}$ amounts to the following parametric updates on the kernel dictionary $\mathbf{X}$ and coefficient vector $\mathbf{w}$:

$$\begin{aligned} \mathbf{X}_{k+1} &= [\mathbf{X}_k,\ \ \mathbf{x}_k], \\ \mathbf{w}_{k+1} &= [(1 - \eta_k\lambda)\mathbf{w}_k,\ \ -\eta_k \ell'(\theta_k(\mathbf{x}_k), y_k, \mathbf{u}_k)]\ , \end{aligned} \tag{6}$$

Observe that this update causes $\mathbf{X}_{k+1}$ to have one more column than $\mathbf{X}_k$. We define the *model order* as number of data points M_t in the dictionary at time t (the number of columns of $\mathbf{X}_k$). FSGD is such that $M_k = t - 1$, and hence grows unbounded with iteration index k. We deal with this through a projection explained in more detail in Section 5, motivated by its ability to trade off memory and convergence [32].

Once the statistical model θ is updated based on the latest observations $(\mathbf{x}_k, y_t k$ given hyper-parameters $\mathbf{u}_k$ [cf. (4)], we then compute the long-run cost $\bar{R}_{k+1}$ of the choice $\mathbf{u}_t$ with respect to utility $r(\theta_{k+1}(\mathbf{x}_k), y_k, \mathbf{u}_k)$ as

$$\bar{R}_{k+1} = \bar{R}_k + r(\theta_{k+1}(\mathbf{x}_k), y_k, \mathbf{u}_k) \tag{7}$$

Then, the hyper-parameters $\mathbf{u}_k$ are updated according to a discrete distribution $\mathbb{P}$ that depends on long-run costs $\bar{R}_{k+1}$:

$$\mathbf{u}_{k+1} \sim \mathbb{P}(\bar{R}_{k+1}) \tag{8}$$

Remark 1 (7), (8) are quite different to classical stochastic approximation algorithms. The hyper-parameter process $\mathbf{u}$ is simulated from a Markovian transition kernel that depends on the estimate $\bar{R}$ from the algorithm; and the hyper-parameter feeds back into the algorithm. Such stochastic approximation algorithms are used in game-theoretic learning and yield interesting asymptotic dynamics. see [23, 24] for an exposition. Special cases exist only for gradient step-size selection in [33]

The state space for distribution $\mathbb{P}$ is $\mathcal{U} = \{\mathbf{u}_{\min}, \ldots, \mathbf{u}_{\max}\}$. In this way, the updates of $\mathbf{u}_t$ define a sample path through a Markov chain. The overall algorithm is the aggregation of (4), (7), and (8), summarized as Algorithm 1. Next, we present a motivating example.

Example 1 (**Bandwidth Adaptation**) An example of selecting hyperparameters according to (7) - (8) is, assuming an RBF kernel with bandwidth initialized as d_0, to repeatedly update it at time $t+1$ with a maximum likelihood step [3]

$$d_{t+1}^2 = \frac{1}{M_t}(M_t - 1)p \sum_n \frac{1}{f_t(\mathbf{x}_n)} \sum_{m\neq n} \|\mathbf{x}_n - \mathbf{x}_m\|^2 \kappa(\mathbf{x}_n, \mathbf{x}_m) \tag{9}$$

with probability $p \in [0, 1]$ that is a function of $\theta(\mathbf{x})$ and d_t. Of course, every choice of kernel comes with different hyperparameters, which give rise to different likelihood models that one may optimize. The emphasis on RBF kernels here is meant for illustration, as well as the fact that their use is akin to an uninformative prior about the geometric structure of the feature space $\mathcal{X}$, i.e., that it is locally flat. The bandwidth determines the degree of locality in this hypothesis.

Algorithm 1 **B**ayesian Nonp**ar**amet**r**ic **E**stimators wi**t**h Adap**t**ive Hyp**e**rparameters (BARRETTE)

Require: $\{\mathbf{x}_k, \mathbf{y}_k, \eta_k\}_{k=0,1,2,\ldots}$

initialize $\theta_0(\cdot) = 0, \mathbf{D}_0 = [], \mathbf{w}_0 = []$, i.e. initial dictionary, coefficient vectors are empty

for $t = 0, 1, 2, \ldots$ **do**

Obtain samples (possibly depending on past) $(\mathbf{x}_k, y_k)$

Compute functional stochastic gradient step

$$\theta_{k+1}(\cdot) = (1 - \eta_k\lambda)\theta_k - \eta_k \ell'(\theta_k(\mathbf{x}_k), \mathbf{y}_k, \mathbf{u}_k)\kappa(\mathbf{x}_k, \cdot)$$

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

Compute hyperparameters fitness $r(\theta_{k+1}(\mathbf{x}_k), y_k, \mathbf{u}_k)$

Update long-run cost $\bar{R}_t$ with step-size γ as :

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

Hyperparameters updated as $\mathbf{u}_{k+1} \sim \mathbb{P}(\bar{R}_{k+1})$

end for

We dedicate our numerical experiments to validating the proposed approach for Example 1. Next, we shift focus to theoretically characterizing Algorithm 1.

4. CONVERGENCE ANALYSIS

The aim of this section is to use the so-called ODE approach of [6] to analyze the convergence of Algorithm 1. To do so, begin by considering the alternative representation of the increments of Algorithm 1 as an abstract sequence ϕ_k in the RKHS $\mathcal{H}$ at discrete step k

$$\phi_k = (\theta_k, \bar{R}_k)\ , \quad Z_k = (\mathbf{x}_k, y_k, u_k)\ , \tag{10}$$

where we also define Z_k as the stacking of realizations of training examples and hyper-parameters $(\mathbf{x}_k, y_k, u_k)$. Further write the RKHS sequence of Algorithm 1 abstractly as

$$\phi_{k+1} = \phi_k + \epsilon_k H(\phi_k, Z_k) \tag{11}$$

Here we denote as $\epsilon_k > 0$ an increment step-size that subsumes the definition of η_k in (4) as well as how much the hyper-parameters $\mathbf{u}_k$ change across steps. Subsequently, we use the short-hand notation $H_k := H(\phi_k, Z_k)$.

We establish the asymptotic weak convergence of Algorithm 1 when the step size ϵ_k is a fixed constant. Weak convergence is a function space generalization of convergence in distribution of random variables defined next.

Definition 1 Consider a continuous time random process $X(t), t \in [0, T]$ which we will denote as X. A sequence of random processes $\{X^{(n)}\}$ (indexed by $n = 1, 2, \ldots$) converges weakly to X if for each bounded continuous real-valued functional ϕ,

$$\lim_{n\to\infty} \mathbb{E}\{\phi(X^{(n)})\} = \mathbb{E}\{\phi(X)\}.$$

Equivalently, a sequence of probability measures $\{P^{(n)}\}$ converges weakly to P if $\int \phi\, dP^{(n)} \to \int \phi\, dP$ as $n \to \infty$.

Note that the functional ϕ maps the entire trajectory of $X^{(n)}(t), 0 \leq t \leq T$ of the random process to a real number.

Let T denote a positive real number which denotes the finite time horizon. For $t \in [0, T]$, define the continuous-time piecewise constant interpolated process parametrized by the step-size $\epsilon > 0$ as

$$\phi^\epsilon(t) = \phi_k\ , \quad Z^\epsilon(t) = Z_k \quad \text{for } t \in [k\epsilon, (k+1)\epsilon) \tag{12}$$

where ϕ_k and Z_k are generated by the algorithm (10). Observe that $\phi^\epsilon(\cdot) \in D([0, \infty) : \mathcal{H})$, namely the space of functions defined on $[0, \infty)$ taking values in $\mathcal{H}$ equipped with kernel κ, such that the functions are right continuous and have left limits endowed with the Skorohod topology, and similarly for Z_k. Then, we impose the following condition.

Assumption 1 Let $\mathbb{E}_k$ denote the σ-algebra generated by $\{Z_l\}_{l<k}$. $\mathbb{E}_k\{H_k\}^{1+\Delta} < \infty$ for some $\Delta > 0$. Also $\{Z_k\}$ is a bounded stationary sequence and

$$\frac{1}{N}\sum_{k=l}^{N+l-1} \mathbb{E}_k\{H_k\} \to h(\phi) \text{ in probability as } N \to \infty \tag{13}$$

Assumption 1 is standard in stochastic approximation [6]. The condition $\mathbb{E}_k\{H_k\}^{1+\Delta} < \infty$ is sufficient for uniform integrability which is crucial for weak convergence. Also, (13) is a weak law of large numbers; the time-average $\frac{1}{N}\sum_{k=l}^{N+l-1} \mathbb{E}_k\{H_k\}$ converges in probability. Observe that Assumption 1 permits us to work with correlated sequences [6], and hence subsumes the examples of i.i.d. realizations from a time-invariant distribution whose variance is bounded [1], martingale difference sequences with finite second moments, moving average processes driving by a martingale difference sequence, as well as mixing sequences in which remote past and distant future are independent.

Next we define key quantities related to the limiting dynamics of (12) as $k \to \infty$. Specifically, denote the gradient of the regularized objective as

$$h(\theta, \mathbf{y}, \mathbf{x}, u) = \left[\ell'(\theta(\mathbf{x}), \mathbf{y}, u)\, \kappa(\mathbf{x}, \mathbf{x})\ .\ + \lambda\, \theta(\mathbf{x})\right] \tag{14}$$

Further define the algebraically constrained ODE as

$$\begin{aligned}
\frac{d\theta}{dt} &= -\int_{\mathcal{X}\times\mathcal{Y}} \sum_{i=1}^{m} h(\theta(\mathbf{x}), \mathbf{y}, \mathbf{x}, i)\, \pi_\theta(d\mathbf{x}, d\mathbf{y}, i) \\
\frac{d\bar{R}}{dt} &= \int_{\mathcal{X}\times\mathcal{Y}} \sum_{i=1}^{m} r(\theta(\mathbf{x}), \mathbf{y}, i)\, \pi_\theta(d\mathbf{x}, d\mathbf{y}, i) - \bar{R} \\
\pi_\theta(j) &= \sum_{i=1}^{m} P_{\bar{R}}(j|i)\, \pi_\theta(i)
\end{aligned} \tag{15}$$

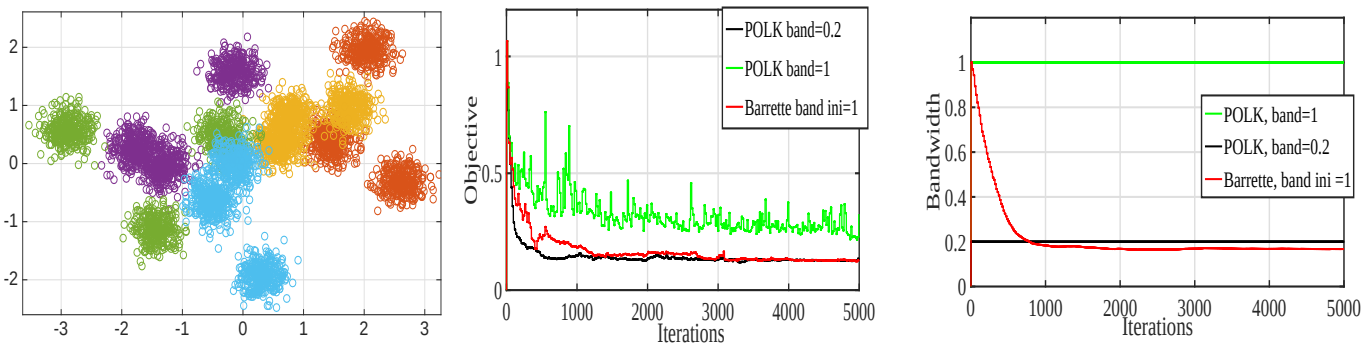
where $\pi_\theta(d\mathbf{x}, d\mathbf{y}, i)$ is the stationary distribution of the Hilbert-space-valued Markov process $(\mathbf{x}_k, y_k, u_k)$. Then we may express (15) as the following ODE system

$$\frac{d\phi}{dt} = h(\phi), \quad h(\phi) = \int H(\phi, Z)\, \pi_\phi(dZ)$$

With the dynamical systems (15) associated with Algorithm 1 defined in terms of the succinct abstract sequence (10) - (11), we are ready to state our main convergence result.

Theorem 1 Consider algorithm (11) with fixed step size ϵ. Suppose that Assumption 1 holds and the system (15) has a unique solution for each initial condition $\phi(0) = \phi_0$ in which the uniqueness is in the sense of distribution. Then the interpolated process ϕ^ϵ converges weakly to ϕ as $k \to \infty$ such that the limit satisfies (15).

Theorem 1, whose proof is deferred to an upcoming journal submission of this work, establishes that the interpolated process ϕ^ϵ defined by Algorithm 1 converges weakly, i.e., in distribution, to a ϕ for each initial condition, as $k \to \infty$. This means that the limiting distribution induced by Algorithm 1 is a well-defined equilibrium for



(a) Training Data

(b) Objective vs. samples processed

(c) Bandwidth vs. samples processed

Fig. 1: Toy data (Fig. 1a) is used to conduct experiments on multi-class kernel support vector machine. BARRETTE (Algorithm 1), by evolving its bandwidth during training, converges to a selection comparable to POLK initialized with oracle knowledge of the optimal bandwidth (Fig. 1c). By contrast, with a hyperparameter mis-specification, POLK yields a fixed bandwidth over the course of training which yields performance degradation. By contrast, BARRETTE's evolving bandwidth is much more effectively able to minimize the multi-class kernel logistic regression training objective despite a poor initial specification of its hyperparameter, as may be gleaned from Fig 1b.

the problem of minimizing θ over the RKHS (1) according to FSGD (4) while its hyperparameters evolve via (8). Note, owing to strong convexity, that equilibria in continuous time exactly correspond to global optimizers of (1). We note that the convergence of stochastic gradient algorithms on Hilbert spaces is studied in [34] and more recently weak convergence for tracking Markovian hyper-parameters on Hilbert spaces is analyzed in [35]. The study of their intertwined evolution is the unique aspect of this work.

5. EXPERIMENTS

We consider the problem of training a multi-class kernel support vector machine (Multi-KSVM). : the target domain $\mathcal{Y} = \{1, \ldots, C\}$ is a set of classes, and the goal is to maximize the class specific classification margin. Specifically, define a class specific activation function $f_c; \mathcal{X} \to \mathbb{R}$ and define them jointly as $\mathbf{f} \in \mathcal{H}^C$. In Multi-KSVM, the classifier is trained by consider the instantaneous loss function to be the hinge loss defined as

$$\ell(\mathbf{f}, \mathbf{x}_n, y_n) = \max(0, 1 + f_r(\mathbf{x}_n) - f_{y_n}(\mathbf{x})) + \lambda \sum_{c'=1}^{C} \|f_{c'}\|_{\mathcal{H}}^2,$$

where $r = \arg\max_{c' \neq y_n}(\mathbf{x}_n)$ for the given data sample $(\mathbf{x}_n, y_n)$. Further details about Multi-KSVM are provide in [36].

For the experiments, in a manner similar to [22], we generate the `multidist` data set using a set of Gaussian mixture models. The data set consists $N = 5000$ feature-label pairs for training and 2500 for testing. Each label y_n was drawn uniformly at random from the label set. The corresponding feature vector $\mathbf{x}_n \in \mathbb{R}^p$ was then drawn from a planar ($p = 2$), equitably-weighted Gaussian mixture model, i.e., $\mathbf{x} \,|\, y \;\sim\; (1/3)\sum_{j=1}^{3} \mathcal{N}(\boldsymbol{\mu}_{y,j}, \sigma_{y,j}^2 \mathbf{I})$ where $\sigma_{y,j}^2 = 0.2$ for all values of y and j. The means $\boldsymbol{\mu}_{y,j}$ are themselves realizations of their own Gaussian distribution with class-dependent parameters, i.e., $\boldsymbol{\mu}_{y,j} \sim \mathcal{N}(\boldsymbol{\theta}_y, \sigma_y^2 \mathbf{I})$, where $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_C\}$ are equitably spaced around the unit circle, one for each class label, and $\sigma_y^2 = 1.0$. We fix the number of classes $C = 5$, meaning that the feature distribution has, in total, 15 distinct modes. The data points are plotted in Figure 1a.

We evaluate Algorithm 1 to learn Multi-KSVM for Gaussian kernel with initialized bandwidth as 1, and evolves according to (9) (Example 1) with the distribution in (8) chosen as a Bernoulli based upon the sign of the gradient with parameter $p = .9$. In this way, the evolution of the bandwidth belongs to a Markov Chain evolving over this discrete set. We compare this with FSGD with constant bandwidths $\sigma \in \{0.2, 1\}$. Note that since we generated the data in Fig. 1a, we have oracle knowledge that the clusters have variance 0.2, and hence this is roughly the optimal selection for this problem instance. Both BARRETTE and FSGD employ the complexity-reducing projections based upon matching pursuit with budget parameter $\epsilon = K\eta^{3/2}$ with parsimony constant $K = 0.04$, which for FSGD is referred to as POLK, short for Parsimonious Online Learning with Kernels [32]. Both algorithms are run with constant step-size $\eta = 6$, mini-batch size 16, and regularization $\lambda = 10^{-6}$. For BARRETTE, we have initialized the bandwidth parameter to 1.

In Fig. 1c, we observe that BARRETTE, by evolving its bandwidth during training, is able to converge to a selection comparable to POLK initialized with oracle knowledge of the optimal bandwidth. By contrast, with a hyperparameter mis-specification, POLK yields a fixed bandwidth over the course of training which yields something to be desired in terms of performance. By contrast, the hyperparameter evolution employed by BARRETTE is much more effectively able to minimize the multi-class kernel hinge training loss despite a poor initialization, as may be gleaned from Fig. 1b.

6. CONCLUSION

We considered algorithms to adapt the hyperparameters of RKHS function iterates via a distribution that depends on the current function as well as training examples, which subsumes numerous online hyperparameter adaptation strategies that arise in practice, such as bandwidth and inducing input search. We encapsulated its limiting behavior as an ordinary differential equation with algebraic constraints, which allowed us to establish its weak convergence under constant step-size selection to the unique equilibrium of this problem. This approach was then used to develop online multi-class kernel classification algorithms with bandwidths that stably evolve during training, which yielded performance comparable to that which is achievable with choice of bandwidth according to oracle knowledge of its optimal value.

7. REFERENCES

[1] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on stochastic programming: modeling and theory*. SIAM, 2014.

[2] A. Berlinet and C. Thomas-Agnan, *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

[3] J. M. L. Murillo and A. A. Rodríguez, "Algorithms for gaussian bandwidth selection in kernel density estimators," in *Advances in Neural Information Processing Systems*, 2008.

[4] E. Snelson and Z. Ghahramani, "Sparse gaussian processes using pseudo-inputs," in *Advances in neural information processing systems*, 2006, pp. 1257–1264.

[5] M. Titsias, "Variational learning of inducing variables in sparse gaussian processes," in *Artificial Intelligence and Statistics*, 2009, pp. 567–574.

[6] H. J. Kushner and G. Yin, *Stochastic Approximation Algorithms and Recursive Algorithms and Applications*, 2nd ed. Springer-Verlag, 2003.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[8] D. Yu, G. Hinton, N. Morgan, J.-T. Chien, and S. Sagayama, "Introduction to the special section on deep learning for speech and language processing," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 4–6, 2011.

[9] N. Sünderhauf, O. Brock, W. Scheirer, R. Hadsell, D. Fox, J. Leitner, B. Upcroft, P. Abbeel, W. Burgard, M. Milford *et al.*, "The limits and potentials of deep learning for robotics," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 405–420, 2018.

[10] J. Kivinen, A. J. Smola, and R. C. Williamson, "Online learning with kernels," *IEEE transactions on signal processing*, vol. 52, no. 8, pp. 2165–2176, 2004.

[11] Y. Zhang, P. Liang, and M. J. Wainwright, "Convexified convolutional neural networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 4044–4053.

[12] T. Fushiki, "Estimation of prediction error by using k-fold cross-validation," *Statistics and Computing*, vol. 21, no. 2, pp. 137–146, 2011.

[13] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 281–305, 2012.

[14] M. Mitchell, *An introduction to genetic algorithms*. MIT press, 1998.

[15] T. Broderick, N. Boyd, A. Wibisono, A. C. Wilson, and M. I. Jordan, "Streaming variational bayes," in *Advances in neural information processing systems*, 2013, pp. 1727–1735.

[16] T. D. Bui, J. Yan, and R. E. Turner, "A unifying framework for gaussian process pseudo-point approximations using power expectation propagation," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 3649–3720, 2017.

[17] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6765–6816, 2017.

[18] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "Dropblock: A regularization method for convolutional networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 10 727–10 737.

[19] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[20] B. Wang, A. Koppel, and V. Krishnamurthy, "A markov decision process approach to active meta learning," *arXiv preprint arXiv:2009.04950*, 2020.

[21] A. S. Bedi, D. Peddireddy, V. Aggarwal, and A. Koppel, "Efficient gaussian process bandits by believing only informative actions," *arXiv preprint arXiv:2003.10550*, 2020.

[22] J. Zhu and T. Hastie, "Kernel Logistic Regression and the Import Vector Machine," *Journal of Computational and Graphical Statistics*, vol. 14, no. 1, pp. 185–205, 2005.

[23] M. Benaim, J. Hofbauer, and S. Sorin, "Stochastic approximations and differential inclusions, Part II: Applications," *Mathematics of Operations Research*, vol. 31, no. 3, pp. 673–695, 2006.

[24] O. Namvar, V. Krishnamurthy, and G. Yin, "Distributed tracking of correlated equilibria in regime switching noncooperative games," *IEEE Transactions Automatic Control*, vol. 58, no. 10, pp. 2435–2450, 2013.

[25] V. Solo and X. Kong, *Adaptive Signal Processing Algorithms – Stability and Performance*. N.J.: Prentice Hall, 1995.

[26] V. S. Borkar, *Stochastic approximation: a dynamical systems viewpoint*. Springer, 2009, vol. 48.

[27] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[28] J. M. Leiva-Murillo and A. ArtéS-RodríGuez, "Algorithms for maximum-likelihood bandwidth selection in kernel density estimators," *Pattern Recognition Letters*, vol. 33, no. 13, pp. 1717–1724, 2012.

[29] R. Wheeden, R. Wheeden, and A. Zygmund, *Measure and Integral: An Introduction to Real Analysis*, ser. Chapman & Hall/CRC Pure and Applied Mathematics. Taylor & Francis, 1977. [Online]. Available: https://books.google.com/books?id=YDkDmQ_hdmcC

[30] G. Kimeldorf and G. Wahba, "Some results on tchebycheffian spline functions," *Journal of mathematical analysis and applications*, vol. 33, no. 1, pp. 82–95, 1971.

[31] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," *Subseries of Lecture Notes in Computer Science Edited by JG Carbonell and J. Siekmann*, p. 416, 2001.

[32] A. Koppel, G. Warnell, E. Stump, and A. Ribeiro, "Parsimonious online learning with kernels via sparse projections in function space," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 83–126, 2019.

[33] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization." *Journal of machine learning research*, vol. 12, no. 7, 2011.

[34] H. J. Kushner and A. Shwartz, "Stochastic approximation in Hilbert space: Identification and optimization of linear continuous parameter systems," *SIAM journal on control and optimization*, vol. 23, no. 5, pp. 774–793, 1985.

[35] M. Hamdi, V. Krishnamurthy, and G. Yin, "Tracking a markov-modulated stationary degree distribution of a dynamic random graph," *IEEE Transactions on Information Theory*, vol. 60, no. 10, pp. 6609–6625, 2014.

[36] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.