2024 32nd Mediterranean Conference on Control and Automation (MED)
June 11-14, 2024 | Chania, Crete, Greece

Constructive Function Approximation with Local Models

Christos N. Mavridis, and Karl Henrik Johansson

***Abstract*— We introduce a constructive function approximation approach as a general tool, particularly useful in adaptive and data-driven methods for perception and control. The key idea is to estimate of a collection of simple local models as opposed to a single and complex regression model trained in the entire input space. We use principles from the Online Deterministic Annealing (ODA) optimization framework to construct an adaptive partition of the input space, which enables the introduction of local function approximation models within each subset of the partition. We show that both the partitioning and the local model training algorithms are stochastic approximation algorithms that operate online, and with the same observations, as part of a two-timescale stochastic approximation scheme. This process constitutes a heuristic method to gradually increase the complexity of the function approximation framework in a task-agnostic manner, giving emphasis to regions of the input space where the regression error is high. As a result this framework has inherent explainability properties, and is suitable for continuous learning applications where regression improvement without retraining from scratch is crucial. Simulation results illustrate the properties of the proposed approach.**

I. Introduction

Learning from observations is pivotal to autonomous decision-making and communication systems. Mathematically, such learning problems are often formulated as constrained stochastic optimization problems: given realizations of a random variable $X \in S$ representing the observations, an optimal parameter vector $\theta \in \Theta$ is to be found such that a well-defined error measure between an unknown function $f(X) \in \mathcal{F}$ and a learning model $\hat{f}(X, \theta) \in \mathcal{F}$, parameterized by θ, is minimized under potentially additional constraints. However, the solution of such problems over the entire domain S often requires the learning model $\hat{f}(X, \theta)$ to be particularly complex, making the estimation of θ costly, and raising issues with respect to phenomena such as over-fitting, generalization, and robustness, connected by an underlying trade-off between complexity and performance [1], [2]. As a result, the ability to gradually approximate a solution to these problems is essential to decision-making systems that often operate in real-time and under limitations in memory and computational resources.

Current deep learning methods have made progress towards the construction of a hierarchical representation of the data space [3]–[6]. However, such approaches do not necessarily satisfy the above description of hierarchical learning, since they typically use overly complex models over the entire data space S, which comes in the expense of time, energy, data, memory, and computational resources [7], [8]. In this work, we are mainly focusing on a framework for hierarchical progressive learning and data representation, where a gradually growing and hierarchically structured set of learning models is used for function approximation. We consider a prototype-based learning framework where, given random observations of $X \in S$, a set of prototypes $\{\mu_i\}$, $\mu_i \in S$ (also called codevectors or neurons), are scattered in the data space S to encode subsets/regions $\{S_i\}$ that form a partition of S [9]. This adheres to the principles of vector quantization for signal compression [10]. In this regard, a knowledge representation can be defined as the set of codevectors $\{\mu_i \in S\}$ that induce a structured partition $\{S_i\}$ of the data space S, along with a set of local learning models $\hat{f}(x, \theta_i)$ associated with each region S_i, parameterized by their own set of parameters θ_i. A structured representation like this allows, among other things, to locate specific regions of the space that the algorithm needs to approximate in greater detail, according to the problem at hand and the designer's requirements. This results in adaptively allocating more resources only in the subsets of the data space that are needed, and provides benefits in terms of time, memory, and model complexity. Moreover, learning with local models that take advantage of the differences in the underlying distribution of the data space provides a means to understand certain properties of the data space itself, i.e., this is an interpretable learning approach [11]–[13]. An illustration of this framework is given in Fig. 1.

Regarding the learning process, we are interested in algorithms that are able to simultaneously solve both the problems of partitioning and function approximation, given online (e.g., real-time) observations. This is of great importance in many applications, and especially in the scope of learning algorithms for inference and control in general cyber-physical systems [1], [14]–[16], as well as complex hybrid systems [17], [18]. To construct a sequence of partitions with increasing number of subsets we build upon the notion of Online Deterministic Annealing [19] and define a series of soft-clustering optimization problems:

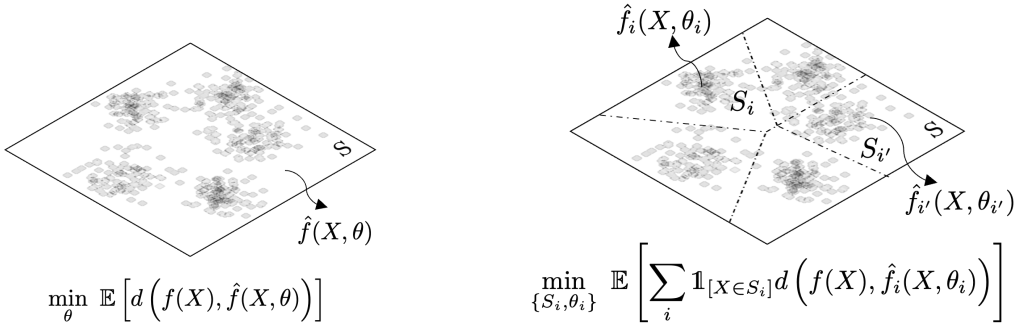
$$\min_{\{\mu_i\}} F_\lambda(X, Q) := (1 - \lambda) D(X, Q) - \lambda H(X, Q),$$

parameterized by a Lagrange coefficient $\lambda \in [0, 1]$ controlling the trade-off between minimizing an average distortion measure $D(X, Q) := \mathbb{E}[d(X, Q)]$, for an appropriately defined dissimilarity measure d, and maximizing the Shannon entropy $H(X, Q)$, with $H(X, Q) :=$

The authors are with the Division of Decision and Control Systems, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm. emails:{mavridis,kallej}@kth.se.

Research partially supported by the Swedish Foundation for Strategic Research (SSF) grant IPD23-0019.

2024 32nd Mediterranean Conference on Control and Automation (MED) | 979-8-3503-9544-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/MED61351.2024.10566262



(a) Classical regression problem.

(b) Combined problem of partitioning and function approximation.

Fig. 1: Comparison of the classical regression problem over the entire domain S with the problem of combined partitioning and regression within each subset of the partition. Here the input $X \in S$ is a random variable and the function $f(X)$ is to be estimated over S by (a) a single learning model $\hat{f}(X,\theta)$, and (b) a set of $\left\{\hat{f}(X,\theta_i)\right\}$ defined in each region S_i, where $\{S_i\}$ is a partition of S to be estimated as well.

$\mathbb{E}[-\log p(X,Q)]$. The novelty of the approach lies in the definition of Q as a random variable described by the association probabilities $p(\mu_i|X=x)$ that represents the probability of a data point x to belong to the subset $S_i := \left\{x \in S : i = \arg\min_j d(x,\mu_j)\right\}$. Once the joint probability space of (X,Q) is defined, successively solving the optimization problems $\min_{\{\mu_i\}} F_\lambda(X,Q)$ for decreasing values of λ, leads in a series of bifurcation phenomena when the cardinality of the set of codevectors $\{\mu_i\}$ increases, resembling an annealing process that introduces inherent robustness and regularization properties [19], [20].

An important property of this approach, initially shown in [19], is that the optimization problems $\min_{\{\mu_i\}} F_\lambda(X,Q)$ can be solved online, using gradient-free stochastic approximation updates [21], as long as the measure d belongs to the family of Bregman divergences, information-theoretic dissimilarity measures that include, among others, the widely used squared Euclidean distance and Kullback-Leibler divergence [22], [23]. We exploit the fact that a stochastic approximation algorithm can be used as a training rule for constructing the partition $\{S_i\}$, to build a framework that simultaneously trains the learning models $\left\{\hat{f}(x,\theta_i)\right\}$ defined in each region S_i. In particular, according to the theory of two-timescale stochastic approximation [21], we define two stochastic approximation algorithms that run at the same time and with the same observations but with different stepsize schedules that define a fast and a slow learning process. The slow process approximates the parameters $\{\mu_i\}$ and as a result the partition $\{S_i\}$, and the fast process executes a function approximation algorithm within each S_i to find the optimal parameters θ_i for the learning model $\hat{f}(x,\theta_i)$.

The paper is organized as follows: Section II introduces the Online Deterministic Annealing (ODA) framework for progressive partitioning along with a mathematical analysis of its properties. Section III develops the two-timescale framework for combined partitioning and function approximation. Finally, Section IV illustrates simulation results, and Section V concludes the paper.

II. Online Deterministic Annealing

We start our analysis with the case of unsupervised learning, where partitioning a space S is equivalent to the problem of clustering and density estimation. In this context, the observations (data) are independent realization of a random variable $X : \Omega \to S$ defined in a probability space $(\Omega,\mathcal{F},\mathbb{P})$, where $S \subseteq \mathbb{R}^d$ is the observation space (data space). In the Online Deterministic Annealing approach [1], [19], one defines a similarity measure $d : S \to ri(S)$[1], and a discrete random variable $Q : S \to ri(S)$ with domain $\mu := \{\mu_i\}_{i=1}^K$, $\mu_i \in ri(S)$ described by the association probabilities $\{p(\mu_i|x) := \mathbb{P}[Q=\mu_i|X=x]\}$, $\forall i$, such that

$$\min_{\mu} F_\lambda(\mu) := (1-\lambda)D(\mu) - \lambda H(\mu) \tag{1}$$

This is a multi-objective optimization problem formulated as the minimization of a Lagrangian function, where $\lambda \in [0,1)$ acts as a Lagrange multiplier controlling the trade-off between the average distortion:

$$\begin{aligned}\min_{\mu}\ D(\mu) &:= \mathbb{E}\left[d\left(X,Q\right)\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[d(X,Q)|X\right]\right] \\ &= \int p(x)\sum_i p(\mu_i|x)d(x,\mu_i)\ dx\end{aligned}$$

and the entropy term:

$$\begin{aligned}H(\mu) &:= \mathbb{E}\left[-\log P(X,Q)\right] \\ &= H(X) - \int p(x)\sum_i p(\mu_i|x)\log p(\mu_i|x)\ dx.\end{aligned}$$

The entropy H, acts as a regularization term, and is given progressively less weight as λ decreases. The term $T := \frac{\lambda}{1-\lambda}$, $\lambda \in [0,1)$ can be seen as a temperature coefficient in a deterministic annealing process [19].

Following the Online Deterministic Annealing (ODA) approach [1], [19], we minimize F_λ in (1) by successively

[1] $ri(S)$ represents the relative interior of S.

minimizing it first respect to the association probabilities $\{p(\mu_i|x)\}$, and then with respect to the codevector locations μ. The solution of the optimization problem

$$F_\lambda^*(\mu) := \min_{\{p(\mu_i|x)\}} F_\lambda(\mu) \quad \text{s.t.} \quad \sum_i p(\mu_i|x) = 1, \tag{2}$$

is given by the Gibbs distributions

$$p^*(\mu_i|x) = \frac{e^{-\frac{1-\lambda}{\lambda} d(x,\mu_i)}}{\sum_j e^{-\frac{1-\lambda}{\lambda} d(x,\mu_j)}}, \ \forall x \in S. \tag{3}$$

Furthermore, it has been shown in [19] that if $d := d_\phi$ is a Bregman divergence[2], then the conditional expectation:

$$\mu_i^* = \mathbb{E}\left[X|\mu_i\right] = \frac{\int x p(x) p^*(\mu_i|x) \ dx}{p^*(\mu_i)} \tag{4}$$

is a solution to the optimization problem

$$\min_\mu F_\lambda^*(\mu). \tag{5}$$

Moreover, a stochastic approximation algorithm can be formulated [19] to recursively estimate $\mathbb{E}\left[X|\mu_i\right]$ directly, according to Theorem 1.

Theorem 1 ([19]): Let $\{x_n\}$ be a sequence of independent realizations of X. Then $\mu_i(n)$, defined by the online training rule

$$\begin{cases} \rho_i(n+1) & = \rho_i(n) + \alpha(n)\left[\hat{p}(\mu_i|x_n) - \rho_i(n)\right] \\ \sigma_i(n+1) & = \sigma_i(n) + \alpha(n)\left[x_n \hat{p}(\mu_i|x_n) - \sigma_i(n)\right] \end{cases} \tag{6}$$

where $\sum_n \alpha(n) = \infty$, $\sum_n \alpha^2(n) < \infty$, and the quantities $\hat{p}(\mu_i|x_n)$ and $\mu_i(n)$ are recursively updated as follows:

$$\mu_i(n) = \frac{\sigma_i(n)}{\rho_i(n)}, \quad \hat{p}(\mu_i|x_n) = \frac{\rho_i(n) e^{-\frac{1-\lambda}{\lambda} d(x_n,\mu_i(n))}}{\sum_i \rho_i(n) e^{-\frac{1-\lambda}{\lambda} d(x_n,\mu_i(n))}} \tag{7}$$

converges almost surely to a locally asymptotically stable solution of the optimization (5), as $n \to \infty$.

Remark 1: Notice that we can express the dynamics of the codevector parameters $\mu_i(n)$ directly as:

$$\mu_i(n+1) = \frac{\alpha(n)}{\rho_i(n)}\left[\frac{\sigma_i(n+1)}{\rho_i(n+1)}(\rho_i(n) - \hat{p}(\mu_i|x_n)) + (x_n \hat{p}(\mu_i|x_n) - \sigma_i(n))\right] \tag{8}$$

where the recursive updates take place for every codevector μ_i sequentially. This is a discrete-time dynamical system that presents bifurcation phenomena with respect to the parameter λ, i.e., the number of equilibria of this system changes with respect to the value λ which is hidden inside the term $\hat{p}(\mu_i|x_n)$ in (7). According to this phenomenon, the number of distinct values of μ_i is finite, and the updates need only be taken with respect to these values that we call "effective codevectors". This is discussed in Section II-A.

Finally, in the limit $\lambda \to 0$, (6) and (7) result in a consistent density estimator, i.e., the representation of the random variable $X \in S$ by the codevectors μ becomes all the more accurate in S, according to the underlying probability density $p(x)$ [1], [12].

A. Bifurcation, Algorithmic Implementation, and Complexity

First, notice that when $\lambda \to 1$ (resp. $T \to \infty$) equation (3) yields uniform association probabilities $p(\mu_i|x) = p(\mu_j|x), \ \forall i,j,\forall x$. As a result of (4), all codevectors are located at the same point:

$$\mu_i = \mathbb{E}\left[X\right], \ \forall i$$

which means that there is one unique effective codevector given by $\mathbb{E}\left[X\right]$.

As λ is lowered below a critical value, a bifurcation phenomenon occurs, when the number of effective codevectors increases. Mathematically, this occurs when the existing solution μ^* given by (4) is no longer the minimum of the free energy F^*, as λ (resp. the temperature T) crosses a critical value. Following principles from variational calculus, we can rewrite the necessary condition for optimality (4) as

$$\frac{d}{d\epsilon} F^*(\mu + \epsilon\psi)|_{\epsilon=0} = 0 \tag{9}$$

with the second order condition being

$$\frac{d^2}{d\epsilon^2} F^*(\{\mu + \epsilon\psi\})|_{\epsilon=0} \geq 0 \tag{10}$$

for all choices of finite perturbations $\{\psi\}$. Here we denote by $\{y := \mu + \epsilon\psi\}$ a perturbed codebook, where ψ are perturbation vectors applied to the codevectors μ, and $\epsilon \geq 0$ is used to scale the magnitude of the perturbation. Bifurcation occurs when equality is achieved in (10) and hence the minimum is no longer stable[3]. These conditions are described in the Theorem 2.

Theorem 2: Bifurcation occurs under the following condition

$$\exists y_i \text{ s.t. } p(y_i) > 0 \text{ and } \det\left[I - \frac{1-\lambda}{\lambda} \frac{\partial^2 \phi(y_i)}{\partial y_i^2} C_{X|y_i}\right] = 0, \tag{11}$$

where $C_{X|y_i} := \mathbb{E}\left[(X - y_i)(X - y_i)^{\mathrm{T}}|y_i\right]$.

Proof: See [12]. ∎

In other words, there exist critical values for λ that depend on the data space itself and the choice of the Bregman divergence (through the function ϕ), such that bifurcation occurs when

$$\frac{\lambda}{1-\lambda} = \frac{\partial^2 \phi(y_n)}{\partial y_n^2} \bar{\nu} \tag{12}$$

where $\bar{\nu}$ is the largest eigenvalue of $C_{X|y_n}$. That is to say that an algorithmic implementation needs only as many codevectors as the number of effective codevectors, which

[2]The function d_ϕ is a Bregman divergence if there exists a strictly convex function ϕ such that $d_\phi(x,\mu) = \phi(x) - \phi(\mu) - \frac{\partial \phi}{\partial \mu}(\mu)(x-\mu)$. Two notable examples are the squared Euclidean distance $d_\phi(x,\mu) = \|x-\mu\|^2$ ($\phi(x) = \langle x, x\rangle, \ x \in \mathbb{R}^d$), and the the generalized Kullback-Leibler divergence $d_\phi(x,\mu) = \langle x, \log x - \log \mu\rangle - \langle \mathbb{1}, x - \mu\rangle$ ($\phi(x) = \langle x, \log x\rangle, \ x \in \mathbb{R}_{++}^d$). For more details see, e.g., [1], [22], [24].

[3]For simplicity we ignore higher order derivatives, which should be checked for mathematical completeness, but which are of minimal practical importance. The result is a necessary condition for bifurcation.

depends only on changes of the temperature parameter below certain thresholds that depend on the dataset at hand and the dissimilarity measure used. However, we can detect the bifurcation points by introducing perturbing pairs of codevectors at each temperature level λ (resp. T). In this way, the codevectors μ are doubled by inserting a perturbation of each μ_i in the set of effective codevectors. The newly inserted codevectors will merge with their pair if a critical temperature has not been reached and separate otherwise. For more details about the implentation of the algorithm the readers are referred to [12], [19].

The complexity of Alg. 1 for fixed coefficient λ_t is $O(N_{c_t}(2K_t)^2 d)$, where N_{c_t} is the number of stochastic approximation iterations needed for convergence which corresponds to the number of data samples observed, K_t is the number of codevectors of the model at temperature λ_t, and d is the dimension of the input vectors, i.e., $x \in \mathbb{R}^d$. Therefore, assuming a coefficient schedule $\{\lambda_1 = \lambda_{max}, \lambda_2, \ldots, \lambda_{N_\lambda} = \lambda_{min}\}$, the time complexity for the training of Algorithm 1 becomes: $O(N_c(2\bar{K})^2 d)$, where $N_c = \max_i \{N_{c_t}\}$ is an upper bound on the number of data samples observed until convergence at each temperature level, and $\bar{K} = \sum_{i=1}^{N_\lambda} K_t$, with $N_\lambda \leq \bar{K} \leq \min\left\{\sum_{n=0}^{N_\lambda - 1} 2^n, \sum_{n=0}^{\log_2 K_{max}} 2^n\right\} < N_\lambda K_{max}$. The actual value of $\bar{K}$ depends on the bifurcations occurred as a result of reaching critical temperatures and the effect of the regularization mechanisms described above. Note that typically $N_c \ll N$ as a result of the stochastic approximation algorithm, and $\bar{K} \ll N_\lambda K_{max}$ as a result of the progressive nature of the algorithm. Prediction scales linearly with $O(K_{N_\lambda} d)$, with $K_{N_\lambda} \leq K_{max}$.

Algorithm 1 Progressive Partitioning.

Select a Bregman divergence d_ϕ
Set stopping criteria T_{stop} (e.g., K_{max}, λ_{min})
Set convergence parameters: γ, ϵ_c, ϵ_n, ϵ_r, δ
Set stepsizes: $\{\alpha_n\}$
Initialize: $K = 1$, $\lambda = 1$,
$\{\mu_0\}$, $p(\mu_0) = 1$, $\sigma(\mu_0) = \mu_0 p(\mu_0)$
repeat
  Perturb codebook: $\{\mu_i\} \leftarrow \{\mu_i + \delta\} \bigcup \{\mu_i - \delta\}$
  Update $K \leftarrow 2K$, $\{p(\mu_i)\}$, $\{\sigma(\mu_i) \leftarrow \mu_i p(\mu_i)\}$
  $n \leftarrow 0$
  repeat
    Observe data point x
    for $i = 1, \ldots, K$ **do**
      Update:
$$p(\mu_i|x) \leftarrow \frac{p(\mu_i) e^{-\frac{1-\lambda}{\lambda} d_\phi(x, \mu_i)}}{\sum_i p(\mu_i) e^{-\frac{1-\lambda}{\lambda} d_\phi(x, \mu_i)}}$$
$$p(\mu_i) \leftarrow p(\mu_i) + \alpha_n \left[p(\mu_i|x) - p(\mu_i)\right]$$
$$\sigma(\mu_i) \leftarrow \sigma(\mu_i) + \alpha_n \left[x p(\mu_i|x) - \sigma(\mu_i)\right]$$
$$\mu_i \leftarrow \frac{\sigma(\mu_i)}{p(\mu_i)}$$
    $n \leftarrow n + 1$
    end for
  until Convergence: $\frac{1-\lambda}{\lambda} d_\phi(\mu_i^n, \mu_i^{n-1}) < \epsilon_c$, $\forall i$
  Keep effective codevectors:
    discard μ_i if $\frac{1-\lambda}{\lambda} d_\phi(\mu_j, \mu_i) < \epsilon_n$, $\forall i, j, i \neq j$
  Remove idle codevectors:
    discard μ_i if $p(\mu_i) < \epsilon_r$, $\forall i$
  Update K, $\{p(\mu_i)\}$, $\{\sigma(\mu_i)\}$
  Lower temperature: $\lambda \leftarrow \gamma\lambda$
until T_{stop}

III. Learning with Local Models

In this section, we investigate the problem of combined partitioning and function approximation, where multiple local models are trained, taking advantage of the differences in the underlying probability distribution of the data space. As a consequence, this approach can circumvent the use of overly complex learning models, reduce time, memory, and computational complexity, and give insights to certain properties of the data space [13].

First, we assume a function $f : S \to \mathcal{F}$ and models $\hat{f}_i(x, \theta_i)$, $f_i : S \to \mathcal{F}$, that are differentiable with respect to a parameter vector $\theta_i \in \Theta$, where Θ is a finite-dimensional vector space. The problem of finding the optimal partition parameters $\{S_i\}_{i=1}^{K(\lambda)}$, for $K(\lambda) < \infty$, given the local model parameters $\{\theta_i\}$, can be formulated as an online deterministic annealing problem of the form (1) in the augmented space of the random variable

$$Z := \begin{bmatrix} X \\ f(X) \end{bmatrix} \in S \times \mathcal{F} \tag{13}$$

and reads as

$$\min_{\{\mu_i\}} \; F_\lambda \left(d\left(Z, Q\right)\right), \tag{14}$$

where the quantizer $Q : S \to S \times \mathcal{F}$ is a stochastic mapping of the form:

$$Q(x) = \begin{cases} \begin{bmatrix} \mu_1 \\ \hat{f}(x, \theta_1) \end{bmatrix}, & p(\mu_1|x) \\ \vdots & \\ \begin{bmatrix} \mu_{\bar{K}} \\ \hat{f}(x, \theta_{\bar{K}}) \end{bmatrix}, & p(\mu_{\bar{K}}|x) \end{cases} \tag{15}$$

It is easy to see that, assuming $\{\mu_i\}$ are the controlled parameters of the mapping Q and $\{\theta_i\}$ are the uncontrolled parameters (to be estimated by external algorithm), equations (3) and (4) continue to hold for $\{p(\mu_i|x)\}$ and $\{\mu_i\}$, respectively. That is, the online deterministic algorithm can be used in the augmented space $S \times \mathcal{F}$ to adaptively estimate a partition of the input space S such that, the error (14) is minimized.

Conversely, given a finite partition set of parameters $\{S_i\}_{i=1}^{K(\lambda)}$, for $K(\lambda) < \infty$, the local function approximation problem is formulated as:

$$\min_{\theta_i} \; \mathbb{E}\left[\mathbb{1}_{[X \in S_i]} d\left(f(X), \hat{f}_i(X, \theta_i)\right)\right], \; i = 1, \ldots, K(\lambda). \tag{16}$$

where $d : \mathcal{F} \times \mathcal{F} \rightarrow [0, \infty)$ is assumed a metric that is differentiable and convex with respect to the second argument. This is a stochastic optimization problem that can be solved using stochastic approximation updates. In particular, one can use stochastic gradient descent:

$$\theta_i(n+1) = \theta_i(n) - \beta(n) \nabla_\theta d(f(x_n), \hat{f}_i(x_n, \theta_i(n))) \tag{17}$$

which is a special case of a stochastic approximation algorithm that, under mild assumptions, converges almost surely to an asymptotically stable local minimum of the objective function $\mathbb{E}\left[\mathbb{1}_{[X \in S_i]} \hat{d}(f(x_n), f_i(x, \theta_i))\right]$ (see e.g., [18]).

However, we are interested in a learning approach that approximates $\{S_i\}$ and $\left\{\hat{f}_i(x, \theta_i)\right\}$ at the same time, and given the same observations $\{(x_n, f(x_n))\}$ which may be available one at a time (i.e, no dataset is stored in memory a priori). This is possible because both learning algorithms for $\{S_i\}$ and $\left\{\hat{f}_i(x, \theta_i)\right\}$ independently are stochastic approximation algorithms. According to the theory of two-timescale stochastic approximation, we can run both learning algorithms at the same time, but using different stepsize profiles $\{\alpha(n)\}$ and $\{\beta(n)\}$, such that $\alpha(n)/\beta(n) \rightarrow 0$. Intuitively, we create a system of two dynamical system running in different "speed", meaning that second system, the one with stepsizes $\{\beta(n)\}$, is updated fast enough that the first system, the one with stepsizes $\{\alpha(n)\}$, can be seen as quasi-static with respect to the second. The following theorem follows directly from the results of Ch. 6 in [25].

Theorem 3 ([12]): Let $\{x_n\}$ be a sequence of independent realizations of X, and assume that $\mu_i(n)$ is a sequence updated using the stochastic approximation algorithm in (6) with stepsizes $\{\alpha(n)\}$ satisfying $\sum_n \alpha(n) = \infty$, and $\sum_n \alpha^2(n) < \infty$. Then, as long as $\{\beta(n)\}$ are designed such that $\sum_n \beta(n) = \infty$, $\sum_n \beta^2(n) < \infty$, and $\alpha(n)/\beta(n) \rightarrow 0$, the asynchronous updates

$$\theta_i(n+1) = \theta_i(n) - \beta(n) \nabla_\theta d(f(x_n), \hat{f}_i(x_n, \theta_i(n))), \tag{18}$$

for $i = \arg\min_j \ d_\phi(x_n, \mu_j(n))$ converges almost surely to a locally asymptotically stable solution $\{\theta_i\}$ of (16), as $n \rightarrow \infty$, for $S_i = \{x \in S : i = \arg\min_j \ d_\phi(x, \mu_j(\infty))\}$, where $\mu_i(\infty))$ is the asymptotically stable equilibrium of (6).

Proof: Follows directly from the two-timescale approximation theory [12]. ■

IV. Simulation Results

We illustrate the properties and evaluate the performance of the proposed learning algorithm in two simple regression problems. In Fig. 2, the evolution of the proposed algorithm is depicted in an one-dimensional function approximation problem using linear local models. The choice of simple linear models is made to illustrate the properties of the approach. Notice that, at first (high temperature coefficient λ), a single linear model is trained. Since the model is not rich enough to capture the original function in the entire space, the regression error is high across the entire input space. As λ decreases, the regions of the input space that correspond to more complex behavior in the output space are gradually divided into finer partitions. As a result, a collection of linear local models is constructed, reducing the function approximation error while increasing the complexity of the algorithm. This process showcases the performance-complexity trade-off described in Section II. Finally, in Fig. 3, we test the proposed methodology in a 2D regression problem using constant local models, resulting in a piecewise constant function approximation scheme.

V. Conclusion and Future Work

We introduced a function approximation framework, where, instead of a single and complex regression model, trained in the entire input space, a collection of simpler local models is used. We used principles from the online deterministic annealing optimization framework to construct an adaptive partition of the input space, which enables the introduction of local function approximation models within each subset of the partition. The proposed method constitutes a heuristic method to gradually increase the complexity of the function approximation framework in a task-agnostic manner, giving emphasis to regions of the input space where the regression error is high. As a result this framework has inherent explainability properties, and is suitable for continuous learning applications where regression improvement without re-training from scratch is crucial.

The properties of the proposed approach will be investigated in the context of closed-loop data-driven control, cyber-physical security, and adaptive identification.

References

[1] C. Mavridis and J. S. Baras, "Annealing optimization for progressive learning with stochastic approximation," *IEEE Transactions on Automatic Control*, vol. 68, no. 5, pp. 2862–2874, 2023.

[2] K. P. Bennett and E. Parrado-Hernández, "The interplay of optimization and machine learning research," *The Journal of Machine Learning Research*, vol. 7, pp. 1265–1281, 2006.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[4] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[5] 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.

[6] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 609–616.

[7] N. C. Thompson, K. Greenewald, K. Lee, and G. F. Manso, "The computational limits of deep learning," *arXiv preprint arXiv:2007.05558*, 2020.

[8] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in nlp," *arXiv preprint arXiv:1906.02243*, 2019.

[9] M. Biehl, B. Hammer, and T. Villmann, "Prototype-based models in machine learning," *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 7, no. 2, pp. 92–111, 2016.

[10] T. Kohonen, *Learning Vector Quantization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 175–189.

[11] C. Mavridis and J. S. Baras, "Explainable learning with hierarchical online deterministic annealing," in *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD), Workshop on Uncertainty meets Explainability in Machine Learning*, 2023.

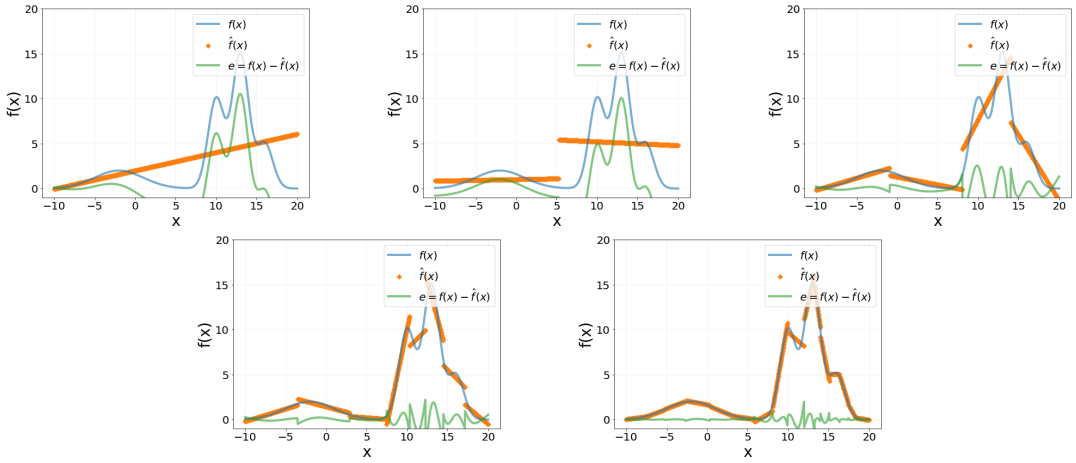
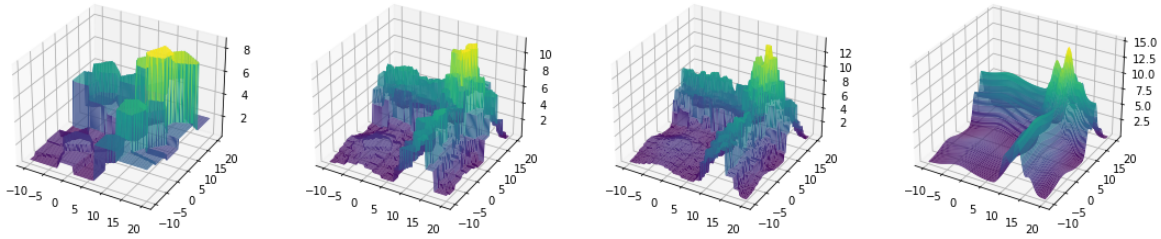
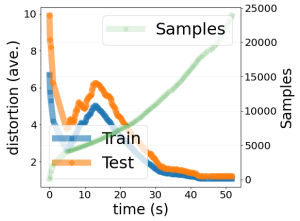


Fig. 2: Evolution of the proposed algorithm with linear local models, resulting in piece-wise linear function approximation in 1D. The regions of the input space that correspond to more complex behavior in the output space are gradually divided into finer partitions.



(a) Evolution of the algorithm in the data space.



(b) Performance curves.

Fig. 3: Performance curves and data space evolution of the proposed algorithm resulting in piece-wise constant function approximation in 2D.

[12] C. Mavridis and J. Baras, "Multi-resolution online deterministic annealing: A hierarchical and progressive learning architecture," *arXiv preprint arXiv:2212.08189*, 2022.

[13] S. Rüping, "Learning with local models," in *Local Pattern Detection*, K. Morik, J.-F. Boulicaut, and A. Siebes, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 153–170.

[14] C. N. Mavridis, G. P. Kontoudis, and J. S. Baras, "Sparse gaussian process regression using progressively growing learning representations," in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 1454–1459.

[15] C. N. Mavridis and J. S. Baras, "Progressive graph partitioning based on information diffusion," in *IEEE Conference on Decision and Control*, 2021, pp. 37–42.

[16] C. N. Mavridis, A. Kanellopoulos, K. G. Vamvoudakis, J. S. Baras, and K. H. Johansson, "Attack identification for cyber-physical security in dynamic games under cognitive hierarchy," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 11 223–11 228, 2023.

[17] C. N. Mavridis and J. S. Baras, "Identification of piecewise affine systems with online deterministic annealing," in *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE, 2023, pp. 4885–4890.

[18] C. N. Mavridis, A. Kanellopoulos, J. S. Baras, and K. H. Johansson, "State-space piece-wise affine system identification with online deterministic annealing," in *European Control Conference (ECC)*. IEEE, 2024.

[19] C. N. Mavridis and J. S. Baras, "Online deterministic annealing for classification and clustering," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 10, pp. 7125–7134, 2023.

[20] K. Rose, "Deterministic annealing for clustering, compression, classification, regression, and related optimization problems," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2210–2239, 1998.

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

[22] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, "Clustering with bregman divergences," *Journal of machine learning research*, vol. 6, no. Oct, pp. 1705–1749, 2005.

[23] T. Villmann, S. Haase, F.-M. Schleif, B. Hammer, and M. Biehl, "The mathematics of divergence based online learning in vector quantization," in *IAPR Workshop on Artificial Neural Networks in Pattern Recognition*. Springer, 2010, pp. 108–119.

[24] C. N. Mavridis and J. S. Baras, "Convergence of stochastic vector quantization and learning vector quantization with bregman divergences," *IFAC-PapersOnLine*, vol. 53, no. 2, 2020.

[25] V. S. Borkar, "Stochastic approximation with two time scales," *Systems & Control Letters*, vol. 29, no. 5, pp. 291–294, 1997.