In: 2024 American Control Conference (ACC), 10-12 July 2024, Toronto, Canada
DOI:10.23919/ACC60939.2024.10644991

Π-ORFit: One-Pass Learning with Bregman Projection

Namhoon Cho, Youngjae Min, Hyo-Sang Shin, and Navid Azizan

***Abstract*— This paper delves into the problem of one-pass learning, where the objective is to train a model on each datapoint in a stream while maintaining performance on past data without retraining on them. An existing approach to this problem in the context of overparameterized (underdetermined) models is Orthogonal Recursive Fitting (ORFit), which fits every new data point while maintaining predictions on previous datapoints by ensuring that parameter updates are orthogonal to the directions that are critical for past data (i.e., the direction of gradient of the model output with respect to the parameters, for those data). For overparameterized linear models, when initialized at zero, ORFit obtains the parameter vector that perfectly fits the data and has the minimum ℓ^2-norm, among the infinitely many perfectly fitting parameter vectors. To generalize this and gain control over the selection of desired parameters, in this paper, we introduce Projected Orthogonal Recursive Fitting (Π-ORFit). We begin by characterizing all parameters that can precisely fit data in general vector-output linear models, employing a formalism based on nullspace projector matrices. This framework yields an alternative derivation of ORFit. Building on this, we further extend ORFit to learn a desired parameter by incorporating a Bregman projection into the update rule. Importantly, we show that the resulting parameter minimizes the potential function that defines the Bregman projection at each update step, enabling the selection of a desired parameter among the (infinitely many) candidates consistent with the data. We provide numerical experiments that validate our analytical findings and underscore the practical significance of this generalized approach.**

I. Introduction

Efficiently learning from a continuous stream of data, capturing their complete information without excessive memory requirements, is of paramount importance, particularly in real-time applications for physical systems. In scenarios where retaining a large data buffer is infeasible, it becomes crucial to capture the essential information with a single pass through the data sequence. However, this paradigm, known as one-pass learning, entails a major challenge to mitigate catastrophic forgetting without resorting to expansive memory buffers.

A compelling approach to overcome the catastrophic forgetting issue in one-pass learning is to design novel recursive optimizers that solve optimization problems for the entire data. This is in contrast to solely tweaking the objective of the optimization problem while adopting off-the-shelf gradient descent optimizers. A prominent example of a recursive optimizer is Orthogonal Recursive Fitting (ORFit) [1], which was recently proposed and considers the geometry of contour lines for the prediction model in the parameter space. This method shares the same spirit with the Orthogonal Gradient Descent (OGD) [2], proposed in the context of continual learning in that both methods steer parameter updates orthogonally to the gradients of the past predictions with respect to the parameter. The underlying reasoning is that the model's output changes the most when moving the parameter along the direction of the gradient of the predicted output, while moving orthogonal to this direction causes the least change.

Recent research highlights that the traditional Recursive Least Squares (RLS) framework can still inspire the development of one-pass learning algorithms. Notably, [1] established a link between ORFit and the discrete-time RLS method in the case of linear prediction models. Further, [3] introduced a distribution-free one-pass method for lifelong learning under possibly time-varying true data distribution, akin to the discrete-time RLS with exponential forgetting factor. When dealing with overparameterized models, where the number of parameters exceeds the number of datapoints, the underdetermined RLS becomes particularly relevant.

In this paper, we highlight that the minimum-ℓ^2-norm solution obtained by ORFit is one special choice among the set of (infinitely many) solutions that perfectly fit the data and one could potentially select other desired solutions within this set. Central to this point of view is the property of online mirror descent (OMD) in regression problems, which was studied in [4] and [5] for the discrete-time update setting and in [6] for the continuous-time update setting. In the overparameterized/interpolating regime, where multiple parameters are capable of perfectly fitting the data, the choice of the potential function associated with the Bregman divergence defining the mirror map can be used to select a parameter as desired. We emphasize here that the same applies to the one-pass learning setup.

Motivated by these observations, this study aims to generalize ORFit to achieve optimality of the learned parameters with respect to a general strictly convex function, subsuming ℓ^2 norm as a special case. To this end, we extend ORFit to perform Bregman projection onto the feasible set at each step while retaining the capability to learn from a sequence of data without retraining on previous datapoints. Moreover, we expand the model class to accommodate multi-dimensional output scenarios.

More specifically, this paper presents a generalized al-

Namhoon Cho and Hyo-Sang Shin are with the Centre for Autonomous and Cyber-Physical Systems, School of Aerospace, Transport and Manufacturing, Cranfield University, Cranfield, Bedfordshire, MK43 0AL, United Kingdom. e-mail: {n.cho, h.shin}@cranfield.ac.uk

Hyo-Sang Shin is with Cho Chun Shik Graduate School of Mobility, Korea Advanced Institute of Science and Technology, Daejeon, 34051, South Korea. e-mail: eshy@kaist.ac.kr

Youngjae Min and Navid Azizan are with the Laboratory for Information & Decision Systems, Massachusetts Institute of Technology, Cambridge, MA, 02139, United States. e-mail: {yjm, azizan}@mit.edu

gorithm named Projected Orthogonal Recursive Fitting (Π-ORFit). The contributions of this study are twofold:

- We introduce a nullspace projector matrix formalism to derive a one-pass learning algorithm for overparameterized linear models. The matrix-based approach fully characterizes the set of solutions and extends ORFit to vector-output models.
- We introduce Bregman projection into the update rule to select a desired parameter vector among the feasible set. The resulting parameter minimizes the strictly convex potential function generating the Bregman divergence, which includes the ℓ^2 norm as a special case.

The rest of the paper is organized as follows: Section II presents Π-ORFit, with the Bregman projection step considering overparameterized linear models. The analysis in Sec. III shows how the local optimality attained at each step induces the global optimality for the optimization problem for the entire data. Section IV presents a numerical experiment with a synthetic dataset in the overparameterized regime to investigate how learned parameters vary along with the choice of the potential function for Bregman projection. Section V summarizes the concluding remarks.

A. Related Work

One-pass learning refers to the setting wherein the learner endeavors explicitly to preserve the model predictions on previous data while learning a single new datapoint at a time [1]. Here, the new data can originate from a different distribution than that of the previously seen data. In this sense, one-pass learning can be distinguished from *online learning* and *incremental learning*, which typically consider points or batches of data arriving sequentially from the same distribution. The related notion of *continual learning* refers to learning from the sequentially presented batches of data from different tasks [2]. The definition of *one-pass learning* is slightly different from *single-pass online learning*, which only aims to match the prediction accuracy of batch learners without executing multiple passes through the same training data [7]–[10].

Mitigation of catastrophic forgetting has been a central direction in the studies on one-pass learning and continual learning [11], [12]. Several approaches have been developed to maintain the model's prediction on previously experienced data. A naive approach is to expand the network architecture to hold more information. The method in [13] increases the width of a radial basis function network if necessary to incorporate new datapoints. However, the amount of computational resources required may grow prohibitively large to perform one-pass learning. Similarly, the ensembling approach [14] to train multiple models and combine their outputs to make predictions is deemed inappropriate for the purpose of one-pass learning.

Alternatively, memory-aware approaches store important datapoints in replay buffers for rehearsal. Experience replay schemes were investigated in [15], [16] for preventing catastrophic forgetting while mainly concerning memory efficiency. Gradient episodic memory ensures that any update does not incur an increase in the loss on previous data. It was utilized as a means to retain the memory of previously visited areas in the lifelong navigation framework in [17]. The fast-slow dual-memory approach shares a similar philosophy with the replay methods [18]. The replay approach typically is more about the design of architectures than developing a new training algorithm for one-pass learning.

On the other hand, regularization-based approaches confine the parameter update through explicit regularization so that the predicted outputs remain close to their previously learned values [19]–[22]. In [21], regularization is performed in function space with representative past datapoints. The function space regularization was combined with parameter space regularization in [22] to take advantage of both approaches. These approaches utilize explicit regularization to implicitly steer the parameter update direction from the greedy one that only considers the current datapoint. This is in contrast to ORFit [1] and OGD [2] which determine the update direction explicitly and result in an implicit regularization effect.

II. Projected Orthogonal Recursive Fitting for One-Pass Learning

This section presents Π-ORFit, which introduces Bregman projection as inspired by OMD for one-pass learning with the flexibility to adjust the solution properties. The proposed approach introduces a formalism based on the nullspace projector matrix. Perfect fitting in the sense of preserving the output of already experienced datapoints can be achieved without retraining on the past history.

A. Problem Formulation

Consider a linear model given by

$$y\left(t\right) = \Phi\left(x\left(t\right)\right)\theta, \tag{1}$$

where $t \in \mathbb{R}$ denotes the index variable (which is not necessarily the wall-clock time), $y \in \mathbb{R}^{n\times 1}$ denotes the output vector, $\Phi\left(x\right) \in \mathbb{R}^{n\times q}$ denotes the feature (basis) matrix that can be a known nonlinear function of some input variable x, and $\theta \in \mathbb{R}^{q\times 1}$ denotes the parameter vector. We will refer to the variables entering into the relation of Eq. (1) at t_k with the shorthand notation $y_k := y\left(t_k\right)$ and $\Phi_k := \Phi\left(x\left(t_k\right)\right)$. Also, we will use θ_k to refer to the parameter estimate (solution) obtained at step k.

We are interested in the online learning scenario in which one datapoint is presented at each step sequentially. We are particularly interested in the case of overparameterized (or underdetermined) scenarios, where q is much greater than the number of available datapoints (samples) so that θ is not uniquely determined. The redundancy in the parameters allows multiple interpolating solutions that perfectly fit the given datapoints. With that in mind, suppose that θ_k exactly solves

$$y_j = \Phi_j\theta_k,\ \forall j = 1,\ldots,k, \tag{2}$$

at step $k \geq 1$. Vertical concatenation of the relations in Eq. (2) gives a matrix expression representing the perfect fitting

condition up to step k as

$$y_{1:k} =: \begin{bmatrix} y_1 \\ \vdots \\ y_k \end{bmatrix} = \begin{bmatrix} \Phi_1 \\ \vdots \\ \Phi_k \end{bmatrix} \theta_k := \Phi_{1:k}\theta_k, \tag{3}$$

with the matrices $y_{1:k}$ and $\Phi_{1:k}$ defined accordingly.

B. Recursive Update of Parameter Estimate

Provided θ_k that solves Eq. (3), we now want to update the parameter estimate recursively at the current step $k+1$ by

$$\theta_{k+1} = \theta_k + \Delta_k, \tag{4}$$

with a suitably designed parameter update step Δ_k. For the purpose of one-pass regression, the output values encountered in the past, i.e., up to step k, should not be altered even after performing the update at the current step $k+1$. At the same time, the updated parameter estimate θ_{k+1} should also fit the current datapoint perfectly. These two requirements lead to the extended perfect fitting condition for step $k+1$, which can be written as

$$\begin{bmatrix} y_{1:k} \\ y_{k+1} \end{bmatrix} = \Phi_{1:k+1}\theta_{k+1} = \begin{bmatrix} \Phi_{1:k} \\ \Phi_{k+1} \end{bmatrix} (\theta_k + \Delta_k). \tag{5}$$

The constraint given by Eq. (5) can be rewritten as

$$\begin{bmatrix} 0 \\ y_{k+1} - \Phi_{k+1}\theta_k \end{bmatrix} = \begin{bmatrix} \Phi_{1:k} \\ \Phi_{k+1} \end{bmatrix} \Delta_k. \tag{6}$$

The past output perfect fitting constraint represented by the upper block in Eq. (6) indicates that the update step Δ_k should lie in the nullspace of $\Phi_{1:k}$. Let

$$P_k := I - \Phi_{1:k}^{\dagger}\Phi_{1:k} \tag{7}$$

denote the orthogonal projector onto the nullspace of the concatenated basis matrix $\Phi_{1:k}$, i.e., the nullspace projector, where $()^{\dagger}$ denotes the Moore-Penrose generalized inverse of a matrix. Then, Eq. (6) requires the parameter update step to be represented as

$$\Delta_k = P_k w_k, \tag{8}$$

for some $q \times 1$ vector w_k.

Next, we can determine the parameter update step by solving the current output perfect fitting constraint represented by the lower block in Eq. (6) for w_k. The affine equality constraint in w_k constitutes a system with fewer equations than variables, leading to the infinitude of possible solutions if any exist. Assuming $\Phi_{k+1}P_k \neq 0$ and the existence of solutions, the entire solution set can be represented as

$$w_k = w_k^{\circ} + Q_k v_k, \tag{9}$$

where $v_k \in \mathbb{R}^{q\times 1}$ denotes a vector of free parameters, and

$$\begin{aligned} w_k^{\circ} &= (\Phi_{k+1}P_k)^{\dagger}(y_{k+1} - \Phi_{k+1}\theta_k) \\ Q_k &= I - (\Phi_{k+1}P_k)^{\dagger}\Phi_{k+1}P_k. \end{aligned} \tag{10}$$

Note that $P_k(\Phi_{k+1}P_k)^{\dagger} = (\Phi_{k+1}P_k)^{\dagger}$ thus $P_k w_k^{\circ} = w_k^{\circ}$ since P_k is symmetric and idempotent, and $Q_k w_k^{\circ} = 0$.

Substituting Eqs. (8) and (9) into Eq. (4) leads to the parameter update law that can be represented as

$$\theta_{k+1} = \theta_k + P_k(w_k^{\circ} + Q_k v_k) = \theta_k + w_k^{\circ} + P_k Q_k v_k. \tag{11}$$

The parameter update law given by Eq. (11) clearly takes a recursive form, if the nullspace projector P_k also has a recursive structure.

One way to uniquely determine v_k is to choose θ_{k+1} to be close to θ_k in a particular sense, i.e., solve the unconstrained optimization problem

$$\mathcal{P}_{k+1}^{v}: \quad \underset{v}{\text{minimize}}\ D_{\rho}(\theta_k + w_k^{\circ} + P_k Q_k v, \theta_k) \tag{12}$$

where

$$D_{\rho}(x, y) := \rho(x) - \rho(y) - \langle \nabla\rho(y), x - y\rangle \tag{13}$$

is the Bregman divergence defined with a strictly convex and differentiable potential function $\rho : \mathbb{R}^{q\times 1} \to \mathbb{R}$. This is equivalent to performing the Bregman projection defined by the constrained optimization problem

$$\mathcal{P}_{k+1}^{\theta}: \quad \begin{aligned} &\underset{\theta}{\text{minimize}} && D_{\rho}(\theta, \theta_k) \\ &\text{subject to} && y_{1:k+1} = \Phi_{1:k+1}\theta. \end{aligned} \tag{14}$$

That is, θ_{k+1} given by substituting the solution v_k of $\mathcal{P}_{k+1}^{v}$ into Eq. (11) will be the same as the solution of $\mathcal{P}_{k+1}^{\theta}$.

It will be shown later in Sec. III that solving the local problem $\mathcal{P}_{k+1}^{\theta}$ at each step induces the global optimality of the learned parameter with respect to a certain objective function, namely $\rho(\theta)$.

Remark 1. $\mathcal{P}_{k+1}^{v}$ is a convex optimization problem since the Bregman divergence defined in Eq. (13) is convex in its first argument. For a particular case when $\rho(\xi) = \frac{1}{2}\|\xi\|_2^2$, we have the closed-form solution $w_k = w_k^{\circ}$ with the free parameter $v_k = 0$. It is evident from Eq. (11) that Π-ORFit reduces to the ORFit developed in [1] in this case. For other choices of differentiable $\rho(\cdot)$, $\mathcal{P}_{k+1}^{v}$ can be solved numerically, e.g., using quasi-Newton methods.

C. Recursive Update of Nullspace Projector with Memory Restriction

Let us consider the nullspace projector P_{k+1} that is related to the concatenated basis matrix $\Phi_{1:k+1}$ at the current step $k+1$. At the next step $k+2$, the calculation of the parameter update step Δ_{k+1} should satisfy the perfect fitting constraint given in the same form as in Eq. (6) but for step $k+2$. In other words, we will need P_{k+1} to satisfy

$$\Phi_{1:k+1}P_{k+1} = \begin{bmatrix} \Phi_{1:k} \\ \Phi_{k+1} \end{bmatrix} P_{k+1} = 0. \tag{15}$$

Once the concatenated basis matrix $\Phi_{1:k}$ is populated to be of full rank at some step k, its nullspace becomes an empty set, i.e., $P_k = 0$, meaning that it is no longer possible to update the parameter to fit the new data point from step $k+1$ without forgetting. A naive solution is to increase q, however, this approach will be unrealistically expensive and make the algorithm unsuitable for real-time operation. In practice, the finite size of memory places a limit on the number of bases

that can be kept while running. In the presence of memory restriction, it is impossible to maintain perfect fitting over all previously seen data points through enforcing Eq. (15).

A reasonable modification to reduce memory burden is to perform dimension reduction based on limited-rank approximation of $\Phi_{1:k}$ to incrementally build the summary of bases $\widehat{\Phi}_{1:k}$. The Incremental Principal Component Analysis (IPCA) technique was adopted for the same purpose in [1] where the case of $n = 1$ was considered. This study presents an extended method similar to the chunk IPCA of [23], [24] to deal with the multi-dimensional output.

Let m represent the memory size and $\widehat{\Phi}_{1:k} = \widehat{U}_k \widehat{\Sigma}_k \widehat{V}_k^T \in \mathbb{R}^{\min(m,nk)\times q}$ represent the economy-sized Singular Value Decomposition (SVD) of the summary of bases. The orthogonal projector onto the nullspace of $\widehat{\Phi}_{1:k}$ can be constructed as

$$P_k = I - \widehat{\Phi}_{1:k}^{\dagger} \widehat{\Phi}_{1:k} = I - \widehat{V}_k \widehat{\Sigma}_k^{\dagger} \widehat{\Sigma}_k \widehat{V}_k^T = I - \widehat{V}_k^r \widehat{V}_k^{rT}, \quad (16)$$

where r is the number of nonzero singular values on the diagonal of $\widehat{\Sigma}_k$ and $\widehat{V}_k^r$ is the first r columns of $\widehat{V}_k$. Equation (16) shows that only the right singular vectors in $\widehat{V}_k$ are relevant to the construction of P_k.

Once $\Phi_{k+1} \in \mathbb{R}^{n\times q}$ is presented at step $k+1$, the SVD of $\begin{bmatrix} \widehat{\Phi}_{1:k} \\ \Phi_{k+1} \end{bmatrix} = \widetilde{U}_{k+1} \widetilde{\Sigma}_{k+1} \widetilde{V}_{k+1}^T$ can be obtained by repeating rank-1 update of SVD n times by appending each row of Φ_{k+1} at a time. Algorithm 1 describes the rank-1 update procedure for row vector addition [25]. Finally, when the number of rows in $\widetilde{\Sigma}_{k+1}$ exceeds m, low-rank approximation can be performed by defining $\widehat{\Sigma}_{k+1}$ to be first $m \times m$ block of $\widetilde{\Sigma}_{k+1}$ and $\widehat{V}_{k+1}$ to be the first m columns of $\widetilde{V}_{k+1}$.

Algorithm 1 Rank-1 Update of SVD for Appending Row

Given: $X = USV^T$

Desired: $\begin{bmatrix} X \\ c^T \end{bmatrix} = \widetilde{U}\widetilde{S}\widetilde{V}^T$

Input: S, V, c

Output: $\widetilde{S}$, $\widetilde{V}$

$q \leftarrow \left(I - VV^T\right) c$

$K \leftarrow \begin{bmatrix} S & 0 \\ c^T V & \|q\|_2 \end{bmatrix}$

$\left[\sim, \widetilde{S}, V_K\right] \leftarrow \mathsf{svd}\left(K,\ \text{`econ'}\right)$

$\widetilde{V} \leftarrow \left[V, \frac{q}{\|q\|_2}\right] V_K$

return $\widetilde{S}$, $\widetilde{V}$

D. Summary

In summary, Π-ORFit for linear models consists of two update rules: one for the parameter in Eq. (11), and another for the nullspace projector in Eq. (16) with repeated rank-1 update process. Full determination of the update step in Eq. (8) requires solving $\mathcal{P}_{k+1}^v$ in Eq. (12). Algorithm 2 provides the pseudocode for Π-ORFit. The algorithm can be initialized by specifying

$$\theta_0 \in \mathbb{R}^{q\times 1}, \quad S_0 = [\,], \quad V_0 = [\,], \quad (17)$$

where $[\,]$ denotes the empty array.

Algorithm 2 Π-ORFit (Memory-Aware Update)

Input: θ_k, $\widehat{\Sigma}_k$, $\widehat{V}_k$, y_{k+1}, Φ_{k+1}, m, $\rho(\cdot)$, $\theta_0 = \arg\min_{\xi} \rho(\xi)$

Output: θ_{k+1}, $\widehat{\Sigma}_{k+1}$, $\widehat{V}_{k+1}$

if $\mathsf{isempty}\left(\widehat{\Sigma}_k\right)$ **then**

$P_k \leftarrow I$

else

$\widehat{V}_k^r \leftarrow \widehat{V}_k\left(:, 1{:}\,\mathrm{rank}\left(\widehat{\Sigma}_k\right)\right)$

$P_k \leftarrow I - \widehat{V}_k^r \widehat{V}_k^{rT}$

end if

$w_k^{\circ} \leftarrow \left(\Phi_{k+1} P_k\right)^{\dagger} \left(y_{k+1} - \Phi_{k+1}\theta_k\right)$

$Q_k \leftarrow I - \left(\Phi_{k+1} P_k\right)^{\dagger} \Phi_{k+1} P_k$

$v_k \leftarrow \arg\min_v D_\rho\left(\theta_k + w_k^{\circ} + P_k Q_k v, \theta_k\right)$

$\theta_{k+1} \leftarrow \theta_k + w_k^{\circ} + P_k Q_k v_k$

$(S, V) \leftarrow \left(\widehat{\Sigma}_k, \widehat{V}_k\right)$

for $i = 1 : \dim(y_{k+1})$ **do**

$[S, V] \leftarrow \mathsf{svdup}\left(S, V, \Phi_{k+1}(i,:)^T\right)$ (Algorithm 1)

end for

$\left(\widehat{\Sigma}_{k+1}, \widehat{V}_{k+1}\right) \leftarrow (S, V)$

if $\dim\left(\mathrm{diag}\left(\widehat{\Sigma}_{k+1}\right)\right) > m$ **then**

$\left(\widehat{\Sigma}_{k+1}, \widehat{V}_{k+1}\right) \leftarrow \left(\widehat{\Sigma}_{k+1}(1{:}m, 1{:}m), \widehat{V}_{k+1}(:, 1{:}m)\right)$

end if

return θ_{k+1}, $\widehat{\Sigma}_{k+1}$, $\widehat{V}_{k+1}$

Remark 2. Π-ORFit can be applied to online learning done in two different manners. In the first case, y_k represents the vector-valued label for a single datapoint to be processed at step k. In the other case, y_k may represent the collection of labels for a set of datapoints to be processed at step k.

Remark 3. The algorithm does not apply to the case when $\Phi_{k+1} P_k = 0$, which would happen when all rows of Φ_{k+1} lie in the subspace spanned by the rows of $\Phi_{1:k}$. The possibility of degeneracy demands a modification to the algorithm or a datapoint selection mechanism to avoid such an issue.

III. Analysis of Induced Global Optimality

This section shows that the parameter returned by Π-ORFit minimizes a strictly convex objective function quantifying a distance from the initial value. The optimality analysis generalizes the connection between the ℓ^2-regularized RLS and ORFit highlighted in [1]. Theorem 1 states the induced global optimality of the learned parameter for the interval before $\Phi_{1:k}$ becomes full rank; see the appendix for its proof.

Theorem 1. *Consider a sequence of optimization problems described by*

$$\mathcal{P}_i^{\theta}: \quad \begin{array}{ll} \underset{\theta}{\textit{minimize}} & D_\rho(\theta, \theta_{i-1}) \\ \textit{subject to} & l_i(\theta) := \Phi_{1:i}\theta - y_{1:i} = 0 \end{array} \quad (18)$$

for $i \geq 1$, where $\rho(\cdot)$ is a strictly convex function, $D_\rho(\cdot,\cdot)$ is the Bregman divergence defined in Eq. (13)*, and θ_j denotes*

the optimal solution of $\mathcal{P}_j^\theta$ with $\theta_0 = \arg\min_\theta \rho(\theta)$. Then, for any $k \geq 1$ such that the equation $l_k(\theta) = 0$ is consistent and underdetermined, θ_k also solves the following problem.

$$\mathcal{P}_k^{\theta'}: \quad \begin{matrix} \underset{\theta}{\textit{minimize}} & \rho(\theta) \\ \textit{subject to} & l_k(\theta) = 0 \end{matrix} \tag{19}$$

Remark 4. The global optimality induced by the local geometry of the fitting process resembles the implicit regularization of mirror descent studied in [4], [5]. However, Π-ORFit itself is different from OMD as it does not involve explicit transfer between the primal and dual spaces through the mirror map.

IV. Numerical Experiments

This section presents a numerical example to show how the potential function defining the Bregman projection step affects the sequence of perfect fitting parameters while updating in the overparameterized regime. The purpose of the computational experiment is to verify the theoretical analysis in Sec. III and to demonstrate the practical significance of Π-ORFit.

A. Setup

This experiment considers the problem of online sparse regression which frequently appears in the context of model discovery or dictionary learning using online streamed data for adaptive prediction and control purposes. The usual approach for solving a sparse regression problem in batch mode proceeds by first specifying the model with a large number of basis function candidates and then minimizing an ℓ^1-regularized loss function. In this problem, the true parameter is assumed to be a sparse vector.

To simulate the scenario which aims to learn a sparse vector $\theta^* \in \mathbb{R}^{q\times 1}$ using an overparameterized model, a synthetic dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ consisting of input $x_i \in \mathbb{R}^{m\times 1}$ and output $y_i \in \mathbb{R}^{n\times 1}$ is generated as follows:

(1) Generate $m \times N$ matrix X of normally distributed random numbers with mean 0 and standard deviation 1.
(2) For $j = 1, \ldots, n$, generate $m \times \lceil \frac{q}{2} \rceil$ matrix Ω_j of uniformly distributed random integers in $[1, q]$.
(3) Define $n \times q$ matrix-valued function representing random Fourier features by

$$\Phi(x) := \begin{bmatrix} \cos\left(x^T\Omega_1\right) & \sin\left(x^T\Omega_1\right) \\ \vdots & \vdots \\ \cos\left(x^T\Omega_n\right) & \sin\left(x^T\Omega_n\right) \end{bmatrix}$$

with element-wise application of $\cos(\cdot)$ and $\sin(\cdot)$ to the vectors $x^T\Omega_j$ for $j = 1, \ldots, n$.
(4) Generate $q \times 1$ sparse vector θ^* by randomly specifying N_{nzero} elements with each integer in $[1, N_{nzero}]$ and all other elements with zero.
(5) For $i = 1, \ldots, N$, construct the input-output pair by taking $x_i = X[:, i]$ and computing $y_i = \Phi(x_i)\theta^*$.

A linear model with q-dimensional parameter vector can perfectly fit up to q scalar elements of output data assuming linear independence of the set of associated basis vectors. Therefore, the number of datapoints is set to be $N = \lfloor \frac{q}{n} \rfloor$ as the present study focuses on the behavior of parameters updated by Π-ORFit in the overparameterized regime.

This experiment considers the family of potential functions given by the p-th power of ℓ^p-norm over p for $p > 1$, i.e., $\rho(\theta) = \frac{1}{p}\|\theta\|_p^p$, which is the function class considered in [5], [26]. The experiment compares the results obtained with various p by repeating the training process using Π-ORFit on a fixed dataset. The software implementation of the Bregman projection step adopts an unconstrained optimization solver based on the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm. Table I summarizes the parameters related to the experiment setup.

TABLE I: Experiment Setup Parameters

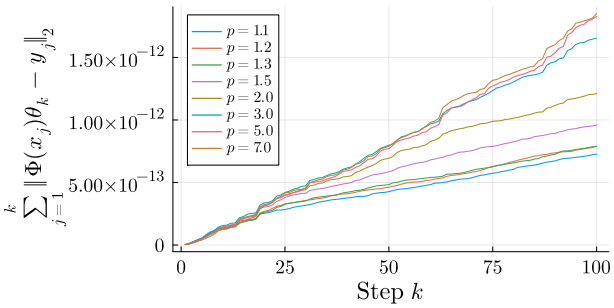
Quantity	Value
m	10
n	3
q	300
θ_0	0
$\rho(\theta)$	$\frac{1}{p}\|\theta\|_p^p$
p	{1.1, 1.2, 1.3, 1.5, 2, 3, 5, 7}
N_{nzero}	10

B. Experimental Results and Discussion

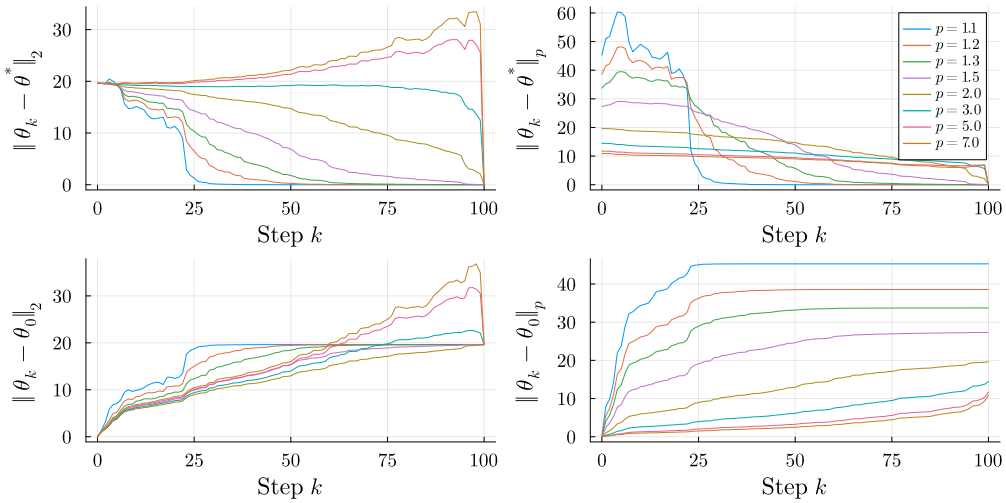
The results show consistent trends in the learned parameters depending on the choice of p within the overparameterized regime.

1) Forgetting: Output Prediction Error: Figure 1a shows the sum of the prediction error over all datapoints presented to the learner up to each training step k which can be written as $\sum_{j=1}^k \|\Phi(x_j)\theta_k - y_j\|_2$. The order of accumulated prediction error evaluated at each training step indicates that the amount of forgetting during the online learning process is negligible for all values of p being tested. The result clearly shows the multiplicity of the sequence of perfect-fitting solutions in the overparameterized regime that all preserve the prediction of the learned model at the past-experienced datapoints. The choice of the potential function $\rho(\theta)$ determines a particular solution among many data-consistent candidates.

2) Identification: Parameter Estimation Error: The first row of Fig. 1b shows the magnitude of parameter estimation error defined by $\theta_k - \theta^*$ as a function of training progress. The error is quantified in terms of ℓ^p-norms for $p = 2$ and the p value used in each case to define the Bregman projection step of Π-ORFit. The parameter estimation error becomes 0 at step $N = 100$ in all cases since the system of associated equations admits a unique solution as the number of fitting constraints reaches the dimension of the parameter vector. The influence of different choices for p on the parameter estimation error is the most prominent in the pattern of convergence to 0. Earlier convergence of the parameter estimation error to 0 is achieved with smaller p. Also, the difference in the norm of parameter estimation error between two consecutive training steps, which can be



(a) Sum of prediction error over all previously experienced datapoints. The model's prediction is preserved on past datapoints.



(b) Parameter estimation error (1st row) and deviation from initial value (2nd row). Smaller p results in earlier and smoother convergence. Large p leads to abrupt correction as soon as the parameter can be uniquely determined.

Fig. 1: Evolution of output prediction and parameter identification accuracy indicators.

expressed as $\|\theta_{k+1} - \theta^*\| - \|\theta_k - \theta^*\|$, converges to 0 as $k \to N$ for $p < 2$. In contrast, using a larger p tends to delay the regulation of parameter estimation error until the associated system of linear equations finally becomes fully determined. The learned parameter almost jumps to the true parameter at training step N for $p > 2$.

The result supports the potential benefits of Π-ORFit employing a sparsity-promoting potential function when the true parameter vector is believed to be sparse. One advantage is the fast identification of the true parameter even before acquiring the minimum number of datapoints to render the system of equations fully determined. This can enable rapid model discovery in the early phase of learning-based task operation. Another benefit is the smooth convergence of the parameter estimation error to 0 for $p < 2$. This is particularly desirable to enable bumpless transition in control applications by preventing abrupt responses that might take place at the boundary of switching from the underdetermined to the overdetermined regime.

3) Regularization: Deviation of Parameter from Initial Value: Figure 2 shows the magnitude of the deviation of the parameter from its initial value given by $\theta_k - \theta_0$ with respect to various norms. In Fig. 2, the case of choosing the hyperparameter $p = r$ of the potential function family attains the minimum of norm $\|\theta_k - \theta_0\|_r$ at each step. This observation is consistent with the analysis of Theorem 1.

4) Sparsity: Absolute Value of Parameters: Theorem 1 implies that performing Bregman projection at each step with the p-th power of ℓ^p-norm as the potential function will provide the capability to adjust the sparsity of learned parameter vector. Figure 3 shows that, while staying in the overparameterized regime, a larger p tends to spread the frequency distribution across a wide range of absolute values as k increases, whereas a small p leaves only a few nonzero parameters.

V. Conclusion

We proposed the Projected Orthogonal Recursive Fitting algorithm (Π-ORFit) for one-pass learning with overparameterized linear models as a generalization of ORFit by introducing Bregman projection into the method. While Π-ORFit perfectly fits the datapoints over a single-pass sweep without catastrophic forgetting, it can select from a range of interpolating solutions which minimize different potential functions. Further, the approach based on the nullspace

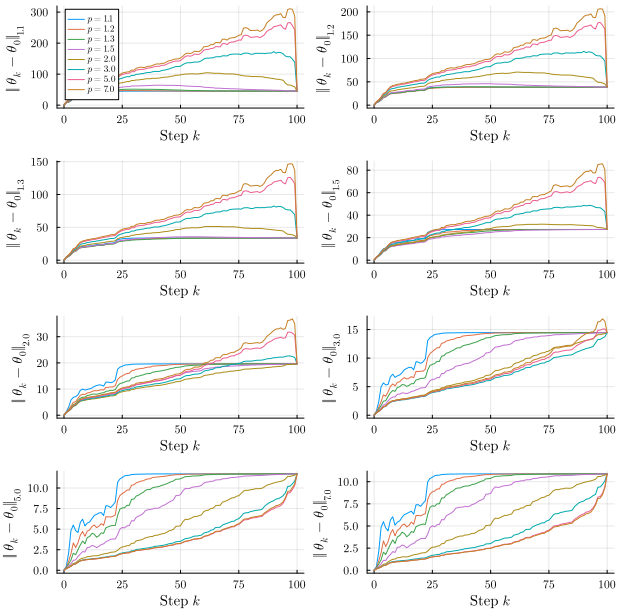


Fig. 2: Deviation of the parameter from the initial value. θ_k minimizes the potential function chosen to define Bregman projection.

projector enables the algorithm to learn models with vector outputs. A numerical example for an online sparse regression problem confirmed the theoretical findings and supported the practical usefulness of Π-ORFit.

REFERENCES

[1] Y. Min, K. Ahn, and N. Azizan, "One-Pass Learning via Bridging Orthogonal Gradient Descent and Recursive Least-Squares," in *61st IEEE Conference on Decision and Control*, Cancun, Mexico, December 2022.

[2] M. Farajtabar, N. Azizan, A. Mott, and A. Li, "Orthogonal Gradient Descent for Continual Learning," in *23rd International Conference on Artificial Intelligence and Statistics*, August 2020, pp. 3762–3773.

[3] P. Zhao, X. Wang, S. Xie, L. Guo, and Z.-H. Zhou, "Distribution-Free One-Pass Learning," *IEEE Transactions ond Knowledge and Data Engineering*, vol. 33, no. 3, pp. 951–963, 2021.

[4] N. Azizan and B. Hassibi, "Stochastic Gradient/Mirror Descent: Minimax Optimality and Implicit Regularization," in *7th International Conference on Learning Representations*, New Orleans, LA, USA, May 2019.

[5] N. Azizan, S. Lale, and B. Hassibi, "Stochastic Mirror Descent on Overparameterized Nonlinear Models," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 7717–7727, 2022.

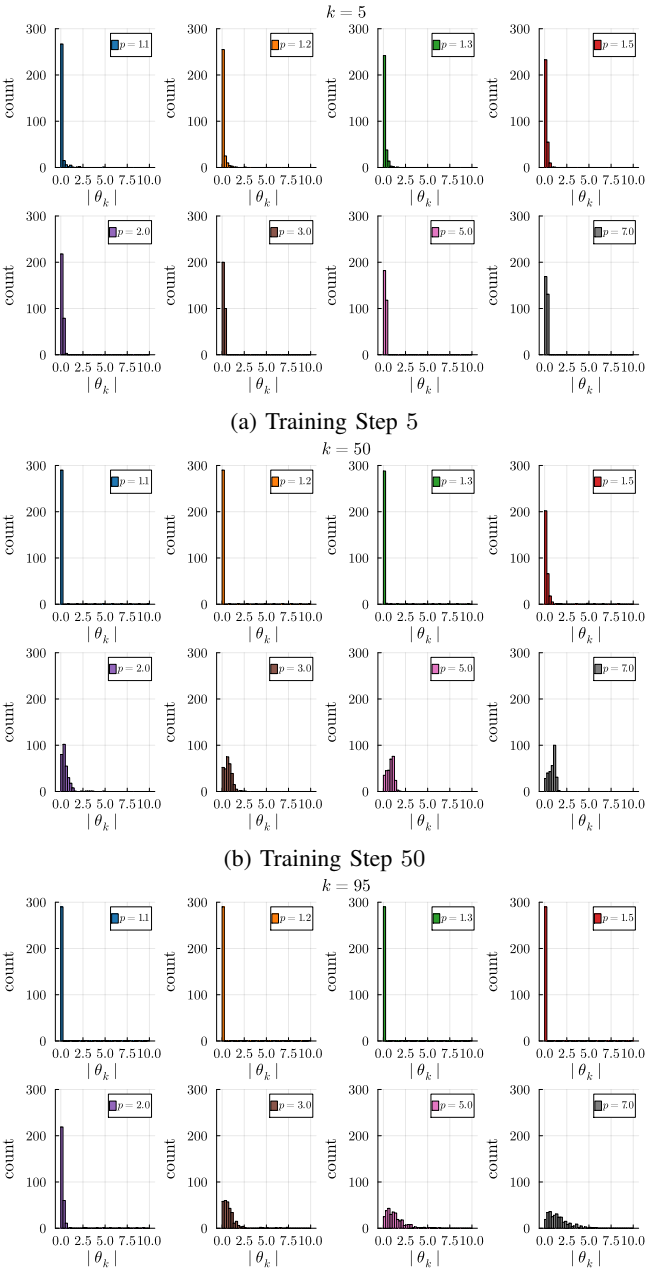
[6] N. M. Boffi and J.-J. E. Slotine, "Implicit Regularization and Momentum Algorithms in Nonlinearly Parameterized Adaptive Control and Prediction," *Neural Computation*, vol. 33, no. 3, pp. 590–673, 2021.

[7] V. R. Carvalho and W. W. Cohen, "Single-Pass Online Learning: Performance, Voting Schemes and Online Feature Selection," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Philadelphia, PA, USA, August 2006, p. 548–553.

[8] Z. You, X. Wang, and B. Xu, "Exploring One Pass Learning for Deep Neural Network Training with Averaged Stochastic Gradient Descent," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Florence, Italy, May 2014, pp. 6854–6858.

[9] Z. Zhou, W.-S. Zheng, J.-F. Hu, Y. Xu, and J. You, "One-Pass Online Learning: A Local Approach," *Pattern Recognition*, vol. 51, pp. 346–357, 2016.

[10] C. Hou and Z.-H. Zhou, "One-Pass Learning with Incremental and Decremental Features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 11, pp. 2776–2792, 2018.



(a) Training Step 5

(b) Training Step 50

(c) Training Step 95

Fig. 3: Snapshots of the histogram for the elementwise absolute value of parameters at training steps $k = 5, 50, 95$. p close to 1 promotes sparsity of the learned parameters.

[11] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A Continual Learning Survey: Defying Forgetting in Classification Tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3366–3385, 2022.

[12] R. Kemker, M. McClure, A. Abitino, T. Hayes, and C. Kanan, "Measuring Catastrophic Forgetting in Neural Networks," in *32nd AAAI Conference on Artificial Intelligence*, New Orleans, LA, USA, February 2018.

[13] M. Thakong, S. Phimoltares, S. Jaiyen, and C. Lursinsap, "One-Pass-Throw-Away Learning Algorithm Based on Hybridization of LDA and PCA," in *International Conference on Information Science and Applications*, Pattaya, Thailand, June 2013, pp. 1–4.

[14] R. Polikar, L. Upda, S. Upda, and V. Honavar, "Learn++: An Incremental Learning Algorithm for Supervised Neural Networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 31, no. 4, pp. 497–508, 2001.
[15] T. L. Hayes, N. D. Cahill, and C. Kanan, "Memory Efficient Experience Replay for Streaming Learning," in *International Conference on Robotics and Automation*, Montreal, QC, Canada, May 2019, pp. 9769–9776.
[16] T. L. Hayes and C. Kanan, "Online Continual Learning for Embedded Devices," 2022, arXiv:2203.10681, Conference on Lifelong Learning Agents. [Online]. Available: https://arxiv.org/abs/2203.10681
[17] B. Liu, X. Xiao, and P. Stone, "A Lifelong Learning Approach to Mobile Robot Navigation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1090–1096, 2021.
[18] R. Kemker and C. Kanan, "FearNet: Brain-Inspired Model for Incremental Learning," in *6th International Conference on Learning Representations*, Vancouver, BC, Canada, April-May 2018. [Online]. Available: https://openreview.net/forum?id=SJ1Xmf-Rb
[19] F. Zenke, B. Poole, and S. Ganguli, "Continual Learning Through Synaptic Intelligence," in *34th International Conference on Machine Learning*, Sydney, Australia, August 2017, pp. 3987–3995. [Online]. Available: https://proceedings.mlr.press/v70/zenke17a.html
[20] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming Catastrophic Forgetting in Neural Networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.
[21] P. Pan, S. Swaroop, A. Immer, R. Eschenhagen, R. Turner, and M. E. E. Khan, "Continual Deep Learning by Functional Regularisation of Memorable Past," in *Conference on Neural Information Processing Systems*, Virtual, December 2020.
[22] M. E. E. Khan and S. Swaroop, "Knowledge-Adaptation Priors," in *35th Conference on Neural Information Processing Systems*, Virtual, December 2021.
[23] S. Ozawa, S. Pang, and N. Kasabov, "An Incremental Principal Component Analysis for Chunk Data," in *IEEE International Conference on Fuzzy Systems*, Vancouver, BC, Canada, July 2006, pp. 2278–2285.
[24] ——, "Incremental Learning of Chunk Data for Online Pattern Classification Systems," *IEEE Transactions on Neural Networks*, vol. 19, no. 6, pp. 1061–1074, 2008.
[25] M. Brand, "Fast Low-Rank Modifications of the Thin Singular Value Decomposition," *Linear Algebra and its Applications*, vol. 415, no. 1, pp. 20–30, 2006.
[26] H. Sun, K. Ahn, C. Thrampoulidis, and N. Azizan, "Mirror Descent Maximizes Generalized Margin and Can Be Implemented Efficiently," in *Conference on Neural Information Processing Systems*, New Orleans, LA, USA, December 2022.

APPENDIX I
PROOF OF THEOREM 1

Proof. By introducing the Lagrange multiplier λ_i, let us define the Lagrangian function associated with $\mathcal{P}_k^\theta$ as

$$\begin{aligned}\overline{J}_i(\theta,\lambda) &= J_i(\theta) + \lambda_i^T l_i(\theta) \\ &= D_\rho(\theta,\theta_{i-1}) + \lambda_i^T l_i(\theta).\end{aligned} \tag{20}$$

If θ_i is an optimal solution of $\mathcal{P}_k^\theta$, then the first-order necessary condition for optimality implies the existence of multiplier λ_i such that

$$\begin{aligned}\nabla_\theta \overline{J}_i(\theta,\lambda_i)\big|_{\theta_i} &= \nabla_\theta D_\rho(\theta,\theta_{i-1})|_{\theta_i} + {\lambda_i}^T \nabla_\theta l_i(\theta_i) \\ &= 0\end{aligned} \tag{21}$$

$$\nabla_\lambda \overline{J}_i(\theta_i,\lambda)\big|_{\lambda_i} = l_i(\theta_i) = 0. \tag{22}$$

To facilitate further analysis, let us recursively define an auxiliary function as

$$\begin{aligned}&L_0(\theta) = \rho(\theta) \\ &L_i(\theta) = L_{i-1}(\theta) + {\lambda_i}^T l_i(\theta), \quad i = 1,2,\cdots.\end{aligned} \tag{23}$$

Note that $L_i(\theta)$ is a strictly convex function in θ since $\rho(\theta)$ is strictly convex and $l_i(\theta)$ is affine. Since a Bregman divergence is invariant under the addition of an affine function to the potential function by definition, we have

$$D_\rho(\theta,\theta_{i-1}) = D_{L_j}(\theta,\theta_{i-1}), \quad j = 0,1,\cdots. \tag{24}$$

Considering Eqs. (23) and (24), Eq. (21) can be rewritten as

$$\begin{aligned}\nabla_\theta \overline{J}_i(\theta,\lambda_i)\big|_{\theta_i} &= \nabla_\theta D_{L_{i-1}}(\theta,\theta_{i-1})\big|_{\theta_i} \\ &\quad + \nabla_\theta L_i(\theta_i) - \nabla_\theta L_{i-1}(\theta_i) \\ &= \nabla_\theta L_{i-1}(\theta_i) - \nabla_\theta L_{i-1}(\theta_{i-1}) \\ &\quad + \nabla_\theta L_i(\theta_i) - \nabla_\theta L_{i-1}(\theta_i) \\ &= \nabla_\theta L_i(\theta_i) - \nabla_\theta L_{i-1}(\theta_{i-1}) \\ &= 0.\end{aligned} \tag{25}$$

A recursive relation arises from Eq. (25) in that

$$\nabla_\theta L_k(\theta_k) = \nabla_\theta L_i(\theta_i), \quad i = 0,1,\cdots, \tag{26}$$

where θ_k denotes the solution of $\mathcal{P}_k^\theta$ which satisfies Eqs. (21)-(22) for $i = k$.

Since θ_0 minimizes $L_0(\theta) = \rho(\theta)$, $\nabla_\theta L_0(\theta_0) = 0$. Then, the recursion relation of Eq. (26) indicates that $\nabla_\theta L_k(\theta_k) = 0$. The critical point θ_k should minimize $L_k(\theta)$, because $L_k(\theta)$ is a strictly convex function and the optimal solution minimizing a strictly convex objective function is unique. (Note that $\nabla l_i(\theta)$ does not depend on θ as $l_i(\theta)$ is affine in θ.)

Unrolling the recursion in Eq. (23) yields

$$\begin{aligned}L_k(\theta) &= \rho(\theta) + \sum_{i=1}^{k} {\lambda_i}^T l_i(\theta) \\ &= \rho(\theta) + \sum_{i=1}^{k} {\lambda_i}^T E_{i\leftarrow k} l_k(\theta) \\ &:= \rho(\theta) + {\Lambda_k}^T l_k(\theta),\end{aligned} \tag{27}$$

where $E_{i\leftarrow k} = \begin{bmatrix} I_{ni\times ni} & 0_{ni\times n(k-i)}\end{bmatrix}$ represents selection of the first ni rows of the post-multiplied matrix, and $\Lambda_k = \sum_{i=1}^k E_{i\leftarrow k}^T \lambda_i$. By interpreting Λ_k as the Lagrange multiplier associated with affine equality constraint $l_k(\theta) = 0$, we can view $L_k(\theta)$ as the Lagrangian function for the problem $\mathcal{P}_k^{\theta'}$ in Eq. (19) whose solution is given by θ_k.

Therefore, we have characterized the optimality of θ_k with respect to the global problem $\mathcal{P}_k^{\theta'}$, while θ_k is the optimal solution to the local subproblem $\mathcal{P}_k^\theta$ at the same time.

□