Proceedings of Machine Learning Research vol 242:1241–1252, 2024 6th Annual Conference on Learning for Dynamics and Control

A Multi-Modal Distributed Learning Algorithm in Reproducing Kernel Hilbert Spaces

Aneesh Raghavan ANEESH@KTH.SE
DCS Division, KTH, Royal Institute of Technology, Stockholm

Karl Henrik Johansson KALLEJ@KTH.SE
DCS Division, KTH, Royal Institute of Technology, Stockholm

Editors: A. Abate, M. Cannon, K. Margellos, A. Papachristodoulou

Abstract

We consider the problem of function estimation by a multi-agent system consisting of two agents and a fusion center. Each agent receives data comprising of samples of an independent variable (input) and the corresponding values of the dependent variable (output). The data remains local and is not shared with other members in the system. The objective of the system is to collaboratively estimate the function from the input to the output. To this end, we present an iterative distributed algorithm for this function estimation problem. Each agent solves a local estimation problem in a Reproducing Kernel Hilbert Space (RKHS) and uploads the function to the fusion center. At the fusion center, the functions are fused by first estimating the data points that would have generated the uploaded functions and then subsequently solving a least squares estimation problem using the estimated data from both functions. The fused function is downloaded by the agents and is subsequently used for estimation at the next iteration along with incoming data. This procedure is executed sequentially and stopped when the difference between consecutively estimated functions becomes small enough. With respect to the algorithm, we prove existence of basis functions for suitable representation of estimated functions and present closed form solutions to the estimation problems at the agents and the fusion center.

Keywords: Distributed Regression, RKHS, Transfer Operators

1. Introduction

1.1. Motivation

Cyber-physical systems are integrations of computational and physical processes, Derler et al. (2011). The sensors onboard such systems collect heterogeneous observations by measuring different aspects of the system and the environment. The observations or the data collected by them can be used to learn complete (given partial models) or partial models of the system or the environment. The heterogeneity of the data motivates us to consider local processing of the data followed by fusion of the models to obtain a model for the system or environment. In many scenarios, the features defining the model are not precisely known. When the agents learn different models based on different features and the final model is obtained through the fusion of the local models, the final model is potentially robust to variations in features.

Consider the following example from distributed SLAM, see, Chellali et al. (2013), Tian et al. (2022), Lajoie et al. (2020). There are two or more agents in an environment with different onboard sensors whose individual aim during the learning phase is to find a mapping from their true position to the sensor output. During the execution phase, this mapping could be utilized for planning and completion of tasks. During the learning phase, each agent is restricted to survey a certain region of the environment they live in and are aware of the predominant features of the map from the position to the sensor output. By gathering data their aim is to collaboratively find a map from position to sensor output. An algorithm which solves the problem would ideally: (i) use local data to estimate local maps from position to sensor output, maps can be viewed as partial models of the environment; (ii) fuse partial models to obtain a global model of the environment (iii) be iterative with exchange of "models" between the agents with sequential processing of data.

Distributed learning algorithms develop models where multiple instances of the same model are trained using different subsets of the training data set or parallel paths of a single model are trained at multiple nodes using the same or different data sets. Algorithms often focus on parallelizing computing for efficient learning, e.g., Chan et al. (1993), Verbraeken et al. (2020), Peteiro-Barral and Guijarro-Berdiñas (2013). For the fusion of the models, some algorithms use tools from meta-learning. Federated learning has emerged as an efficient approach for distributed learning using heterogeneous data and has found applications in IoT, healthcare etc, e.g. Liu et al. (2022), Nguyen et al. (2021).

Multimodal learning algorithms use data sets obtained using multiple kinds of sensors for training models. For example, images and 3D depth scans can be used for edge detection,and, audio and visual data for speech recognition, see, Baltrušaitis et al. (2018), Ngiam et al. (2011). Lanckriet et al. (2004) study the problem of learning a kernel matrix using data, from the space of kernel matrices generated by linear combinations of known kernel matrices. Following the same idea, multiple kernel learning (MKL) algorithms have been investigated. Multimodal learning using kernel methods has been applied to disease detection (Duan et al., 2012; Liu et al., 2013), sentiment analysis (Poria et al., 2017), emotion recognition (Sikka et al., 2013), etc. Most of these algorithms are centralized, i.e., data from sensors are collected and analyzed simultaneously at one location.

The desired algorithm mentioned above has some properties of existing algorithms in the distributed learning and some from multi-modal learning literature. However, algorithms from neither solve the problem completely. In this paper, we use the term "model" and "knowledge" interchangeably with following reasoning. Formal approaches to study knowledge and its properties are crucial for development of intelligent collaborative multi-agent systems (Rosenschein, 1985). (Dynamic) Epistemic logic is used as a formal language to describe knowledge and learning formally (Van Ditmarsch et al., 2007). Knowledge can be viewed as mapping from the set of events to the set $\{0, 1, \varnothing\}$, i.e., if an event is true, false or its validity is unknown. Given input-output data, the function estimated from it enables us to state which events of the form "input = x and output = y" are true and which are false as long as (x, y) belongs to the domain of the estimated function. In this spirit, we use the term *Knowledge* for the mapping learned and the term *Knowledge Space* for the function space where the agent is learning.

1.2. Problem Considered

The problem considered in this paper is as follows. There are two agents, Agent 1 and Agent 2, receiving data comprising of an independent variable and the corresponding value of a dependent variable. The data is received sequentially, one sample at a time. There exists a fusion center that can communicate with each agent. When an agent transmits to a fusion center, it is referred to as *upload operation*. When the fusion center transmits to the an agent, it is referred to as *download operation*. The data collected by the agents is private to the agents and is not shared with other agents in the system. Agents are allowed to upload and download knowledge to and from the fusion

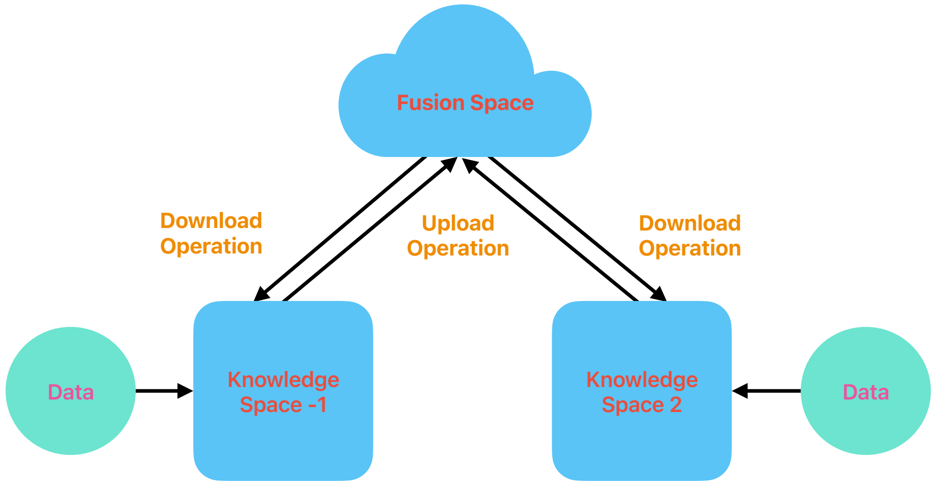

Figure 1: The Learning Architecture

center. Our objective is to develop an iterative collaborative function estimation scheme which eventually converges to an estimate of the mapping from the input to the output.

1.3. Contributions

Our contributions are as follows. The architecture of the learning scheme is as follows. The function estimated at stage n by each agent is uploaded to the fusion center, Figure 1. At the fusion center, the estimated functions are fused using a meta-learning method. The fused function is downloaded by the agents which reflects their final estimate at stage n.

We propose the following learning algorithm for the problem. Given the downloaded function at stage $n-1$ and data point at stage n, each agent estimates the mapping from the input to the output by solving a least-squares regression problem. The estimated functions are uploaded to the fusion center. At the fusion center, the data received by the agents is estimated from the functions received considering that the agents would have performed optimal estimation. Using the estimated data points, a least-squares regression problem is solved to obtain the fused function. The fused function is downloaded on to the knowledge space of each agent. n is incremented by 1 and the sequence is repeated.

With respect to the above algorithm, we: (i) prove the existence of local basis functions used to represent the uploaded and downloaded models; (ii) present closed form solutions to estimation problems at the agents and the regression problem at the fusion center using the basis functions. For the proof of consistency of the learning algorithm and an example demonstrating the same, we refer to Raghavan and Johansson (2024a); these have not been included due to space restrictions. We note that the architecture and the learning algorithm can be extended to any finite number of agents with modifications to the fusion problem. We restrict ourselves to two agents as from conceptual standpoint it is the same as considering more than two agents.

Note that, the data collected by the agents is transformed into the estimated functions and forgotten. At the fusion center, when the data that could have generated these functions is estimated, it is relearned. Our previous work, Raghavan and Johansson (2023), is a special case of the work presented here as in the former, we considered learning in the same knowledge space for the both agents, the estimation problem was a one shot problem and the fusion problem was an optimization problem over linear combinations of the functions estimated by the agents.

1.4. Outline and Notation

The organization of the paper is as follows. In Section 2, we describe the learning architecture, the estimation problems and the learning algorithm. In Section 3, we present the main results including the solution to the estimation problems. We conclude with some comments and future work in Section 4. Notation: for a function $f \in V$, V vector space, we use the notation f when it is treated as a vector and the notation $f(\cdot)$ when it is treated as a function. The null element of a vector space, V, is denoted by θ_V. The span of vectors $\{v_j\}_{j=1}^{j=n} \subset V$ is denoted as $\text{Span}\Big(\{v_j\}_{j=1}^{j=n}\Big)$. The projection onto a subspace $\mathcal{M}$ of a Hilbert space H is denoted by $\Pi_{\mathcal{M}}$. The dual space of a vector space is denoted by V^*.

2. The Distributed Learning System

In this section, we describe the learning system. We begin with the description of the learning architecture, followed by the estimation and fusion problems, and finally present the learning algorithm.

2.1. The Learning Architecture

The learning system comprises of two agents, Agent 1 and Agent 2, and a fusion center. Let $\mathcal{X} \subset \mathbb{R}^I$. The set of features for agent i is a set of continuous functions, $\{\varphi_j^i(\cdot)\}_{j\in\mathcal{I}^i}$, where $\varphi_j^i : \mathcal{X} \to \mathbb{R}$ and $|\mathcal{I}^i| < \infty$. For agent i, we assume that the set of features are linearly independent. Let $K^i(x,y) = \sum_{j\in\mathcal{I}^i} \varphi_j^i(x)\varphi_j^i(y)$ be the kernel for agent i and the corresponding RKHS generated by it be $(H^i, \langle\cdot,\cdot\rangle_{H^i}, ||\cdot||_{H^i})$. Then, the knowledge space constructed for Agent i is the RKHS, H^i, with kernel K^i, Figure 1. Given the knowledge spaces H^1 and H^2 with kernels K^1 and K^2 at Agents 1 and 2 respectively, the fusion space is a RKHS with kernel $K^1 + K^2$, Theorem 6.

Given the downloaded function from iteration $n-1$, $\bar{f}_{n-1}^i$ at Agent i and the data point (x_n^i, y_n^i) at iteration n, each agent solves an estimation problem (subsection 2.2) to arrive at the estimate f_n^i. The locally estimated functions, f_n^1 and f_n^2 are uploaded to the fusion space using the operators, $\hat{L}^1$ and $\hat{L}^2$ respectively, Corollary 7. The uploaded functions are fused in the fusion space (subsection 2.3) to obtain f_n. The fused function is downloaded onto the KS of Agent i using the download operator $\sqrt{\bar{L}^i} \circ \Pi_{\mathcal{N}\left(\sqrt{\bar{L}^i}\right)^\perp}$, Theorem 9, and is denoted as $\bar{f}_n$. $\bar{f}_n$ is considered as the final estimate at the agents at iteration n. The upload and download operators are refereed to as *transfer* operators.

2.2. Estimation at The Agents

In this subsection, we discuss the estimation problem at the agents. At iteration n, given the data point (x_n^i, y_n^i) and the downloaded function from iteration $n-1$, $\bar{f}_{n-1}^i$, the objective of each agent is to minimize the error square between the true output and the estimated output at the received input data point while simultaneously minimizing the norm square between the downloaded function at stage $n-1$ and the current estimate. The latter term represents the complexity of the difference between the two estimates. Thus, the estimation problem for agent i is,

$$(P1)_n^i : \min_{f_n^i \in H^i} C^i(f_n^i),\ C(f_n^i) = (y_n^i - f_n^i(x_n^i))^2 + \varrho_n^i ||f_n^i - \bar{f}_{n-1}^i||_{H^i}^2$$

The above optimization is a trade off between error (difference between estimated output and true output) and complexity of the difference between the estimates at consecutive stages controlled by the ρ_n^i parameter. When ρ_n^i is small the first term gets precedence and when ρ_n^i is large the latter gets precedence.

Algorithm 1 Distributed Learning Algorithm in RKHS

Initialize $\bar{f}_0^1, \bar{f}_0^2, \epsilon, \max, k_{\max}$ where $\max > \epsilon$
$n \leftarrow 0$
while $\max \geq \epsilon$ **do**
    $n \leftarrow n + 1$
    Agent i collects sample $(x_n^i; y_n^i)$.
    Agent i solves optimization problem $(P1)_n^i$ to find f_n^i
    Agent i uploads f_n^i to fusion space.
    Feasibility problem $(P2)_n^i$ is solved find to $\{(\bar{x}_j^i; \bar{y}_j^i)\}_{j=1}^m$, $i = 1, 2$.
    Fusion problem $(P3)_n$ is solved to find f_n.
    f_n is downloaded onto knowledge space of agent i, H^i as $\bar{f}_n^i$.
    if $n \geq k_{\max}$ **then**
        $j \leftarrow 1$
        while $j \leq k_{\max}$ **do**
            temp $= ||\bar{f}^1_{n-k_{\max}+j} - \bar{f}^1_{n-k_{\max}}|| + ||\bar{f}^2_{n-k_{\max}+j} - \bar{f}^2_{n-k_{\max}}||$
            if temp $> \max$ **then**
                $\max \leftarrow$ temp
            end if
            $j \leftarrow j + 1$
        end while
    end if
end while

2.3. Fusion Problem

From Corollary 7, the function uploaded by Agent i to H is f_n^i, with different norm. Invoking Proposition 3, the function uploaded by agent i can be expressed as $f_n^i = \sum_{j=1}^m \alpha_{n,j}^i K^i(\cdot, \bar{x}_j^i)$. Given f_n^1 and f_n^2, the first goal of the fusion center is to estimate the data points $\{(\hat{x}_{n,j}^i, \hat{y}_{n,j}^i)\}_{j=1}^m$ which under optimal estimation would result in the functions f_n^1 and f_n^2 being estimated. We formulate the problem as a feasibility problem:

$$(P2)_n^i : \min_{\{(\hat{x}_{n,j}^i, \hat{y}_{n,j}^i)\}_{j=1}^m} \bar{c}^i \ \text{ s.t } \ f_n^i = \arg\min_{g_n^i \in H^i} \sum_{j=1}^m (\hat{y}_{n,j}^i - g_n^i(\hat{x}_{n,j}^i))^2 + \varrho_n ||g_n^i||^2$$

Given $\{(\hat{x}_{n,j}^1, \hat{y}_{n,j}^1)\}_{j=1}^m \cup \{(\hat{x}_{n,j}^2, \hat{y}_{n,j}^2)\}_{j=1}^m$, the fusion problem defined as,

$$(P3)_n : \min_{f_n \in H} C(f_n), \ C(f_n) = \sum_{i=1,2} \sum_{j=1}^m (\hat{y}_{n,j}^i - f_n(\hat{x}_{n,j}^i))^2 + \varrho_n ||f_n||_H^2,$$

is a least squares regression problem. From Theorem 9, it follows that the downloaded function at Agent i is $\sqrt{L^i} \circ \Pi_{\mathcal{N}\left(\sqrt{L^i}\right)^\perp}(f_n)$. From Proposition 3 it follows that, function downloaded can be expressed as $\bar{f}_n^i = \sum_{j=1}^m \bar{\alpha}_{n,j}^i K^i(\cdot, \bar{x}_j^i)$ which is useful in solving problem $(P1)_n^i$ as demonstrated in Proposition 4.

2.4. The Learning Algorithm

The learning algorithm is described in Algorithm 1. Lines $1-10$ of the algorithm have been described while describing the learning architecture, in subsection 2.1. Lines $11-21$, execute the stopping criterion. Essentially, it is verified that the difference in norm between the downloaded functions at any two iterations in $[n-k_{\max}, n]$ is upper bounded by 2ϵ. Consider $n_1, n_2 \in [n-k_{\max}, n]$.

$$||\bar{f}_{n_1}^1 - \bar{f}_{n_2}^1||_{H^1} + ||\bar{f}_{n_1}^2 - \bar{f}_{n_2}^2||_{H^2} \leq \sum_{i=1,2} ||\bar{f}_{n_1}^i - \bar{f}_{n-k_{\max}}^i|| + ||\bar{f}_{n_2}^i - \bar{f}_{n-k_{\max}}^i|| < 2\epsilon,$$

where the inequality follows from the triangle inequality of norm and Corollary 7. Thus, the stopping criterion approximately verifies that the sequence is Cauchy in norm.

Assumption *We assume that $\mathcal{X}$ is a closed, connected set with no isolated points. We assume that the RKHS H^1 and H^2 are finite dimensional.*

3. Main Results

We begin this section by finding basis functions in local knowledge space of each agent to represent transferred models using the properties of the transfer operators specifically the download operators. Then, we present solutions to the problems formulated in subsections 2.2 and 2.3.

3.1. Basis Functions from Transfer Operators

In this subsection, first we present a simple geometric result pertaining to preserving geometry of Hilbert spaces under suitable transformations and then prove two algebraic results on basis functions of RKHSs.

Proposition 1 *Let $V = V^1 \oplus V^2$ and $U = U^1 \oplus U^2$ be Hilbert spaces with $V^1 = V^{2^\perp}$ and $U^1 = U^{2^\perp}$. If V is isomorphic to U and V^2 is isomorphic to U^2 under the same isomorphism from V to U, then V^1 is isomorphic to U^1.*

Proof Since V is isomorphic to U, there exits $\mathbb{L} : V \to U$ such that $\mathbb{L}$ is a bijection, is linear, and, has a well defined inverse $\mathbb{L}^{-1} : U \to V$ which is also linear. The subspace V^2 of V is isomorphic to some subspace of U, however it is given that it is isomorphic to U^2. Thus $\mathbb{L}(V^2) = U^2$ and $\mathbb{L}^{-1}(U^2) = V^2$. Let $v \in V^1$, i.e, $\Pi_{V^2}(v) = \theta_V$. We claim that $\mathbb{L}(v) \in U^1$. Suppose not. Then, $\Pi_{U^2}(\mathbb{L}(v)) \neq \theta_U$, which implies that $\mathbb{L}^{-1}(\mathbb{L}(v)) = \mathbb{L}^{-1}(\Pi_{U^1}(\mathbb{L}(v)) + \Pi_{U^2}(\mathbb{L}(v))) = \mathbb{L}^{-1}(\Pi_{U^1}(\mathbb{L}(v))) + \mathbb{L}^{-1}(\Pi_{U^2}(\mathbb{L}(v)))$. Since $\mathbb{L}\big|_{V^2}$ is an isomorphism from V^2 to U^2, $\mathbb{L}^{-1}(\Pi_{U^2}(\mathbb{L}(v))) = \mathbb{L}^{-1}\big|_{U^2}(\Pi_{U^2}(\mathbb{L}(v))) \in V^2 \neq \theta_V$. Hence, $\Pi_{V^2}(\mathbb{L}^{-1}(\Pi_{U^2}(\mathbb{L}(v)))) = \mathbb{L}^{-1}\big|_{U^2}(\Pi_{U^2}(\mathbb{L}(v)))$. This implies that,

$$\Pi_{V^2}(v) = \Pi_{V^2}(\mathbb{L}^{-1}(\mathbb{L}(v))) = \Pi_{V^2}(\mathbb{L}^{-1}(\Pi_{U^1}(\mathbb{L}(v)))) + \mathbb{L}^{-1}\big|_{U^2}(\Pi_{U^2}(\mathbb{L}(v))).$$

There exists a unique $u' \in U, u' \neq \theta_U$, such that $\mathbb{L}^{-1}(u') + \mathbb{L}^{-1}(\Pi_{U^2}(\mathbb{L}(v))) = \theta_V$ and $u' \in U^2$. The only way $\Pi_{V^2}(\mathbb{L}^{-1}(\Pi_{U^1}(\mathbb{L}(v)))) + \mathbb{L}^{-1}(\Pi_{U^2}(\mathbb{L}(v))) = \theta_V$ is, if $\Pi_{U^1}(\mathbb{L}(v)) = u' + u''$ where $u' \in U^2$ is defined as before and $u'' \in U^1$ is such that $\mathbb{L}^{-1}(u'') \in V^1$. This is clearly not possible as $U^1 \cap U^2 = \theta_U$ and hence $\Pi_{V^2}(v) \neq \theta_V$ which is clearly a contradiction. Thus, $\mathbb{L}$ maps every

$v \in V^1$ to a unique $u \in U^1$. Similarly, it can be shown that $\mathbb{L}\big|_{V^1}$ is surjective. Since $\mathbb{L}$ and $\mathbb{L}^{-1}$ are linear, V^1 and U^1 are isomorphic under the morphism $\mathbb{L}$. ■

Let $H^1 \times H^2$ be the product space with inner product $\langle (f^1, f^2), (g^1, g^2)\rangle_{H^1\times H^2} = \langle f^1, g^1\rangle_{H^1} + \langle f^2, g^2\rangle_{H^2}$. Let $L : H^1 \times H^2 \to H$, where H is the fusion space (Theorem 6), be a operator defined as $L((f^1, f^2)) = f^1 + f^2$. L is a linear operator and its null space, $\mathcal{N}(L) = \{(f^1, f^2) \in H^1 \times H^2 : f^1 + f^2 = \theta\}$ is a closed subspace as it is finite dimensional. Thus, there exists a unique closed subspace $\mathcal{M}$ such that $H^1 \times H^2 = \mathcal{M} \oplus \mathcal{N}(L)$, where $\mathcal{M} = \mathcal{N}^\perp$. The mapping $L_\mathcal{M} = L \circ \Pi_\mathcal{M}$ (operator L restricted to subspace $\mathcal{M}$) is an isomorphism from $\mathcal{M}$ to H. Let the dimension of $\mathcal{M}$ be m and the dimension of $\mathcal{N}(L)$ be $|\mathcal{I}^1| + |\mathcal{I}^2| - m$. The basis vectors, $\{\varphi^1_j, \theta^2\}_{j\in\mathcal{I}^1} \cup \{\theta^1, \varphi^2_j\}_{j\in\mathcal{I}^2}$, for $H^1 \times H^2$ induce a isomorphism from $H^1 \times H^2$ to $\mathbb{R}^{|\mathcal{I}^1|+|\mathcal{I}^2|}$. Under this isomorphism, $\mathcal{N}(L)$ is isomorphic to

$$\mathcal{N} = \Big\{\Big(\boldsymbol{\alpha^1}, \boldsymbol{\alpha^2}\Big) \in \mathbb{R}^{|\mathcal{I}^1|+|\mathcal{I}^2|} : \sum_{j\in\mathcal{I}^1} \alpha^1_j\varphi^1_j + \sum_{j\in\mathcal{I}^2} \alpha^2_j\varphi^2_j = \theta, \boldsymbol{\alpha^i} = \Big(\alpha^i_1, \ldots, \alpha^i_{|\mathcal{I}^i|}\Big), i = 1, 2\Big\}.$$

Proposition 2 *There exists two sets of m elements each, $\{\bar{x}^i_j\}_{j=1}^m$, $i = 1, 2$ such that (i) $\{\bar{x}^1_j\}_{j=1}^m \cap \{\bar{x}^2_j\}_{j=1}^m = \emptyset$; (ii) $\{K(\cdot, \bar{x}^1_j)\}_{j=1}^m$ and $\{K(\cdot, \bar{x}^2_j)\}_{j=1}^m$, each form a basis for H. Thus, for any function $f \in H$, $\exists!\{\alpha^i_j\}_{j=1}^m$ such that $f(\cdot) = \sum_{j=1}^m \alpha^1_j K(\cdot, \bar{x}^1_j) = \sum_{j=1}^m \alpha^2_j K(\cdot, \bar{x}^2_j)$.*

Proof Let $\mathbb{R}^{|\mathcal{I}^1|+|\mathcal{I}^2|} = \bar{\mathcal{M}} \oplus \mathcal{N}$, where, $\bar{\mathcal{M}} = \mathcal{N}^\perp$. From Proposition 1, under the isomorphism induced by the basis vectors of $H^1 \times H^2$, $\mathcal{M}$ and $\bar{\mathcal{M}}$ are isomorphic. Let, $\boldsymbol{\varphi}(x) = [\varphi^1_1(x), \ldots, \varphi^1_{\mathcal{I}^1}(x), \varphi^2_1(x), \ldots, \varphi^2_{\mathcal{I}^2}(x)] \in \mathbb{R}^{|\mathcal{I}^1|+|\mathcal{I}^2|}$, $x \in \mathcal{X}$. Let $\hat{\mathcal{M}}$ be the span of $\{\boldsymbol{\varphi}(x)\}_{x\in\mathcal{X}}$. From the definition of $\mathcal{N}$, it follows that $\mathcal{N} = \hat{\mathcal{M}}^\perp$. Since $\mathcal{N}$ is a closed subspace, $\mathcal{N} = \mathcal{N}^{\perp^\perp} = \bar{\mathcal{M}}^\perp$. Thus, $\hat{\mathcal{M}}^\perp = \bar{\mathcal{M}}^\perp$ which implies that $\hat{\mathcal{M}}^{\perp^\perp} = \bar{\mathcal{M}}^{\perp^\perp}$. Since $\hat{\mathcal{M}}$ and $\bar{\mathcal{M}}$ are finite dimensional subspaces, they are closed, which implies that $\hat{\mathcal{M}} = \hat{\mathcal{M}}^{\perp^\perp} = \bar{\mathcal{M}}^{\perp^\perp} = \bar{\mathcal{M}}$. Hence, $\hat{\mathcal{M}} = \bar{\mathcal{M}}$. Since H is isomorphic to $\mathcal{M}$, it is isomorphic to $\bar{\mathcal{M}}$ and hence to $\hat{\mathcal{M}}$. We choose a basis for $\hat{\mathcal{M}}$ as follows. First, we choose $\bar{x}^1_1$ arbitrarily to obtain the first basis vector $\boldsymbol{\varphi}(\bar{x}^1_1)$. Let $\hat{M} = \hat{M}^1_1 \oplus \text{Span}\Big(\boldsymbol{\varphi}(\bar{x}^1_1)\Big)$. $\boldsymbol{\varphi}(\bar{x}^1_2)$ is chosen from $\hat{M}^1_1$. This process is repeatedly iteratively where $\boldsymbol{\varphi}(\bar{x}^1_j)$ is chosen from $\hat{\mathcal{M}}^1_{j-1}$ with $\hat{\mathcal{M}} = \hat{\mathcal{M}}^1_{j-1} \oplus \text{Span}\Big(\{\boldsymbol{\varphi}(\bar{x}^1_k)\}_{k=1}^{k=j-1}\Big)$ for $j = 2, \ldots, m$. Thus, $\{\boldsymbol{\varphi}(\bar{x}^1_j)\}_{j=1}^m$ spans $\hat{\mathcal{M}}$. Using the isomorphism induced by the basis vectors of $H^1 \times H^2$ from $\hat{\mathcal{M}}$ to $\mathcal{M}$, we note that each $\boldsymbol{\varphi}(\bar{x}^1_k)$ gets mapped to $\Big(\sum_{j\in\mathcal{I}^1} \varphi^1_j(\bar{x}^1_k)(\varphi^1_j(\cdot), \theta^2) + \sum_{j\in\mathcal{I}^2} \varphi^2_j(\bar{x}^1_k)(\theta^1, \varphi^2_j(\cdot))\Big) = \Big(\sum_{j\in\mathcal{I}^1} \varphi^1_j(\bar{x}^1_k)\varphi^1_j(\cdot), \sum_{j\in\mathcal{I}^2} \varphi^2_j(\bar{x}^1_k)\varphi^2_j(\cdot)\Big)$. Invoking the isomorphism $L_\mathcal{M}$ from $\mathcal{M}$ to H, each $\boldsymbol{\varphi}(\bar{x}^1_k)$ gets mapped to $\sum_{j\in\mathcal{I}^1} \varphi^1_j(\bar{x}^1_k)\varphi^1_j(\cdot) + \sum_{j\in\mathcal{I}^2} \varphi^2_j(\bar{x}^1_k)\varphi^2_j(\cdot) = K(\cdot, \bar{x}^1_k)$. Thus, $\{K(\cdot, \bar{x}^1_j)\}_{j=1}^m$ spans H. Let $\tilde{\mathcal{M}}$ be the span of $\{\boldsymbol{\varphi}(x)\}_{x\in\mathcal{X}\sim\{\bar{x}^1_j\}_{j=1}^m}$ and $\tilde{\mathcal{N}}$ be defined as,

$$\tilde{\mathcal{N}} = \Big\{\Big(\boldsymbol{\alpha^1}, \boldsymbol{\alpha^2}\Big) \in \mathbb{R}^{|\mathcal{I}^1|+|\mathcal{I}^2|} : \sum_{j\in\mathcal{I}^1} \alpha^1_j\varphi^1_j(x) + \sum_{j\in\mathcal{I}^2} \alpha^2_j\varphi^2_j(x) = 0, \forall x \in \mathcal{X} \sim \{\bar{x}^1_j\}_{j=1}^m\Big\},$$

where, $\boldsymbol{\alpha^i} = \Big(\alpha^i_1, \ldots, \alpha^i_{|\mathcal{I}^i|}\Big), i = 1, 2$. Clearly, $\mathcal{N} \subset \tilde{\mathcal{N}}$. Since $\mathcal{X}$ is closed, connected subset of $\mathbb{R}^d$ without isolated points. Let $\{x_n\} \subset \mathcal{X} \sim \{\bar{x}^1_j\}_{j=1}^m$ be a sequence such that it converges to one of the $\bar{x}^1_j$. Suppose $\Big(\boldsymbol{\alpha^1}, \boldsymbol{\alpha^2}\Big) \in \tilde{\mathcal{N}}$. Then,

$$\sum_{j\in\mathcal{I}^1} \alpha^1_j\varphi^1_j(x_n) + \sum_{j\in\mathcal{I}^2} \alpha^2_j\varphi^2_j(x_n) = 0, \forall n \implies \lim_{n\to\infty} \sum_{j\in\mathcal{I}^1} \alpha^1_j\varphi^1_j(x_n) + \sum_{j\in\mathcal{I}^2} \alpha^2_j\varphi^2_j(x_n) = 0$$

By continuity of the feature maps, the above implies, $\sum_{j\in\mathcal{I}^1}\alpha_j^1\varphi_j^1(\bar{x}_j^1)+\sum_{j\in\mathcal{I}^2}\alpha_j^2\varphi_j^2(\bar{x}_j^1)=0$. Since the same argument can be presented for all $\{\bar{x}_j^1\}_{j=1}^m$, this implies that $\Big(\boldsymbol{\alpha^1},\boldsymbol{\alpha^2}\Big)\in\mathcal{N}$. Thus, $\mathcal{N}=\tilde{\mathcal{N}}$ and $\hat{\mathcal{M}}=\tilde{\mathcal{M}}$. Since there is no loss in dimensionality, $\{\boldsymbol{\varphi}(\bar{x}_j^2)\}_{j=1}^m$ can be chosen using exactly the same procedure described to choose $\{\boldsymbol{\varphi}(\bar{x}_j^1)\}_{j=1}^m$, however using $\tilde{\mathcal{M}}$ in the place of $\hat{\mathcal{M}}$ leading to the construction of $\{K(\cdot,\bar{x}_j^2)\}_{j=1}^m$ which spans H. ■

Proposition 3 *The uploaded and downloaded function at Agent i can be expressed uniquely as a linear combination of $\{K^i(\cdot,\bar{x}_j^i)\}_{j=1}^m$.*

Proof Since $\bar{L}^i$ is symmetric (Lemma 8), from the spectral theorem, it follows that (i) the eigenvectors of $\bar{L}^i$, $\{\bar{\varphi}_j^i\}_{j=1}^m$ are an orthonormal basis for H; (ii) the eigenvalues of $\bar{L}^i$, $\{\lambda_j^i\}_{j=1}^m$, are real. The square root of operator $\bar{L}^i$, is defined as $\sqrt{\bar{L}^i}(\bar{\varphi}_j^i)=\sqrt{\lambda_j^i}\bar{\varphi}_j^i$. Any $f\in H$, specifically the fused function at stage n, f_n is first expressed using the eigen vectors of $\bar{L}^i$ as $f_n=\sum_{k=1}^m b_{n,k}^i\bar{\varphi}_k^i$. Invoking Proposition 2, $\bar{\varphi}_k^i=\sum_{j=1}^m a_{k,j}^iK(\cdot,\bar{x}_j^i)$. This implies that, $\bar{L}^i\big(\bar{\varphi}_k^i\big)=\sum_{j=1}^m a_{k,j}^i\bar{L}^i\big(K(\cdot,\bar{x}_j^i)\big)=\sum_{j=1}^m a_{k,j}^iK^i(\cdot,\bar{x}_j^i)$. The last equality follows from definition in Lemma 8 and has been proved in Raghavan and Johansson (2024b). Since $\bar{L}^i(\bar{\varphi}_k^i)=\lambda_k^i\bar{\varphi}_k^i$, it follows that $\bar{\varphi}_k^i=\frac{1}{\lambda_k^i}\sum_{j=1}^m a_{k,j}^iK^i(\cdot,\bar{x}_j^i)$, $\lambda_k^i\neq 0$. From Theorem 9, we note that, $\{\bar{\varphi}_j^i\}_{\lambda_j^i\neq 0}$ spans H^i. Hence, any vector $f^i\in H^i$, $f^i=\sum\limits_{k:\lambda_j^i\neq 0}c_k^i\sum_{j=1}^m a_{k,j}^iK^i(\cdot,\bar{x}_j^i)$, which implies that $\{K^i(\cdot,\bar{x}_j^i)\}_{j=1}^m$ spans H^i. Thus, the uploaded function can be expressed uniquely as a linear combination of $\{K^i(\cdot,\bar{x}_j^i)\}_{j=1}^m$. For $\lambda_k^i\neq 0$, $\sqrt{\bar{L}^i}(\bar{\varphi}_k^i)=\frac{1}{\sqrt{\lambda_k^i}}\bar{L}^i(\bar{\varphi}_k^i)=\frac{1}{\sqrt{\lambda_k^i}}\Big(\sum_{j=1}^m a_{k,j}^iK^i(\cdot,\bar{x}_j^i)\Big)$. From Theorem 9, the function downloaded on to the knowledge space of the agent i at stage n, $\bar{f}_n^i$, is

$$\sqrt{\bar{L}^i}\circ\Pi_{\mathcal{N}\left(\sqrt{\bar{L}^i}\right)^\perp}(f_n)=\sum_{k:\lambda_k^i\neq 0}b_{n,k}^i\sqrt{\bar{L}^i}\big(\bar{\varphi}_k^i\big)=\sum_{k:\lambda_k^i\neq 0}\frac{b_{n,k}^i}{\sqrt{\lambda_k^i}}\Big(\sum_{j=1}^m a_{k,j}^iK^i(\cdot,\bar{x}_j^i)\Big),$$

which is equal to $\sum_{j=1}^m\Big(\sum\limits_{k:\lambda_k^i\neq 0}\frac{b_{n,k}^ia_{k,j}^i}{\sqrt{\lambda_k^i}}\Big)K^i(\cdot,\bar{x}_j^i)$. Thus, downloading the fused function is equivalent to the fusion center transmitting the vector $\Big\{\sum\limits_{k:\lambda_k^i\neq 0}\frac{b_{n,k}^ia_{k,j}^i}{\sqrt{\lambda_k^i}}\Big\}_{j=1}^m$ to agent i. ■

To obtain the closed form expression for the downloaded function, we note that it is crucial to express the fused function in terms of the eigenvectors of $\bar{L}^i$. If not, if f_n is expressed directly using the basis vectors $\{K(\cdot,\bar{x}_j^i)\}_{j=1}^m$, then computation of $\sqrt{\bar{L}^i}\big(K(\cdot,\bar{x}_j^i)\big)$ is not straight forward and $\sqrt{\bar{L}^i}(f)=\frac{1}{\sqrt{\lambda}}\bar{L}^i(f)$ if only if f is an eigenvector of $\bar{L}^i$.

3.2. Estimation at The Agents

In this section, we present the solution to the estimation problem at the agents. We define and study the asymptotic properties of the learning operator at the agents. For each agent i, let $\mathbf{K^i}=(K^i(\bar{x}_j^i,\bar{x}_k^i))_{jk}=(\langle K^i(\cdot,\bar{x}_k^i),K^i(\cdot,\bar{x}_j^i)\rangle_{H^i})$. Since the inner product is symmetric, $\mathbf{K^i}\in\mathbb{R}^{m\times m}$ is symmetric matrix. Let $\bar{\mathbf{K}}^{\mathbf{i}}:\mathcal{X}\to\mathbb{R}^m$ be defined as $\bar{\mathbf{K}}^{\mathbf{i}}(\cdot)=[K^i(\cdot,\bar{x}_1^i),\dots,K^i(\cdot,\bar{x}_m^i)]$. For any data point, $\bar{\mathbf{K}}^{\mathbf{i}}(x_n^i)=[K^i(x_n^i,\bar{x}_1^i);\dots;K^i(x_n^i,\bar{x}_m^i)]$ is a column vector in $\mathbb{R}^m$. Given vector $\boldsymbol{\alpha^i}=[\alpha_1^i;\dots;\alpha_m^i]\in\mathbb{R}^m$, we use notation $f^i=\boldsymbol{\alpha^T}\bar{\mathbf{K}}^{\mathbf{i}}(\cdot)$ for the function $f^i=\sum_{j=1}^m\alpha_j^iK(\cdot,\bar{x}_j^i)\in H^i$.

Proposition 4 *Let* $\boldsymbol{\alpha_n^{1,*}} = \Big(\varrho_n^i \mathbf{K^i} + \bar{\mathbf{K}}^{\mathbf{i}}(x_n^i)\bar{\mathbf{K}}^{\mathbf{i}^{\mathbf{T}}}(x_n^i)\Big)^{-1}\Big(\bar{\mathbf{K}}^{\mathbf{i}}(x_n^i)y_n^i + \varrho_n^i \mathbf{K^i}\bar{\boldsymbol{\alpha}}^{\mathbf{1}}_{\boldsymbol{n-1}}\Big)$*. Then,* $f_n^{i,*} = \boldsymbol{\alpha_n^{1,*}}^T \bar{\mathbf{K}}^{\mathbf{i}}(\cdot)$ *solves the optimization problem in problem* $(P1)_n^i$*.*

Proof Using the notation, $f_n^1 = \boldsymbol{\alpha_n^1}^T \bar{\mathbf{K}}^{\mathbf{i}}(\cdot)$, the cost function of the estimation problem in subsection 2.1 can be expressed as $C^i(f_n^i) = (y_n^1 - \boldsymbol{\alpha_n^1}^T \bar{\mathbf{K}}^{\mathbf{i}}(x_n^i))^2 + \varrho_n^i(\boldsymbol{\alpha_n^1} - \bar{\boldsymbol{\alpha}}^{\mathbf{1}}_{\boldsymbol{n-1}})^T \mathbf{K^i}(\boldsymbol{\alpha_n^1} - \bar{\boldsymbol{\alpha}}^{\mathbf{1}}_{\boldsymbol{n-1}})$. The gradient of $C^i(f_n^i)$ with respect to $\boldsymbol{\alpha_n^1}$ is,

$$\nabla_{\boldsymbol{\alpha_n^1}} C^i(f_n^i) = -2(y_n^1 - \boldsymbol{\alpha_n^1}^T \bar{\mathbf{K}}^{\mathbf{i}}(x_n^i))\bar{\mathbf{K}}^{\mathbf{i}}(x_n^i) + \varrho_n^i \mathbf{K^i}^T(\boldsymbol{\alpha_n^1} - \bar{\boldsymbol{\alpha}}^{\mathbf{1}}_{\boldsymbol{n-1}}) + \varrho_n^i \mathbf{K^i}(\boldsymbol{\alpha_n^1} - \bar{\boldsymbol{\alpha}}^{\mathbf{1}}_{\boldsymbol{n-1}}).$$

It can be verified that, $\Big(\boldsymbol{\alpha_n^1}^T \bar{\mathbf{K}}^{\mathbf{i}}(x_n^i)\Big)\bar{\mathbf{K}}^{\mathbf{i}}(x_n^i) = \bar{\mathbf{K}}^{\mathbf{i}}(x_n^i)\bar{\mathbf{K}}^{\mathbf{i}^{\mathbf{T}}}(x_n^i)\boldsymbol{\alpha_n^1}$. Thus,

$$\nabla_{\boldsymbol{\alpha_n^1}} C^i(f_n^i) = 2(\varrho_n^i \mathbf{K^i} + \bar{\mathbf{K}}^{\mathbf{i}}(x_n^i)\bar{\mathbf{K}}^{\mathbf{i}^{\mathbf{T}}}(x_n^i))\boldsymbol{\alpha_n^1} - 2(\bar{\mathbf{K}}^{\mathbf{i}}(x_n^i)y_n^i + \varrho_n^i \mathbf{K^i}\bar{\boldsymbol{\alpha}}^{\mathbf{1}}_{\boldsymbol{n-1}}).$$

By setting the gradient to zero, we obtain the desired result. ■

3.3. The Fusion Problem

Let $\mathbf{K} = (K(\bar{x}_j^p, \bar{x}_k^q))_{jk} = (\langle K(\cdot, \bar{x}_k^q), K(\cdot, \bar{x}_j^p)\rangle_H)_{j,k}, p, q = 1, 2, j, k, = 1 \ldots m$ be a symmetric matrix in $\mathbb{R}^{2m \times 2m}$ and $\hat{\mathbf{K}}^{\mathbf{i}}_{\mathbf{n}} = (K^i(\hat{x}_{n,j}^i, \hat{x}_{n,k}^i))_{jk} = (\langle K^i(\cdot, \hat{x}_{n,k}^i), K^i(\cdot, \hat{x}_{n,j}^i)\rangle_{H^i})_{j,k} \in \mathbb{R}^{m \times m}$ be a symmetric matrix. Let $\bar{\mathbf{K}} : \mathcal{X} \to \mathbb{R}^{2m}$ be defined as $\bar{\mathbf{K}}(\cdot) = \Big[K(\cdot, \bar{x}_1^1); \ldots, K(\cdot, \bar{x}_m^1); K(\cdot, \bar{x}_1^2); \ldots; K(\cdot, \bar{x}_m^2)\Big]$. Let $\tilde{\mathbf{K}}^{\mathbf{i}}_{\mathbf{n}} : \mathcal{X} \to \mathbb{R}^m$ be defined as $\tilde{\mathbf{K}}^{\mathbf{i}}_{\mathbf{n}}(\cdot) = [K(\cdot, \hat{x}_{n,1}^i), \ldots, K(\cdot, \hat{x}_{n,m}^i)]$. Let $\hat{\mathbf{Y}}_n^i = \Big[\hat{y}_{n,1}^i; \ldots; \hat{y}_{n,m}^i\Big] \in \mathbb{R}^m$ and $\hat{\mathbf{Y}}_n = \Big[\hat{\mathbf{Y}}_n^1; \hat{\mathbf{Y}}_n^2\Big] \in \mathbb{R}^{2m}$. Let $\check{\mathbf{K}}^{\mathbf{i}} : \mathcal{X} \to \mathbb{R}^m$ be defined as $\check{\mathbf{K}}^{\mathbf{i}}(\cdot) = \Big[K(\cdot, \bar{x}_1^i); \ldots, K(\cdot, \bar{x}_m^i)\Big]$ and $\check{\mathbf{K}}^{\mathbf{i}} = (K(\bar{x}_j^i, \bar{x}_k^i))_{jk} = (\langle K(\cdot, \bar{x}_k^i), K(\cdot, \bar{x}_j^i)\rangle_H)_{j,k}, i = 1, 2,\ j, k = 1 \ldots m$ be a symmetric matrix in $\mathbb{R}^{m \times m}$. A function f^i which belongs to H^i and H can be expressed as $\boldsymbol{\alpha^i}^T \bar{\mathbf{K}}^{\mathbf{i}}(\cdot)$ and $\hat{\boldsymbol{\alpha}}^{\boldsymbol{i}^T} \check{\mathbf{K}}^{\mathbf{i}}(\cdot)$. Indeed, since $\{K(\cdot, \bar{x}_j^i)\}_{j=1}^m$ is basis for H, $\exists! \{M_{kj}^i\}_{k=1}^m$ such that $K^i(\cdot, \bar{x}_j^i) = \sum_{k=1}^m M_{kj}^i K(\cdot, \bar{x}_k^i), \forall j$. Thus,

$$\sum_{j=1}^m \alpha_j^i K^i(\cdot, \bar{x}_j^i) = \sum_{k=1}^m \sum_{j=1}^m M_{kj}^i \alpha_j^i K(\cdot, \bar{x}_k^i) = \sum_{k=1}^m \hat{\alpha}_k K(\cdot, \bar{x}_k^i),$$

which implies that $\hat{\boldsymbol{\alpha}}^{\boldsymbol{i}} = M^i \boldsymbol{\alpha^i}$ where $M^i = (M_{kj})_{k,j} \in \mathbb{R}^{m \times m}$. In the following, we solve the problems $(P2)_n^i$ and $(P3)_n$.

Proposition 5 $\hat{x}_{n,j}^i = \bar{x}_j^i\ \forall n \in \mathbb{N}, i = 1, 2, j = 1, \ldots, m$ *and* $\hat{\mathbf{Y}}_n^i = \Big(\mathbf{K^i} + \varrho_n^i \mathbb{I}_m\Big)\boldsymbol{\alpha_n^i}$ *solve the problem* $(P2)_n^i$*, where* $f_n^i = \boldsymbol{\alpha_n^i}^T \bar{\mathbf{K}}^{\mathbf{i}}(\cdot)$*.* $f_n^* = \boldsymbol{\alpha_n^T} \bar{\mathbf{K}}(\cdot)$*, where* $\boldsymbol{\alpha_n} = \Big(\mathbf{K^T K} + \varrho_n \mathbf{K}\Big)^{-1}\Big(\mathbf{K^T}\hat{\mathbf{Y}}_n\Big)$ *solves* $(P3)_n$*.*

Proof If Agent i had received the data points $\{(\hat{x}_{n,j}^i, \hat{y}_{n,j}^i)\}_{j=1}^m$, by the *Representer Theorem*, Hofmann et al. (2008), the optimal solution for a least squares regression problem for Agent i is given by $g_n^i = \bar{\boldsymbol{\alpha}}_{\boldsymbol{n}}^{\boldsymbol{i}^T} \tilde{\mathbf{K}}^{\mathbf{i}}_{\mathbf{n}}(\cdot)$, where $\bar{\boldsymbol{\alpha}}_{\boldsymbol{n}}^{\boldsymbol{i}} = \Big(\hat{\mathbf{K}}_{\mathbf{n}}^{\mathbf{i}^{\mathbf{T}}} \hat{\mathbf{K}}^{\mathbf{i}}_{\mathbf{n}} + \varrho_n \hat{\mathbf{K}}^{\mathbf{i}}_{\mathbf{n}}\Big)^{-1}\Big(\hat{\mathbf{K}}_{\mathbf{n}}^{\mathbf{i}^{\mathbf{T}}} \hat{\mathbf{Y}}^{\mathbf{i}}_{\mathbf{n}}\Big)$. From Proposition 3, the uploaded function is of the form $f_n^i = \boldsymbol{\alpha_n^i} \bar{\mathbf{K}}^{\mathbf{i}}(\cdot)$. The feasibility problem in $(P2)_n^i$ is solved if and only if $g_n^i = f_n^i$. To achieve the same we let, $\tilde{\mathbf{K}}^{\mathbf{i}}_{\mathbf{n}}(\cdot) = \bar{\mathbf{K}}^{\mathbf{i}}(\cdot), \forall n$ and $\bar{\boldsymbol{\alpha}}_{\boldsymbol{n}}^{\boldsymbol{i}} = \boldsymbol{\alpha_n^i}, \forall n$.

If $\hat{x}_{n,j}^i = \bar{x}_j^i$, then $\tilde{\mathbf{K}}_{\mathbf{n}}^{\mathbf{i}}(\cdot) = \bar{\mathbf{K}}^{\mathbf{i}}(\cdot)$. This implies that $\hat{\mathbf{K}}_{\mathbf{n}}^{\mathbf{i}} = \mathbf{K}^{\mathbf{i}}$. Using $\bar{\boldsymbol{\alpha}}_{\boldsymbol{n}}^{\boldsymbol{i}} = \boldsymbol{\alpha}_{\boldsymbol{n}}^{\boldsymbol{i}}$, $\hat{\mathbf{Y}}_n^i$ is obtained as $\hat{\mathbf{Y}}_n^i = \mathbf{K}^{\mathbf{i}^{\mathbf{T}^{-1}}}\Big(\mathbf{K}^{\mathbf{i}^{\mathbf{T}}}\mathbf{K}^{\mathbf{i}} + \varrho_n \mathbf{K}^{\mathbf{i}}\Big)\boldsymbol{\alpha}_{\boldsymbol{n}}^{\boldsymbol{i}} = \Big(\mathbf{K}^{\mathbf{i}} + \varrho_n \mathbb{I}_m\Big)\boldsymbol{\alpha}_{\boldsymbol{n}}^{\boldsymbol{i}}$. Given the gram matrix generated from the kernel K at the fusion input data points, $\{\bar{x}_j^1\}_{j=1}^m \cup \{\bar{x}_j^2\}_{j=1}^m$, $\mathbf{K}$, and the fusion output data points $\hat{\mathbf{Y}}_n$, the solution of the estimation problem, $(P3)_n$, is given by the *Representer Theorem*, $f_n^* = \boldsymbol{\alpha}_{\boldsymbol{n}}^{\boldsymbol{T}}\bar{\mathbf{K}}$, where $\boldsymbol{\alpha}_{\boldsymbol{n}} = \Big(\mathbf{K}^{\mathbf{T}}\mathbf{K} + \varrho_n \mathbf{K}\Big)^{-1}\Big(\mathbf{K}^{\mathbf{T}}\hat{\mathbf{Y}}_n\Big)$. ■

4. Conclusion and Future Work

To conclude, we presented a distributed algorithm for estimation of functions given data. Key aspects of the algorithm included use of heterogeneous data, use of different features by different agents, fusion of models by estimating data which could have generated the models, and, the use of uploading and downloading operators due to different agents learning in different spaces. Going forward, we are interested in studying nonparametric estimation problems using the algorithm developed in this paper. We would like to investigate transform methods for the fusion problem, i.e, the meta-learning problem in RKHS. Quantification and formal guarantees of transfer of knowledge from one agent to another is also of interest.

5. Appendix: Knowledge Space, Fusion Space, Uploading and Downloading Operator

The construction of the knowledge spaces for the individual agents and the fusion space has been discussed in detail in Raghavan and Johansson (2024b). We mention the key results here which have been referenced in the previous sections.

Theorem 6 *If $K^i(\cdot,\cdot)$ is the reproducing kernel of Hilbert space H^i, with norm $||\cdot||_{H^i}$, then $K(x,y) = K^1(x,y) + K^2(x,y)$ is the reproducing kernel of the space $H = \{f | f = f^1 + f^2 | f^i \in H^i\}$ with the norm:*

$$||f||_H^2 = \min_{\substack{f^1+f^2=f,\\ f^i\in H^i}} ||f^1||_{H^1}^2 + ||f^2||_{H^2}^2.$$

Corollary 7 *The uploading operator from agent i's knowledge space, H^i, to the fusion space H, $\hat{L}^i : H^i \to H$, is $\hat{L}(f) = f$. $\hat{L}^i(\cdot)$, is linear and is bounded, $||\hat{L}^i|| = \sup\{||f||_H : f \in H^i, ||f||_{H^i} = 1\} \leq 1$.*

Lemma 8 *Given the RKHS, $(H, \langle\cdot,\cdot\rangle_H)$, with kernel $K(\cdot,\cdot)$ and the kernels $K^i(\cdot,\cdot)$, $i = 1,2$, such that $K(x,y) = K^1(x,y) + K^2(x,y)$, we define operators, $\bar{L}^i : H \to H$, as*

$$\bar{L}^i(f)(x) = \langle f(\cdot), K^i(\cdot,x)\rangle_H, \text{ for, } i = 1,2.$$

Then, $\bar{L}^i$ is linear, symmetric, positive and bounded, $||\bar{L}^i|| \leq 1$.

Theorem 9 *The linear space $\bar{H}^i = \{g : g = \sqrt{\bar{L}^i}(f), f \in H\}$ is a RKHS with kernel K^i. $\sqrt{\bar{L}^i}(\cdot)$ establishes an isometric isomorphism between $\mathcal{N}\big(\sqrt{\bar{L}^i}\big)^\perp$ and $\bar{H}^i$, and the norm, $||f||_{\bar{H}^i} = ||g||_H$, where $f = \sqrt{\bar{L}^i}g, g \in \mathcal{N}\big(\sqrt{\bar{L}^i}\big)^\perp$. The downloading operator from the fusion space H to agent i's knowledge space, H^i, is $\sqrt{\bar{L}^i} \circ \Pi_{\mathcal{N}\left(\sqrt{\bar{L}^i}\right)^\perp}$. The downloading operator is linear and bounded.*

For the proofs of Theorem 6, Corollary 7, Lemma 8 and Theorem 9 we refer to Raghavan and Johansson (2024b).

Acknowledgments

This work was supported in part by Swedish Research Council Distinguished Professor Grant 2017-01078, Knut and Alice Wallenberg Foundation Wallenberg Scholar Grant, and the Swedish Strategic Research Foundation FUSS SUCCESS Grant.

References

Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2): 423–443, 2018.

Phillip K Chan, Salvatore J Stolfo, et al. Toward parallel and distributed learning by meta-learning. In *AAAI workshop in Knowledge Discovery in Databases*, number 1993, page 227. Citeseer, 1993.

Ryad Chellali et al. A distributed multi robot slam system for environment learning. In *2013 IEEE Workshop on Robotic Intelligence in Informationally Structured Space (RiiSS)*, pages 82–88. IEEE, 2013.

Patricia Derler, Edward A Lee, and Alberto L Sangiovanni-Vincentelli. Addressing modeling challenges in cyber-physical systems. *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2011-17*, 2011.

Lixin Duan, Ivor W Tsang, and Dong Xu. Domain transfer multiple kernel learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):465–479, 2012.

Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola. Kernel methods in machine learning. *The Annals of Statistics*, 36(3):1171–1220, 2008.

Pierre-Yves Lajoie, Benjamin Ramtoula, Yun Chang, Luca Carlone, and Giovanni Beltrame. Door-slam: Distributed, online, and outlier resilient slam for robotic teams. *IEEE Robotics and Automation Letters*, 5(2):1656–1663, 2020.

Gert RG Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine learning research*, 5(Jan):27–72, 2004.

Fayao Liu, Luping Zhou, Chunhua Shen, and Jianping Yin. Multiple kernel learning in the primal for multimodal alzheimer's disease classification. *IEEE journal of biomedical and health informatics*, 18(3):984–990, 2013.

Ji Liu, Jizhou Huang, Yang Zhou, Xuhong Li, Shilei Ji, Haoyi Xiong, and Dejing Dou. From distributed machine learning to federated learning: A survey. *Knowledge and Information Systems*, 64(4):885–917, 2022.

Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696, 2011.

Dinh C Nguyen, Ming Ding, Pubudu N Pathirana, Aruna Seneviratne, Jun Li, and H Vincent Poor. Federated learning for internet of things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 23(3):1622–1658, 2021.

Diego Peteiro-Barral and Bertha Guijarro-Berdiñas. A survey of methods for distributed machine learning. *Progress in Artificial Intelligence*, 2:1–11, 2013.

Soujanya Poria, Haiyun Peng, Amir Hussain, Newton Howard, and Erik Cambria. Ensemble application of convolutional neural networks and multiple kernel learning for multimodal sentiment analysis. *Neurocomputing*, 261:217–230, 2017.

Aneesh Raghavan and Karl Henrik Johansson. Distributed regression by two agents from noisy data. In *2023 European Control Conference (ECC)*, pages 1–6. IEEE, 2023.

Aneesh Raghavan and Karl Henrik Johansson. Distributed learning and function fusion in reproducing kernel hilbert space. *arXiv preprint arXiv:2401.03012*, 2024a.

Aneesh Raghavan and Karl Henrik Johansson. Distributed estimation by two agents with different feature spaces. *arXiv preprint arXiv:2403.07749*, 2024b.

Stanley J Rosenschein. Formal theories of knowledge in artificial intelligence and robotics. *New generation computing*, 3:345–357, 1985.

Karan Sikka, Karmen Dykstra, Suchitra Sathyanarayana, Gwen Littlewort, and Marian Bartlett. Multiple kernel learning for emotion recognition in the wild. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pages 517–524, 2013.

Yulun Tian, Yun Chang, Fernando Herrera Arias, Carlos Nieto-Granda, Jonathan P How, and Luca Carlone. Kimera-multi: Robust, distributed, dense metric-semantic slam for multi-robot systems. *IEEE Transactions on Robotics*, 38(4), 2022.

Hans Van Ditmarsch, Wiebe van Der Hoek, and Barteld Kooi. *Dynamic epistemic logic*, volume 337. Springer Science & Business Media, 2007.

Joost Verbraeken, Matthijs Wolting, Jonathan Katzy, Jeroen Kloppenburg, Tim Verbelen, and Jan S Rellermeyer. A survey on distributed machine learning. *Acm computing surveys (csur)*, 53(2): 1–33, 2020.