A Collaborative Training Algorithm for Distributed Learning

Joel B. Predd, *Member, IEEE*, Sanjeev R. Kulkarni, *Fellow, IEEE*, and H. Vincent Poor, *Fellow, IEEE*

***Abstract*—In this paper, an algorithm is developed for collaboratively training networks of kernel-linear least-squares regression estimators. The algorithm is shown to distributively solve a relaxation of the classical centralized least-squares regression problem. A statistical analysis shows that the generalization error afforded agents by the collaborative training algorithm can be bounded in terms of the relationship between the network topology and the representational capacity of the relevant reproducing kernel Hilbert space. Numerical experiments suggest that the algorithm is effective at reducing noise. The algorithm is relevant to the problem of distributed learning in wireless sensor networks by virtue of its exploitation of local communication. Several new questions for statistical learning theory are proposed.**

***Index Terms*—Collaboration, distributed learning, empirical risk minimization, kernel methods, learning, nonparametric, sensor networks.**

I. Introduction

In this paper, we address the problem of *distributed learning under communication constraints*, motivated primarily by distributed signal processing in wireless sensor networks (WSNs). WSNs are *a fortiori* designed to make inferences from the environments they are sensing; however, they are typically characterized by constraints on energy and bandwidth, which limit the sensors' ability to communicate data with each other or with a centralized fusion center for centralized signal processing. Nonparametric methods studied within machine learning have demonstrated widespread empirical success in many centralized (i.e., communication *unconstrained*) signal processing applications. Thus, a natural question arises: can the power of machine learning methods be tapped for nonparametric inference in distributed learning under communication constraints?

Manuscript received March 30, 2007; revised August 30, 2008. Current version published March 18, 2009. This work was supported in part by the Army Research Office under Grant DAAD19-00-1-0466, in part by the U. S. Army Pantheon Project, in part by Draper Laboratory under Grant IR&D 6002, in part by the National Science Foundation under Grants CCR-0020524 and CCR-0312413, and in part by the Office of Naval Research under Contracts W911NF-07-1-0185 and N00014-07-1-0555. The material in this paper was presented in part at the IEEE Information Theory Workshop, Punta del Este, Uruguay, March 2006.

J. B. Predd is with the RAND Corporation, Pittsburgh, PA 15213 USA (e-mail: jpredd@rand.org).

S. R. Kulkarni and H. V. Poor are with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: kulkarni@princeton.edu; poor@princeton.edu).

Communicated by A. Krzyżak, Associate Editor for Pattern Recognition, Statistical Learning and Inference.

Color versions of Figures 1–10 in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2009.2012992

Many classical learning rules are infeasible in wireless sensor networks, because constraints on energy and bandwidth constraints preclude one from accessing the entire training set. One approach to extending classical learning rules to distributed learning, and in particular to wireless sensor networks, focuses on developing communication-efficient *training algorithms* (several specific approaches are discussed below). While recognizing the strong theoretical foundation on which existing learning rules are designed, this approach interprets communication constraints as imposing computational limits on training, and assumes there is a methodology for assessing the efficiency of training algorithms from an energy and bandwidth perspective.

The importance of local communication transcends many analyses and implementations of wireless networks. Loosely speaking, local communication is that which occurs between neighboring sensors in a communication network. In wireless networks, the topology of the network is in correspondence with the topology of the environment, which is to say that a sensor's network neighborhood is roughly similar to its physical neighborhood. By an inverse square law, the energy required for two sensors to (wirelessly) communicate decreases with the distance between them; by the same law, multiple-access interference decreases with the distance between pairs of communicating nodes. Thus, by minimizing energy expenditure and by enabling spectral reuse, local communications are often an efficient mode of information transport in wireless networks.

The foregoing observation has motivated the development and analysis of many so-called *local message-passing algorithms* for distributed inference in wireless sensor networks. Roughly speaking, message-passing algorithms are those that use only local communication to achieve the same end (or approximately the same end) as "global" (i.e., centralized) algorithms that require sending "raw" data to a central processing facility. Message-passing algorithms are thought to be efficient by virtue of their exploitation of local communication. In practice, such intuitions must be formally justified. In theory, application-layer abstractions of local communication constitute a reasonable framework for studying distributed inference in general, and for developing communication-efficient training algorithms for distributed learning in particular.

In this paper, we develop a local message-passing algorithm for collaboratively training networks of kernel-linear least-squares regression estimators. The algorithm is constructed to solve a relaxation of the classical centralized kernel-linear least-squares regression problem. A statistical analysis shows that the generalization error afforded agents by the collaborative training algorithm can be bounded in terms of the relationship between the network topology and the repre-

sentational capacity of the relevant reproducing kernel Hilbert space; this is in contrast to related approaches which relate the similarity structure encoded in the kernel and the network topology. Numerical experiments suggest that the algorithm is effective at reducing noise. As above, the algorithm is relevant to the problem of distributed learning in wireless sensor networks by virtue of its exploitation of local communication.

A. Organization

The remainder of this paper is organized as follows. In Section II, we review the supervised learning model for nonparametric least-squares regression, reproducing kernel methods, and alternating projection algorithms (the tool from mathematical programming on which our analysis relies). In Section III, we introduce a general model for distributed learning, and discuss related work on developing communication-efficient training algorithms for distributed learning. In Section IV, we develop a novel local message-passing algorithm that admits an interpretation as a collaborative training algorithm. A statistical analysis of generalization error is presented in Section V, and numerical experiments are summarized in Section VI. We conclude in Section VII. The proofs of main results are left to the Appendix.

II. Preliminaries

In this section, we briefly review the supervised learning model for nonparametric least-squares regression, reproducing kernel methods, and alternating projection algorithms. Since a thorough introduction to these models and methods is beyond the scope of this paper, we refer the reader to standard references on the topics; see, for example, [11], [21], [22], [49] and references therein.

A. Nonparametric Least-Squares Regression

Let X and Y be $\mathcal{X}$ and $\mathcal{Y}$-valued random variables, respectively. $\mathcal{X}$ is known as the feature, input, or observation space; $\mathcal{Y}$ is known as the label, output, or target space. For now, we allow $\mathcal{X}$ to be arbitrary, but take $\mathcal{Y} = \Re$. In the least-squares estimation problem, we seek a decision rule mapping inputs to outputs that minimizes the expected squared error. In particular, we seek a function $g : \mathcal{X} \to \mathcal{Y}$ that minimizes

$$\boldsymbol{E}\{|g(X) - Y|^2\}.$$

It is well-known that $\eta(x) = \boldsymbol{E}\{Y \mid X = x\}$ is the loss minimizing rule. However, without prior knowledge of the joint distribution of (X, Y), this regression function cannot be computed. In the supervised learning model, one is instead provided a database $S = \{(x_i, y_i)\}_{i=1}^n$ of training examples with $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}\ \forall i \in \{1, \ldots, n\}$; the learning task is to use S to estimate $\eta(x)$.

B. Regularized Kernel Methods

Regularized kernel methods [3], [49] offer one approach to nonparametric regression. In particular, let $\mathcal{H}_K$ denote the *reproducing kernel Hilbert space* (RKHS) induced by a *positive semi-definite kernel* $K(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \to \Re$; let $\| \cdot \|_{\mathcal{H}_K}$ denote the norm associated with $\mathcal{H}_K$. In practice, the kernel K is a design parameter, chosen as a similarity measure between inputs to reflect prior application-specific domain knowledge. The regularized kernel least-squares estimate is defined as the solution $f_\lambda \in \mathcal{H}_K$ of the following optimization problem:

$$\min_{f \in \mathcal{H}_K} \left[\frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \lambda \|f\|_{\mathcal{H}_K}^2 \right]. \tag{1}$$

The statistical behavior of this estimator is well understood under various assumptions on the stochastic process that generates the examples $\{(x_i, y_i)\}_{i=1}^n$ [49], [52]. In this paper, we focus primarily on algorithmic aspects of computing a solution to (1) (or an approximation thereof) in distributed environments. To this end, consider the following "Representer Theorem" established by [4], [24].

Theorem 1 ([24]): Let $g_n \in \mathcal{H}_K$ be the minimizer of (1). Then, there exists $\boldsymbol{c}_n \in \Re^n$ such that

$$g_n(\cdot) = \sum_{i=1}^{n} c_{c_{n,i}} K(\cdot, x_i).$$

From a computational perspective, the result is significant because it states that while the objective function (1) is defined over a potentially infinite-dimensional Hilbert space, its minimizer must lie in a finite-dimensional subspace.[1]

C. Alternating Projections Algorithms

Let $\mathcal{H}$ be a Hilbert space with a norm denoted by $\| \cdot \|$. Let $C_1, \ldots, C_m$ be closed convex subsets of $\mathcal{H}$ whose intersection $C = \cap_{i=1}^m C_i$ is nonempty. Let $P_C(\hat{x})$ denote the orthogonal projection of $\hat{x} \in \mathcal{H}$ onto C, i.e.,

$$P_C(\hat{x}) \triangleq \arg\min_{x \in C} \|x - \hat{x}\|.$$

Define $P_{C_i}(\hat{x})$ analogously.

Successive orthogonal projection (SOP) algorithms [11] provide a natural way to compute $P_C(\cdot)$ given $\{P_{C_i}(\cdot)\}_{i=1}^m$. For example, the (unrelaxed) SOP algorithm is defined as follows:

$$x_0 := \hat{x} \quad x_n := P_{C_{(n \bmod m)+1}}(x_{n-1}). \tag{2}$$

In words, the algorithm first projects $\hat{x}$ onto C_1, and then projects $P_{C_1}(\hat{x})$ onto C_2; it continues successively and iteratively projecting the image of the previous projection onto the next subset in sequence. In the case where C_i is a linear subspace for all $i \in \{1, \ldots, m\}$, this algorithm was first studied by von Neumann [51]. Often examined in the context of the *convex feasibility problem*, SOP has been generalized in various ways [11], to address more general convex sets and nonorthogonal (e.g., Bregman) projections; accordingly, the algorithm often takes on other names (e.g., the von Neumann–Halperin algorithm, Bregman's algorithm). Much of the behavior of this algorithm can be understood through Theorem 2; the proof of this fundamental result can be found in [7].

[1]Note that the minimizer g_n of (1) depends on λ. To simplify exposition, we omit this dependency from the notation.

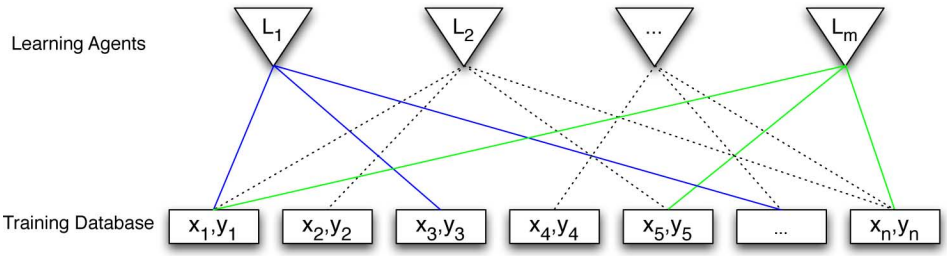

Fig. 1. A bipartite graph model for distributed learning.

Theorem 2: Let $\{C_i\}_{i=1}^m$ be a set of closed, convex subsets of $\mathcal{H}$ whose intersection $C = \cap_{i=1}^m C_i$ is nonempty. Let x_n be defined as in (2). Then, for every $x \in C$ and every $n \geq 1$

$$\|x_n - x\| \leq \|x_{n-1} - x\|.$$

Moreoever, $\lim_{n\to\infty} x_n \in \cap_{i=1}^m C_i$. If C_i are affine for all $i \in \{1, \ldots, m\}$, then $\lim_{n\to\infty} \|x_n - P_C(\hat{x})\| = 0$.

III. Distributed Learning

To motivate our model for distributed learning and to contrast related work, it is helpful to consider the following toy example.

Suppose that the feature space $\mathcal{X}$ models a set of observables measured by sensors in a wireless network. For example, the components of an element $x \in \mathcal{X} = \Re^3$ may model coordinates in a (planar) environment and time. $\mathcal{Y} = \Re$ may represent the space of temperature measurements. A decision maker may wish to know the temperature at some point in space–time; to reflect that these coordinates and the corresponding temperature are unknown, let us model them with the random variable (X, Y). A joint distribution $\boldsymbol{P}_{XY}$ may model the spatio-temporal correlation structure of a temperature field. If the field's structure is well understood, i.e., if $\boldsymbol{P}_{XY}$ can be assumed known *a priori*, then an estimate may be designed within the Bayesian inference framework [42]. However, if such prior information is unavailable, an alternative approach is necessary.

Suppose that sensors are randomly deployed about the environment, and collectively acquire a set $S_n \subset \mathcal{X} \times \mathcal{Y}$ of temperature measurements at various points in space–time.[2] The set S_n is akin to the training data described in Section II, and thus reproducing kernel methods (and indeed, many other supervised learning algorithms) seem naturally applicable to this field-estimation problem. However, the supervised learning model has abstracted away the process of data acquisition, and generally does not incorporate communication constraints that may limit a learning algorithm's access to data. Indeed, classical supervising learning algorithms depend critically on the assumption that the training data is entirely available to a single processor. However, in wireless sensor networks, the energy and bandwidth required to collect the sensors' raw measurements may be prohibitively large. Thus, training centralized learning rules may limit the sensors' battery life, may waste bandwidth, and may ultimately preclude one from realizing the potential of wireless sensor networks.

Sensors in WSNs are equipped with on-board processing capabilities, and thus have the ability to locally process information. Can this processing be exploited to develop communication-efficient learning algorithms that respect constraints on energy and bandwidth?

For another example, individuals in social networks (e.g., Facebook) have privileged access to data that they may be unwilling to share with people they do not trust. For example, an individual may have access to their annotated home photos and to the annotated photos shared by their friends. Each individual may want to train a classifier to support automated image annotation, but such a classifier may generalize poorly with the limited data available. Can individual data, together with localized processing, be exploited in privacy-sensitive learning algorithms in social networks?

A. A Model for Distributed Learning

As a starting point to studying the aforementioned questions, consider a more general model for distributed learning. Suppose that each member of a collection of m *learning agents* (e.g., sensors in a wireless network) has limited access to the training database $S_n = \{(x_i, y_i)\}_{i=1}^n$. In particular, assume that learning agent j has access only to the training examples in a subset $S_n^j \subseteq S_n$. We shall henceforth refer to $\{S_n^j\}_{j=1}^m$ as an *ensemble*.

A bipartite graph is a convenient way to represent an ensemble in this model for distributed learning. As depicted in Fig. 1, nodes on the top level of the graph represent learning agents; nodes on the bottom level represent training examples. An edge between a learning agent j and a training sample i posits the existence of a communication channel over which agent j can retrieve information about example i. For now, we make no additional assumptions on the structural relationship between the agents' locally accessible training sets; for example, we do not require the ensemble $\{S_n^j\}_{j=1}^m$ to partition S_n, nor do we require the corresponding bipartite graph to be connected in any way.

The generality of this model is reflected in a few examples. The centralized model for supervised learning can be represented by the graph in Fig. 2, where each of the m learning agents has access to all exemplars in the training database. Fig. 3 illustrates an ensemble where a publicly available database is available to all the learning agents, each of which retains a private training set. Fig. 4 illustrates agents who access training examples that are "nearby" with respect to an underlying topology. This latter example may reflect what intuitively corresponds to a wireless sensor network, where a sensor (agent) has access to

[2]A host of localization algorithms have been developed to enable sensors to measure their location; see, for example, [17], [33], [39].

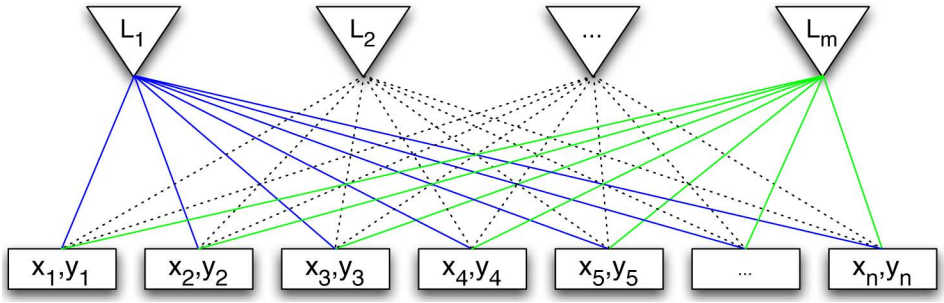

Fig. 2. A "centralized" ensemble.

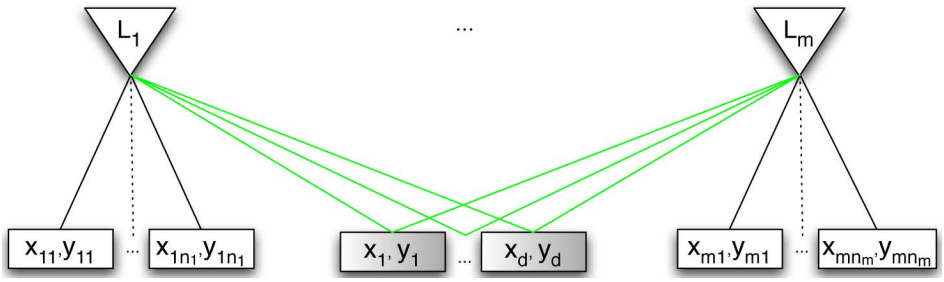

Fig. 3. An ensemble with a public database.

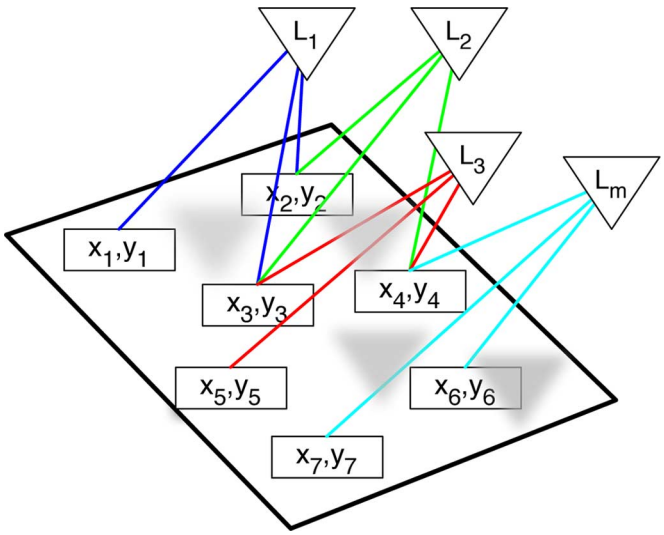

Fig. 4. A sensor network: an ensemble with topology dependent structure.

the data it measures but also to the data its neighbors measure. The general case depicted in Fig. 1 may model learning agents employed by individuals who have access to the data made accessible by trusted friends in a social network.

Much of the work in distributed learning differs in the way that the capacity of the links is modeled. Given that learning is already a complex problem, simple application-layer abstractions are typically preferred over detailed physical-layer models. The links are often assumed to support the exchange of "simple" real-valued messages, where simplicity is assessed relative to the application (e.g., sensors share summary statistics rather than entire data sets). Lacking a formal communication model, quantifying the efficiency of various methods from an energy and bandwidth perspective is not always straightforward.

As discussed in the Introduction, the importance of local communication transcends many analyses and implementations of wireless networks. Loosely speaking, local communication is that which occurs between neighboring sensors in a communication network. In wireless networks, topology of the network is in correspondence with the topology of the environment, which is to say that a sensor's network neighborhood is roughly equal to its physical neighborhood. By an inverse square law, the energy required for two sensors to exchange one bit of information decreases with the distance between them; by the same law, multiple-access interference decreases with the distance between pairs of communicating nodes. Thus, by minimizing energy expenditure and by enabling spectral reuse, local communication are often an efficient mode of information transport in WSNs.[3]

The foregoing observation is the starting point for many studies on distributed learning (and indeed, on distributed inference more generally). Rather than formalize a detailed physical-layer communication model, which may or may not be relevant to any specific WSN application, studies of distributed learning often posit a model for local communication and then study how sensor-to-sensor (sensor-to-data) interactions can improve learning by enabling collaboration. Ultimately, assumptions about the efficiency of local communication must be justified, perhaps by formalizing a physical model or through scaling law analyses. However, application-layer abstractions of local communication are nonetheless a reasonable starting point to investigate the fundamental limits of distributed learning.

B. Related Work

For learning rules motivated by the principle of empirical risk minimization, a training algorithm often must solve an optimization problem, e.g., (1). As a result, distributed and parallel optimization, fields with rich histories in their own right [8], [11], have an immediate bearing on distributed learning. Indeed, many tools from distributed and parallel optimization have

[3]These tradeoffs are studied more formally, for example, in the literature on scaling laws in WSNs; see, e.g., [20], [27] and references therein and thereto.

been applied to develop tools for distributed inference; see, for example, [13], [30], [43]–[46], [50].

One class of distributed training algorithms is constructed to exploit an assumed relationship between the topology of the wireless network and the correlation structure of sensors' measurements. In the toy example discussed above, for example, the temperature field may be slowly varying in space–time and thus it may be reasonable to assume that physically nearby sensors have similar temperature measurements. Since the sensors "exist" in the space–time feature space $\mathcal{X}$, the network topology is intimately related to the topology of feature space, and hence the correlation structure of the temperature field.

To see how such a relationship may be exploited in developing a distributed training algorithm, note that in least-squares estimation, the Representer Theorem shows that the minimizer g_n of (1) is implied by the solution to a system of linear equations

$$(K + \lambda I)\boldsymbol{c}_n = \boldsymbol{y} \tag{3}$$

where $K = (K_{ij})$ is the kernel matrix with $K_{ij} = K(x_i, x_j)$. If each sensor acquires a single training datum so that there is a one-to-one correspondence between training examples and sensors, then K is a matrix of sensor-to-sensor similarity measurements. For many kernels, K can have a sparsity structure[4] that admits distributed message-passing implementations of algorithms to solve the linear systems [18], [37]. When the sparsity structure of K is in appropriate correspondence with the topology of the network, the messages are passed between neighboring nodes in the network, and the result is a training algorithm that implements a classical learning rule in a distributed way.

For example, [19] adopts such an approach, and develops a training algorithm based on a distributed Gaussian elimination algorithm executed on a cleverly engineered junction tree. Developed within a very general framework for distributed inference in sensor networks [38], the approach is applicable in many applications, including some where the correspondence between the network topology and the correlation structure of the sensors' observations may not be intuitive. We refer the reader to [19] for additional detail and a description of several interesting experiments.

Assumptions that couple the network and the correlation structure of the sensors' observations are powerful, but may be of limited use, since it is easy to envision examples where those assumptions break down. For example, sensors deployed about a city may observe correlated measurements of traffic flow, despite being unable to communicate due to a signal-obstructing skyscraper; and data available to learning agents employed by individuals on Facebook may be unrelated to structure of the social network. In general, there is no fundamental, application-independent reason to assume a correspondence between the topology of the feature space $\mathcal{X}$ and the topology of the network.

[4]Generally speaking, sparse matrices are those with a large number of zero elements that allow for specialized processing techniques [1]. A discussion of sparse matrices and related computational methodsis outside the scope of this paper.

The training algorithm developed in this paper avoids such assumptions, and treats the network and the signal as distinct objects. The algorithm is constructed to solve a relaxation of the classical least-squares problem; the relaxation is motivated by the topological structure of the communication network, and is independent of the kernel, which retains its interpretation as a similarity measure on inputs. A statistical analysis shows that the generalization error afforded agents by the collaborative training algorithm depends on the relationship between the topology of the network and the *the representational capacity of* $\mathcal{H}_K$, the reproducing kernel Hilbert space induced by K.

Another approach to developing distributed training algorithms exploits the additive structure of the regularized empirical loss functional. To illustrate, suppose that agent (sensor) i has access to a single training datum $(x_i, y_i) \in S_n$, and for reasons that will soon become clear, let us rewrite (1) as

$$\min_{f \in \mathcal{H}_K} \left[\sum_{i=1}^{n} (f(x_i) - y_i)^2 + \sum_{i=1}^{n} \lambda_i \|f\|_{\mathcal{H}_K}^2 \right]. \tag{4}$$

When $\frac{1}{n}\sum_{i=1}^{n} \lambda_i = \lambda$, the (unique) minimizer of (4) is clearly equivalent to the minimizer of (1).

Gradient and subgradient methods (e.g., gradient descent) are popular iterative algorithms for solving optimization problems. In a centralized setting, the gradient descent algorithm for solving (4) defines a sequences of estimates

$$\hat{f}^{(k+1)} = \hat{f}^{(k)} - \alpha_k \frac{\partial F}{\partial f}\left(\hat{f}^{(k)}\right) \tag{5}$$

where $F(f) = \sum_{i=1}^{n} (f(X_i), Y_i) + \lambda \|f\|_{\mathcal{H}_K}^2$ is the objective function, and $\frac{\partial F}{\partial f}$ denotes its functional derivative. Note that $\frac{\partial F}{\partial f}(f^{(k)})$ factors due to its additive structure. *Incremental subgradient methods* exploit this additivity and define an alternative set of update equations

$$j_k = k \bmod m \tag{6}$$

$$\hat{f}^{(k+1)} = \hat{f}^{(k)} - \alpha_k \frac{\partial G_{j_k}}{\partial f}\left(\hat{f}^{(k)}\right) \tag{7}$$

where $G_j = (f(X_i), Y_i) + \lambda_j \|f\|_{\mathcal{H}_K}^2$. In other words, the update equations iterate over the n terms in F.

Incremental subgradient algorithms have been studied in detail in [31], [32]. Under reasonable regularity (e.g., bounded $\|\frac{\partial G_j}{\partial f}\|$), one can show that if $\alpha_k \to 0$, then $\|\hat{f}^{(k+1)} - g_n\|_{\mathcal{H}_K} \to 0$; with a constant step size (i.e., $\alpha_k = \alpha$), one can bound the number of iterations required to make $\|\hat{f}^{(k)} - g_n\|_{\mathcal{H}_K} \leq \epsilon$.

These ideas were exploited in [45], [46] to develop a message-passing algorithm that may be applied as a distributed training algorithm. After noting that the update equation at iteration k depends only on the data observed by sensor $k \bmod m$, a two-step process is proposed. First, a path is established that visits every sensor. Then, the incremental subgradient updates are executed by iteratively visiting each sensor along the path. For example, sensor one may initialize $\hat{f}^{(0)} = 0 \in \mathcal{H}_K$ and then compute $\hat{f}^1$ according to the update equations (which depend on sensor one's only training datum). Once finished, sensor one passes $\hat{f}^1$ on to the second sensor in the path, which performs a similar update before passing its estimate onto the third sensor.

The process continues over multiple passes through the network, at each stage, data is not exchanged—only the current estimates. By the comments above, only a finite number of iterations are required for *each* sensor to arrive at an estimate f with $\|f - g_n\|_{\mathcal{H}_K} \leq \epsilon$.

Notably, this idea is slightly different than the one originally conceived in [45], [46]. Whereas here we are interested in learning a function, there the focus was on estimating a real-valued parameter. From a theoretical perspective, this difference is primarily technical. However, practically speaking, the difference is important. The incremental subgradient method requires sensors to exchange a description of the parameter (i.e., the function), which by the Representer Theorem requires a description whose size is of the same order as the training set. Thus, the subgradient approach to training may not in general be more efficient than naive strategies which require sending "raw" data to a centralized processing facility. However, if $\mathcal{F}$ admits a lower dimensional parameterization—for example, if $\mathcal{F}$ is the reproducing kernel Hilbert space for the linear kernel—then messages may be communicated more efficiently to the tune of considerable energy savings. Note that [50] addressed a generalization of this incremental subgradient message-passing methodology by considering a clustered network topology.

The algorithm developed in this paper posits the exchange of simple real-valued messages, which may be significantly smaller than descriptions of functions. Specifically, sensors iteratively communicate real-valued "labels" of training data. Though a formal model is required to rigorously assess communication efficiency, this promises to make the present approach more broadly applicable.

Note that there are other approaches to distributed learning that similarly focus on developing distributed training algorithms. For example, in a data-mining context, [16], [28] developed a distributed extension of Adaboost [15]. In the context of boundary estimation in wireless sensor networks, [35] derived a hierarchical processing strategy through which sensors collaboratively prune a regression tree. The algorithm exploits additivity in the objective function of the complexity penalized estimator (i.e., an optimization similar in structure to (1)), and permits an interesting energy-accuracy analysis. Reference [36] derives a distributed expectation–maximization (EM) algorithm for density estimation in sensor networks. Though formally parametric, EM is popular for clustering problems and thus the approach may be broadly applicable. Reference [47] considered the existence of consistent learning algorithms in several models for distributed learning under communication constraints.

C. Other Work

Within the context of wireless sensor networks, [34] develops a nonparametric kernel-based methodology for decentralized detection. As in centralized learning, a training set is assumed available offline to a single processor. The data is used to train a learning rule that solves an optimization problem similar to (1), with the additional constraint that the resulting decision rule lies within a restricted class which is deployable across a sensor network; the powerful notion of a marginal kernel is exploited in the process. This setting is fundamentally different from the present context in that the data is centralized. Thus, one might distinguish the former topic of *centralized learning for decentralized inference* from the present topic of *distributed learning for decentralized inference*.

Reference [33] considered a clustered approach to distributed learning with a fusion center[5] to address sensor network localization. There, the feature space $\mathcal{X} = \Re^2$ models points in a planar terrain, and the output space $\mathcal{Y} = \{0, 1\}$ models whether or not a point belongs to (a specifically designed) convex region within the terrain. Training data is acquired from a subset of sensors (base stations) whose positions were estimated using various physical measurements. The fusion center uses reproducing kernel methods for learning, with a kernel designed using signal-strength measurements. The output is a rule for determining whether any sensor (i.e., non-base stations) lay in the convex region using only a vector of signal-strength measurements. We refer the reader to the paper for additional details, and reports on several real-world experiments. However, we highlight this as an example of a clustered approach to distributed learning in a parallel network with a fusion center, a methodology which is broadly applicable.

Finally, message-passing algorithms are a hot topic in many fields, wireless communications and machine learning notwithstanding. This surge in popularity is inspired in part by the powerful graphical model framework that has enabled many exciting applications and inspired new theoretical tools [5], [23], [26], [29], [37], [40], [41]. These tools are often applicable to signal processing in wireless sensor networks, since often the correlation structure of the phenomenon under observation (e.g., a temperature field) can be represented using a graphical model (e.g., Markov networks) and since inter-sensor communications are envisioned to occur over similar graphical structures. Indeed, graphical models and their application to wireless sensor networks are broad topics in their own right (see, e.g., [12]).

IV. A Collaborative Training Algorithm

In this section, we develop a local message-passing algorithm that admits an interpretation as a collaborative training algorithm. We rely on the model for distributed learning introduced in Section III.

A. The Algorithm

For any ensemble $\{S_n^j\}_{j=1}^m$, let the sets $\{\bar{S}_n^j\}_{j=1}^m$ be such that $\bar{S}_n^j \subseteq \{1, \ldots, n\}$, and that $i \in \bar{S}_n^j$ if and only if $(x_i, y_i) \in S_n^j$. In other words, $\bar{S}_n^j$ is the set of indices of training exemplars in S_n^j as enumerated in S_n. Analogously, let $\bar{S}_n = \{1, \ldots, n\}$.

Recall, as in (1), the classical kernel-linear least-squares regression estimator, which is taken to solve the following optimization problem:

$$\min_{f \in \mathcal{H}_K} \quad \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \lambda \|f\|_{\mathcal{H}_K}^2. \tag{8}$$

[5]Distributed clustering algorithms have been developed with such applications in mind; see [6] for example.

Here, the optimization variable (function) is f, which is constrained to be in the reproducing kernel Hilbert space $\mathcal{H}_K$ for a positive semidefinite kernel K.

Let us introduce a decision rule $f_j \in \mathcal{H}_K$ for each agent $j = 1, \ldots, m$, and consider the following constrained optimization program:

$$\begin{aligned} \min \quad & \sum_{i=1}^{n}(z_i - y_i)^2 + \sum_{j=1}^{m}\lambda_j \|f_j\|^2_{\mathcal{H}_K} && (9) \\ \text{s.t.} \quad & z_i = f_j(x_i) \qquad \forall i \in \bar{S}_n, j = 1, \ldots, m \\ & f_j \in \mathcal{H}_K \qquad j = 1, \ldots, m. && (10) \end{aligned}$$

In this program, the optimization variables are $\boldsymbol{z} \in \Re^n$ and $\{f_j\}_{j=1}^m \subset \mathcal{H}_K$; S_n and $\{\lambda_j\}_{j=1}^m \subset \Re$ are data. The constraints in (10) couple the decision rules by requiring that agents agree on the training data. More precisely, the *coupling constraints* dictate that a vector $(\boldsymbol{z}, f_1, \ldots, f_m)$ is feasible if and only if $f_j(x_i) = z_i = f_k(x_i)$ for $i = 1, \ldots, n$ and for $j, k = 1, \ldots, m$.

One may think about this program as being an equivalent representation of the centralized least-squares regression problem (8) in the following sense.

Lemma 1: Suppose that $(\boldsymbol{z}, g_n^1, \ldots, g_n^m) \in \Re^n \times \mathcal{H}_K^m$ is the solution of (9), that $g_n^\lambda \in \mathcal{H}_K$ is the solution of (8), that $\lambda_j > 0$ for $j = 1, \ldots, m$, and that $\lambda = \frac{1}{n}\sum_{j=1}^m \lambda_j$. Then $g_n = g_n^j$ for $j = 1, \ldots, m$.

A proof of Lemma 1 appears in the Appendix.

This equivalence suggests an association between *centralized regression* and *global agreement*, and motivates a similar association between *distributed regression* and *local agreement*. Consider the learning rule formulated by relaxing the coupling constraints in a way that requires the agents to agree, but only on training examples they share

$$\begin{aligned} \min \quad & \sum_{i=1}^{n}(z_i - y_i)^2 + \sum_{j=1}^{m}\lambda_j \|f_j\|^2_{\mathcal{H}_K} && (11) \\ \text{s.t.} \quad & z_i = f_j(x_i) \qquad \forall i \in \bar{S}_n^j, j = 1, \ldots, m \\ & f_j \in \mathcal{H}_K \qquad j = 1, \ldots, m. && (12) \end{aligned}$$

In this formulation, the coupling constraints dictate that a vector $(\boldsymbol{z}, f_1, \ldots, f_m)$ is feasible if and only if $f_j(x_i) = z_i = f_k(x_i)$ for $(x_i, y_i) \in S_n^j \cap S_n^k$ and for $j, k = 1, \ldots, m$; that is, if and only if every pair of sensor decision rules agree on exemplars that the sensors share.

Constraining learning rules to satisfy local agreement constraints can be defended by the principle of empirical risk minimization. Note that by convexity

$$\frac{1}{|S|}\sum_{j\in S}(f_j(x) - y)^2 \geq \frac{1}{|S|}\sum_{j\in S}\left(\frac{1}{|S|}\sum_{j\in S} f_j(x) - y\right)^2 \quad (13)$$

for any $x \in \mathcal{X}$, $y \in \mathcal{Y}$, and $S \subseteq \{1, \ldots, m\}$, and that the inequality is strict whenever $f_j(x) \neq \frac{1}{|S|}\sum_{j=1}^m f_j(x)$ for some $j \in S$. In other words, the average sensor's error on a given exemplar can be lowered whenever there is local disagreement on the label of the exemplar. Applying this argument to all exemplars in a training data set suggests that the average empirical error of the ensemble can always be lowered if the sensors disagree. This logic could be applied to defend global agreement: our assumption is that global agreement (i.e., centralized regression) is infeasible and propose local agreement as a goal given communication constraints.

One may think about this formulation as defining a set of m learning rules, one for each learning agent. In particular, suppose that $(\boldsymbol{z}, g_n', \ldots, g_n^m)$ minimizes (11). Though the coupling constraints suggest that g_n^j is a function of only the training examples in S_n^j, clearly, as part of a joint minimizer of (11), g_n^j in general depends on all the data. Thus, with $g_n^j : \mathcal{X} \times (\mathcal{X} \times \mathcal{Y})^n \to \Re$, $\{g_n^j\}_{n=1}^\infty$ is a learning rule as typically conceived [14].

To emphasize, the fully coupled formulation (9) requires global agreement, reduces to a centralized learning rule, and intuitively models the centralized ensemble depicted in Fig. 2. The relaxed formulation (11) requires local agreement, jointly defines m learning rules, and reflects the more general structure of distributed learning depicted in Fig. 1. We now show that (11) can be solved distributively using an algorithm that admits a natural interpretation as a collaborative training algorithm.

Let $\mathcal{H} = \Re^n \times \mathcal{H}_K^m$ be the Hilbert space with norm $\|(\boldsymbol{z}, f_1, \ldots, f_m)\|^2 = \|z\|_2^2 + \sum_{i=1}^m \lambda_i \|f_i\|^2_{\mathcal{H}_K}$. Note that (11) can be interpreted as the orthogonal projection (in $\mathcal{H}$) of the vector $(\boldsymbol{y}, 0, \ldots, 0) \in \mathcal{H}$ onto the set $C = \cap_{j=1}^m C_i \subset \mathcal{H}$, with

$$C_j = \{(\boldsymbol{z}, f_1, \ldots, f_m) : f_j(x_i) = z_i \qquad \forall i \in \bar{S}_n^j, \boldsymbol{z} \in \Re^n, \{f_j\}_{j=1}^m \subset \mathcal{H}_K\} \subset \mathcal{H}.$$

This observation is significant because it highlights the fact that this relaxation of the standard centralized kernel-linear least-squares estimator can be interpreted as a projection onto the intersection of m linear subspaces. As a result, successive orthogonal projection algorithms such as (2) can be used to solve the relaxed problem (11).

Note that computing $P_{C_j}(v) = \arg\min_{v' \in C_j} \|v - v'\|$ requires agent j to gather only examples within its locally accessible database. More precisely, for any $v = (\boldsymbol{z}, f_1, \ldots, f_m) \in \mathcal{H}$, $P_{C_j}(v) = (\boldsymbol{z}^\star, f_1^\star, \ldots, f_m^\star)$, where

$$\begin{aligned} f_k^\star &= f_k \qquad \forall k \neq j \\ f_j^\star &= \arg\min_{f \in \mathcal{H}_K} \sum_{i \in \bar{S}_n^j} (f(x_i) - z_i)^2 + \lambda_j \|f - f_j\|^2_{\mathcal{H}_K} \\ z_i^\star &= z_i \qquad \forall i \notin \bar{S}_n^j \\ z_i^\star &= f_j^\star(x_i) \quad \forall i \in \bar{S}_n^i. \end{aligned}$$

In other words, computing $P_{C_j}(v)$ leaves z_i unchanged for all $i \notin \bar{S}_n^j$ and leaves f_k unchanged for all $k \neq j$. The function associated with agent j, $f_j^\star$ can be computed using $f_j, \{x_i\}_{i \in \bar{S}_n^j}$ and the "message variables" $\{z_i\}_{i \in \bar{S}_n^j}$ (which have the role of the training data labels in the classical formulation). Tying these observations together, we are left with an algorithm for collaboratively training networks of kernel-linear least-squares regression estimators. The algorithm is summarized in psuedocode in Table I and depicted pictorially in Fig. 5. In words, the algorithm iterates over the agents, having each locally train and then update the messages variables which function as training data labels in subsequent interations. Multiple passes over the agents are made.

The asymptotic behavior of the collaborative training algorithm is implied by the analysis of the SOP algorithm, and is given by the following theorem.

TABLE I
AN ALGORITHM FOR TRAINING COLLABORATIVELY

Input: Ensemble $\{S_n^j\}_{j=1}^m$

Initialize: $\mathbf{z} = \mathbf{y}$

$g_{j,0} = 0,\ j = 1, \ldots, m$

Train: for $t = 1, \ldots, T$

for $j = 1, \ldots, m$

Compute:

$$g_n^{j,t} := \arg\min_{f \in \mathcal{H}_K} \left[\sum_{i \in \bar{S}_j} (f(x_i) - z_{i,})^2 + \lambda_j \| f - g_n^{j,t-1} \|_{\mathcal{H}_K}^2 \right]$$

Update:

$$z_i \leftarrow g_n^{j,t}(x_i) \ \ \forall i \in \bar{S}_n^j$$

end

end

Output: $\{g_n^{j,T}\}_{j=1}^m$, $\mathbf{z}_T := \mathbf{z}$

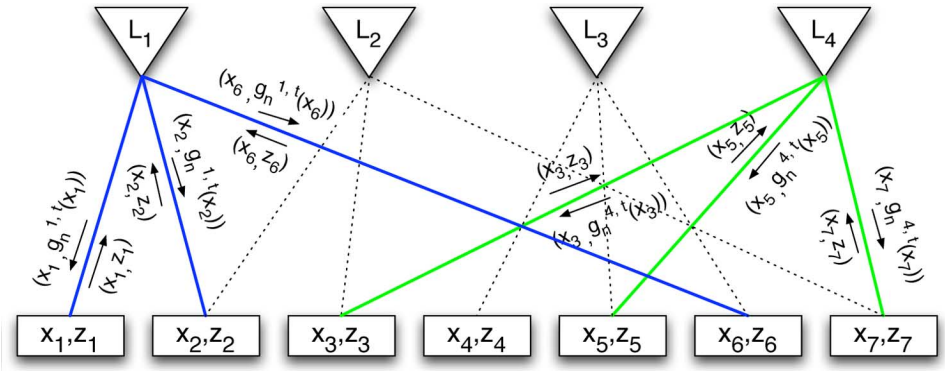

Fig. 5. A collaborative training algorithm.

Theorem 3: Suppose that $(\boldsymbol{z}, g_n^1, \ldots, g_n^m) \in \Re^n \times \mathcal{H}_K^m$ is the solution to (11) and let $\{g_n^{j,T}\}_{j=1}^m$ be as defined in Table I. Then

$$\lim_{T \to \infty} g_n^{j,T} = g_n^j \tag{14}$$

for $j = 1, \ldots, m$.

Theorem 3 follows immediately from Theorem 2 and the fact that convergence in norm implies pointwise convergence in RKHSs.

Observe that Theorem 3 characterizes the output of collaborative training algorithm relative to (11). This characterization is useful insofar as it sheds light on the relationship between the algorithm's output and (1), the centralized least-squares estimator. The following generalization of the Representer Theorem (Theorem 1) is a step toward further understanding this important relationship.

Theorem 4: Let $\{g_n^{j,T}\}_{j=1}^m$ be as defined in Table I. Then, for all $j = 1, \ldots, m$ and all $T \geq 1$, $g_n^{j,T}$ admits a representation of the form

$$g_n^{j,T}(\,\cdot\,) = \sum_{i \in \bar{S}_n^j} c_{n,i}^{j,T} K(\cdot, x_i). \tag{15}$$

for some $\boldsymbol{c}_n^{j,T} \in \Re^{|S_n^j|}$.

The proof of this theorem follows immediately from Theorem 4.2 and Remark 4.4 in [49]; we omit the details here. Note that since $\mathcal{H}_K$ is closed, it follows from Theorem 4 that $g_n^j = \lim_{T \to \infty} g_n^{j,T}$ admits a similar representation.

The significance of Theorem 4 lies in the fact that an agent's locally accessible database fundamentally limits the accuracy of that agent's estimate. In particular, an agent having access to only a few exemplars in an otherwise large training database will still be limited to estimates that lie in the span of functions determined by its locally observed data; in other words, local connectivity influences an estimator's *bias*. Intuitively, the message passing through the training database may optimize the estimator within that limited span if the ensemble is "connected" in some meaningful way. In the next subsection, we probe this intuition by considering a simple notion of connectedness that relates the topology of the network to the representational capacity of $\mathcal{H}_K$.

B. Relevance to Wireless Sensor Networks

Let us make explicit the relevance of this algorithm to distributed learning in wireless sensor networks. In particular, suppose that in the bipartite graph model for distributed learning, each agent (sensor) is connected to the data it measures, and in addition is connected to the data its neighbors measure. Here, the agents' neighborhoods are defined with respect to an arbitrary inter-agent graph, and is represented only implicitly in the

bipartite graph model for distributed learning. The collaborative training algorithm proceeds to iterate over each agent, at each step having an agent compute a fit to its locally observed data. Subsequently, the agent updates the labels in the training database.

Inter-agent communication is assumed to occur at two phases. First, in the *initialization phase*, neighboring sensors must exchange training data. This exchange occurs once, and can be viewed as a sunk cost. Secondly, communication occurs in the *update phase*, wherein sensors update the message variables. Since only the message variables (which function as training data labels) are updated—not feature-level descriptions of training data inputs—this is presumably less costly from both an energy and bandwidth perspective than communication in the initialization phase. Nevertheless, it occurs repeatedly. Communication in both phases is local, the level of which can be controlled by the number exemplars neighboring sensors share, and by the number of iterations T through the collaborative training algorithm.

Computationally, each agent is required to compute a local fit at each step. As dictated by Theorem 4, this requires each agent to solve a system of linear equations with dimensionality on the order of the number locally observed examples. Implicitly, this is related to the size of its neighborhood. So long as the number of exemplars is kept low, the computational burden placed on each agent is kept small.

Note that as described in Table I, the inner loop of the collaborative training algorithm iterates over agents in the ensemble serially. The ordering is unimportant and parallelism may be introduced. In fact, two agents can train simultaneously as long as they do not share exemplars in their locally accessible training databases. In practical settings, multiple-access algorithms (e.g., ALOHA) may be adapted to negotiate an ordering in a distributed fashion. Since successive orthogonal projection algorithms and Theorem 2 have been generalized to a very general class of (perhaps random) control orderings [7], Theorem 3 can be extended in many cases.

V. Generalization Error Analysis

In this section, we study the statistical behavior of the collaborative training algorithm. Our analysis is in the limit as $T \to \infty$, i.e., we assume that the network of agents have collaboratively solved (11).

For any ensemble, kernel pair $(\{S_n^j\}_{j=1}^m, K)$, let us construct an auxiliary inter-agent graph as follows: let there be a node in the graph for every learning agent, and let there be an edge between node (i.e., agent) j and node k if and only if

$$\begin{aligned}\operatorname{span}\left(\{K(\cdot,x_i)\}_{i\in\bar{S}_n^j}\right) &= \operatorname{span}\left(\{K(\cdot,x_i)\}_{i\in\bar{S}_n^k}\right)\\ &= \operatorname{span}\left(\{K(\cdot,x_i)\}_{i\in\bar{S}_n^j\cap\bar{S}_n^k}\right). \quad (16)\end{aligned}$$

In other words, by Theorem 4, an edge connects nodes j and k if and only if g_n^j and g_n^k admit representations in a span of functions determined by their shared training examples. Let us call the ensemble, kernel pair $(\{S_i\}_{i=1}^m, K)$ *connected* if and only if the inter-agent graph so constructed is connected.

This notion of connectedness leads us to Theorem 5, which is best viewed as a generalization of Lemma 1. The theorem follows from the observation that the auxiliary inter-agent graph for a connected ensemble, kernel pair $(\{S_i\}_{i=1}^m, K)$ is fully connected (by the transitivity of equality). See the Appendix for a proof.

Theorem 5: Suppose that $(\{S_n^j\}_{j=1}^m, K)$ is connected, that $(\boldsymbol{z}, g_n^1, \ldots, g_n^m) \in \Re^n \times \mathcal{H}_K^m$ is the solution to (11), and that g_n^λ denote the solution to (8) for $\lambda = \frac{1}{n}\sum_{j=1}^m \lambda_j$. Then

$$g_n = g_n^j \quad (17)$$

for $j = 1, \ldots, m$.

Theorem 5 is significant because it equates the centralized regression formulation (8) and the distributed regression formulation (11) under significantly sparser network topologies than the fully connected network that corresponds to the fully coupled formulation (9) that motivated our development. To illustrate, suppose that the learning agents employ the linear kernel. In this case, (16) is satisfied if agents j and k share $d+1$ linearly independent training examples. One can easily envision examples of sparse ensembles which together with the linear kernel are connected; consider, for example, the ensemble with a publicly available database (Fig. 3).

With this correspondence, we may study the statistical behavior of the collaborative training algorithm using known results in statistical learning theory.

Theorem 6: Suppose that $\mathcal{Y} = [0, B]$, that $K(x,x) \le \kappa$ for all $x \in \mathcal{X}$, and that $\{(X_i, Y_i)\}_{i=1}^n$ is independent and identically distributed (i.i.d.) with $(X_i, Y_i) \sim \boldsymbol{P}_{XY}$. Suppose further that $(\{S_n^j\}_{j=1}^n, K)$ is connected for all $n \ge 1$, and that $(\boldsymbol{z}, g_n^1, \ldots, g_n^m) \in \Re^n \times \mathcal{H}_K^m$ is the solution to (11). Then, for all $j = 1, \ldots, m$

$$\begin{aligned}\boldsymbol{E}\left\{\left(g_n^j(X) - Y\right)^2\right\} \le \frac{1}{n}\sum_{i=1}^n \left(g_n^j(X_i) - Y_i\right)^2\\ + \frac{4\kappa^2 B^2}{\sum_{j=1}^n \lambda_j} + \left(\frac{8\kappa^2 B^2}{\frac{1}{n}\sum_{j=1}^n \lambda_j} + 2B\right)\sqrt{\frac{\ln(1/\delta)}{2n}}\end{aligned}$$

with probability greater than $1 - \delta$.

Theorem 6 provides conditions on the network under which the agents' jointly defined learning rules generalize, that is, when the empirical loss of their estimates is a good approximation for their expected loss with high probability. Note that Theorem 6 bounds the generalization error of sensor j in terms of the empirical error of sensor j on the entire data set, not in terms of empirical error on the data to which sensor j has access. This is an interesting feature of the bound because it reveals cases where sensors that solve the relaxation generalize as though they have solved the classical unrelaxed problem, even though the collaborative training algorithm requires them only to observe a small fraction of the data. Since sensors cannot measure the empirical error on the entire set, the bound is not practical in the same way that generalization error bounds often are in classical (i.e., nondistributed) learning; in classical learning, generalization errors analysis can support things like parameter tuning by characterizing the uncertainty in the empirical error. Given Theorem 5, the result follows immediately from a known result in statistical learning theory; see [9, Sec. 5.2.2] for a proof.

Theorem 7: Suppose that the conditions of Theorem 6 hold, and that $\eta(x) = \boldsymbol{E}\{Y \mid X = x\} \in \mathcal{H}_K$. If $\{\lambda_j\}_{j=1}^m$ are chosen to depend on n such that $\frac{1}{n}\sum_{j=1}^{m_n} \lambda_j \to 0$ and $n(\frac{1}{n}\sum_{j=1}^{m_n} \lambda_j)^3 \to \infty$ as $n \to \infty$, then

$$\left\| g_n^j - \eta \right\|_{\mathcal{H}_K} \to 0 \tag{18}$$

in probability, for all $j = 1, \ldots, m$.

Given Theorem 5, this result also follows immediately from Theorem 1 in [48].

To illustrate the significance of this result, let us again recall the example of an ensemble with a public database. Suppose that such an ensemble, together with a linear kernel, is connected. Assuming the regression function is also linear, Theorem 7 implies that each agent can consistently estimate the regression function in the limit of the number of agents in network, as long each agent maintains at least one unique example outside the publicly available database. Thus, consistent learning is possible with collaboration, in the limit of the amount of information the network observes, even in cases where each agent observes a finite training set.

The preceding results depend critically on a notion of connectedness that couples the kernel K with the topology of the sensor network. This coupling appears contrary to the discussion in Section I, which advocated a decoupling of these objects. To clarify, our notion of connectivity relates the topology of the network with the *representational capacity of the reproducing kernel Hilbert space* implied by K (i.e., $\mathcal{H}_K$), and not the correlation structure encoded by the kernel. We also note that the collaborative training is valid under arbitrary kernels and network topologies; the notion of connectedness merely serves as a means to analyze the generalization error.

The strongest precondition of Theorems 6 and 7 is the requirement of connectedness. This condition will not be satisfied in many interesting cases; with the Gaussian kernel, for example, only fully coupled ensembles are connected. An interesting direction for future work is to understand the effect that collaboration has on generalization under weaker notions of connectivity and in more general network topologies. One approach is to generalize Theorem 5 under relaxed conditions relaxed conditions on the relation between the network and the representational capacity of the RKHS.

Note the dependence in both theorems on $\sum_{j=1}^m \lambda_j$. This correctly suggests that the parameters $\{\lambda_j\}_{j=1}^m$ must be tuned jointly, and opens the door for research on distributed regularization strategies. It is apparent that agent j should choose λ_j as a function of the structure of the network in which it is collaborating. However, extensions of Theorems 6 and 7 are necessary to formalize regularization strategies that generalize, and algorithmic development may be needed to facilitate implementation of such strategies in real networks.

VI. Experiments

A. The Data

The data in these experiments is generated artificially. We take $\mathcal{X} = \Re^{10}, \mathcal{Y} = \Re, X \sim \mathcal{N}(0, \sigma_X^2 I^{10\times 10})$ with $\sigma_X^2 = 1$, and $Y = \eta(X) + N$ with $\eta(x) = 2\sum_{i=1}^{10} x_i + 1$ and $N \sim \mathcal{N}(0, \sigma_N^2)$. There are two cases of interest. In *Case #1*, the noise variance $\sigma_N^2 = 4$; in *Case #2*, $\sigma_N^2 = 0$. In both Cases #1 and #2, the agents employ the linear kernel, so that $\mathcal{H}_K$ is the set of linear functions on $\mathcal{X}$. In all experiments, ensembles are randomly constructed in the following two-step process: first, an i.i.d. training S_n is assembled, and then, m learning agents are randomly connected to k of the n training examples in S_n.

B. The Method

When the collaborating training algorithm is employed, learning agent j's decision rule $g_n^{j,T}$ is implicitly dependent on $n, m, k, T, \{\lambda_j\}_{j=1}^m$ and the randomly generated ensemble $\{S_n^j\}_{j=1}^m$. Thus

$$g_n^{j,T}(X) = g_n^{j,T}\left(X, n, m, k, \{\lambda_j\}_{j=1}^m, S_n, \{S_n^j\}_{j=1}^m\right). \tag{19}$$

With J uniformly distributed on $\{1, \ldots, m\}$, we define

$$\mathrm{MSE} = \mathrm{MSE}\left(n, m, k, \{\lambda_j\}_{j=1}^m, T\right) \tag{20}$$

$$= \boldsymbol{E}\left\{\left|g_n^{J,T}(X) - \eta(X)\right|^2\right\} \tag{21}$$

where the expectation is taken with respect to J, X, S_n, and the random ensemble $\{S_n^j\}_{j=1}^m$. In other words, MSE is the expected mean-squared error of the agents' estimates.

C. Experiment #1: Centralized Regression

In this experiment, we explore the performance of the centralized regression estimator, which will subsequently be used as a baseline. This corresponds to choosing $m = 1$ and $S_n^1 = S_n$; in this setting, $T = 1$ is sufficient. For various $\lambda = \lambda_0$, we plot the MSE versus n. Fig. 6 depicts the result for Case #1, and Fig. 7 depicts the result for Case #2.

Note there there are 11 free parameters to be estimated when using the linear kernel on $\mathcal{X} = \Re^{10}$. Thus, in Case #2, the noiseless case, MSE drops off sharply after $n = 11$. The decay is more gradual in Case #1 since $\sigma_N^2 = 4$; we note, however, that beyond $n = 200$, the rate at which MSE decreases is small. In Case #1, the performance of the centralized estimator is insensitive to the choice of λ. In Case #2, a smaller λ leads to lower MSE, as expected given the noise-free data.

D. Experiment #2: Convergence Rate

In this experiment, we explore the performance of the collaborative training algorithm as a function of T, the number of iterations through the network. We take $m = 500, n = 200, k = 15$, and select various $\lambda_j = \lambda_0$ for $j = 1, \ldots, m$. We consider how

$$\|\boldsymbol{z}_T\|_2^2 + \sum_{j=1}^m \lambda_j \left\|g_n^{j,T}\right\|_{\mathcal{H}_K}^2 - \left(\|\boldsymbol{z}_{100}\|_2^2 + \sum_{j=1}^m \lambda_j \left\|g_n^{j,100}\right\|_{\mathcal{H}_K}^2\right) \tag{22}$$

varies with $T \in \{1, \ldots, 100\}$. Equation (22) is expected to be monotonically decreasing in T by Theorem 2, and allows us

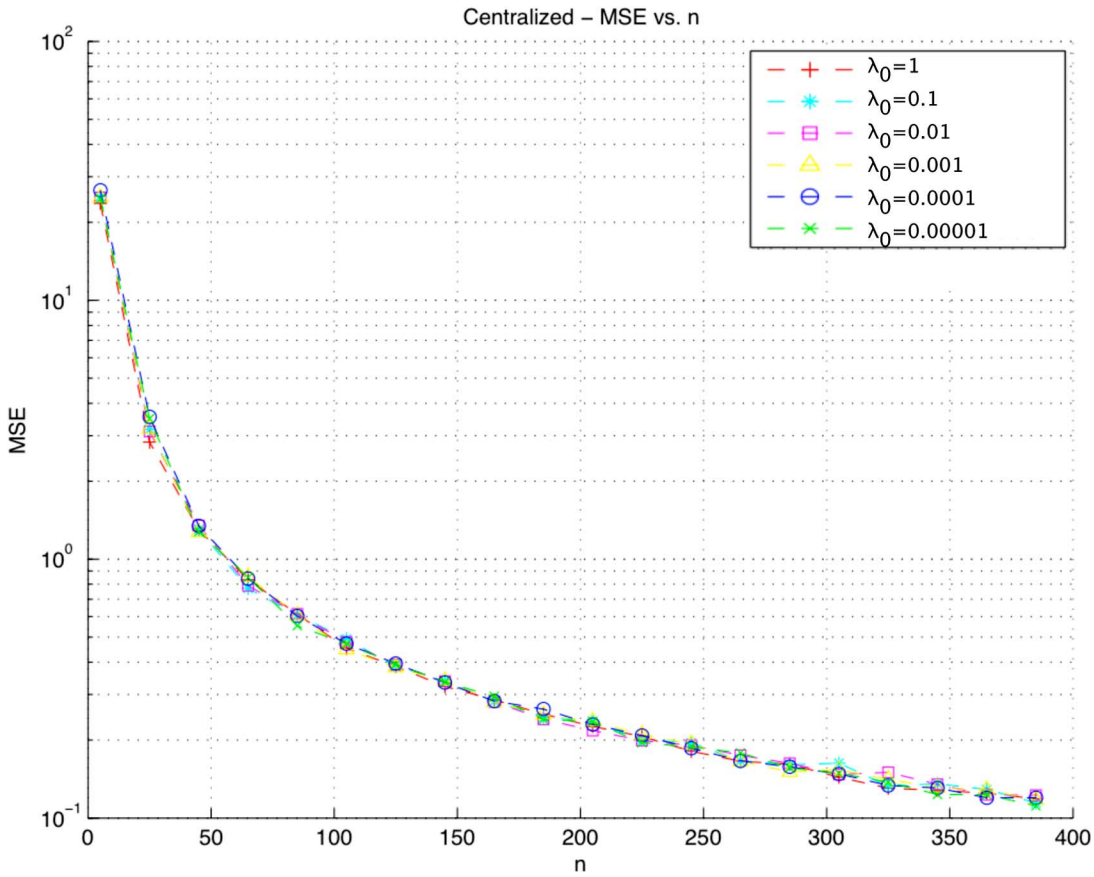

Fig. 6. Centralized, Case 1: MSE versus n.

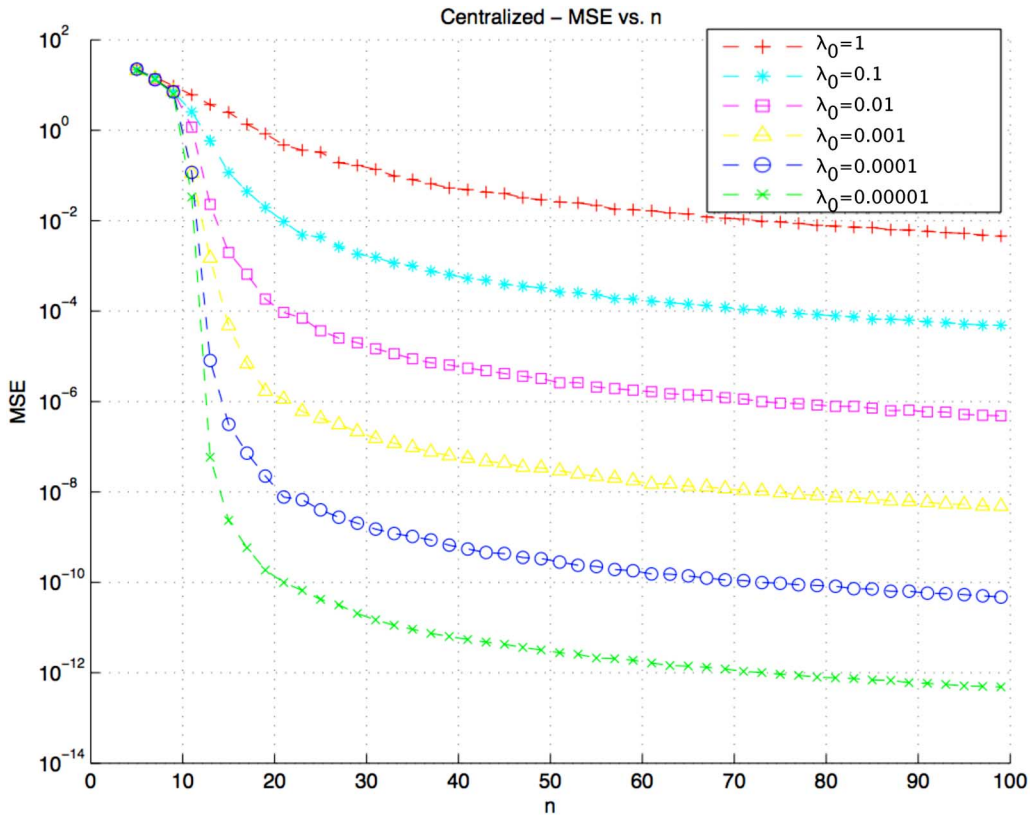

Fig. 7. Centralized, Case 2: MSE versus n.

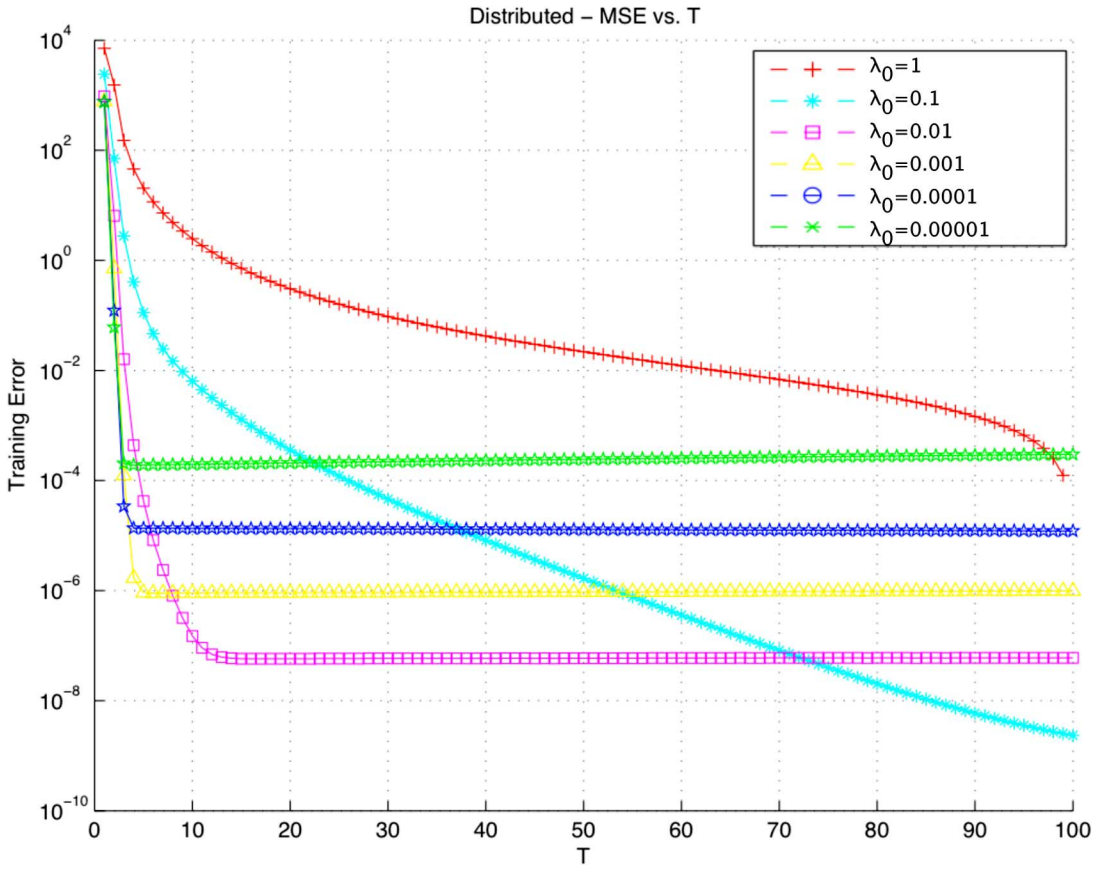

Fig. 8. Distributed, Case 1: MSE versus T.

to assess the rate of convergence of the collaborative training algorithm.

The result is depicted in Fig. 8, averaged over 200 realizations of S_n and $\{S_n^j\}_{j=1}^m$. We notice an unexpectedly large variation in behavior as a function of λ_0. In experiments not documented here, we notice similar behavior throughout the range of n, m, k relevant to the remaining experiments, and in Case #2. We henceforth fix $T = 25$, having observed that after 25 iterations through the network, the algorithm typically converges.

We emphasize that Fig. 8 should not be interpreted as an absolute measurement of execution time. In particular, the inner loop of the collaborative training algorithm iterates over sensors, and thus for fixed T, the execution time is expected to grow linearly with m.

E. Experiment #3: Collaboration & Generalization

In this experiment, we explore the generalization error afforded agents by the collaborative training algorithm as a function of network connectivity. In Case #1, we take $n = 200, k = 15, T = 25$, and plot MSE versus m for various $\lambda_j = \lambda_0\,(j = 1\ldots, m)$. As m increases, the network will become increasingly connected; intuitively, the amount of information propagating through the network increases with m via collaboration. Since $k > 11$, the ensemble paired with the linear kernel is expected to be connected (in the sense discussed in Section V) for sufficiently large m. Thus, as m increases, we expect the generalization error to approach the performance the centralized rule for $\lambda = \frac{1}{n}\sum_{j=1}^m \lambda_j$; recall that Fig. 6 depicts the generalization error of the centralized estimator. Since n is fixed and m grows, λ will grow with m, and thus, we expect that MSE will increase for sufficiently large m, as the complexity term begins to dominate.

These expectations are borne out in Fig. 9, where MSE is plotted by averaging 200 random realizations of S_n, X, and $\{S_n^j\}_{j=1}^m$. The horizontal lines in Fig. 9 depict the MSE of a single learning agent trained on a randomly selected a training set of $k = 15$ exemplars. This is an interesting point of comparison, since it is exactly the collaborative training algorithm without the update stage. In all cases, the collaborative training algorithm provides a significant improvement in MSE over the noncollaborating approach. This may be interpreted as illustrating the potential value of the collaborating training algorithm in reducing noise.

In Case #2, we take $n = 50, k = 7, T = 25$, and similarly plot MSE versus m for various λ_0. Since the training data labels are noiseless in Case #2 and since $k < 11$, we do not expect collaboration to improve the error rate. Indeed, the update stage of the algorithm can only introduce noise into the noiseless data, and thus can only increase MSE. This expectation is borne out in Fig. 10.

VII. Summary, Extensions, and Future Work

In this paper, we have developed an algorithm for collaboratively training networks of kernel-linear least-squares regression estimators. The algorithm has been shown to distributively

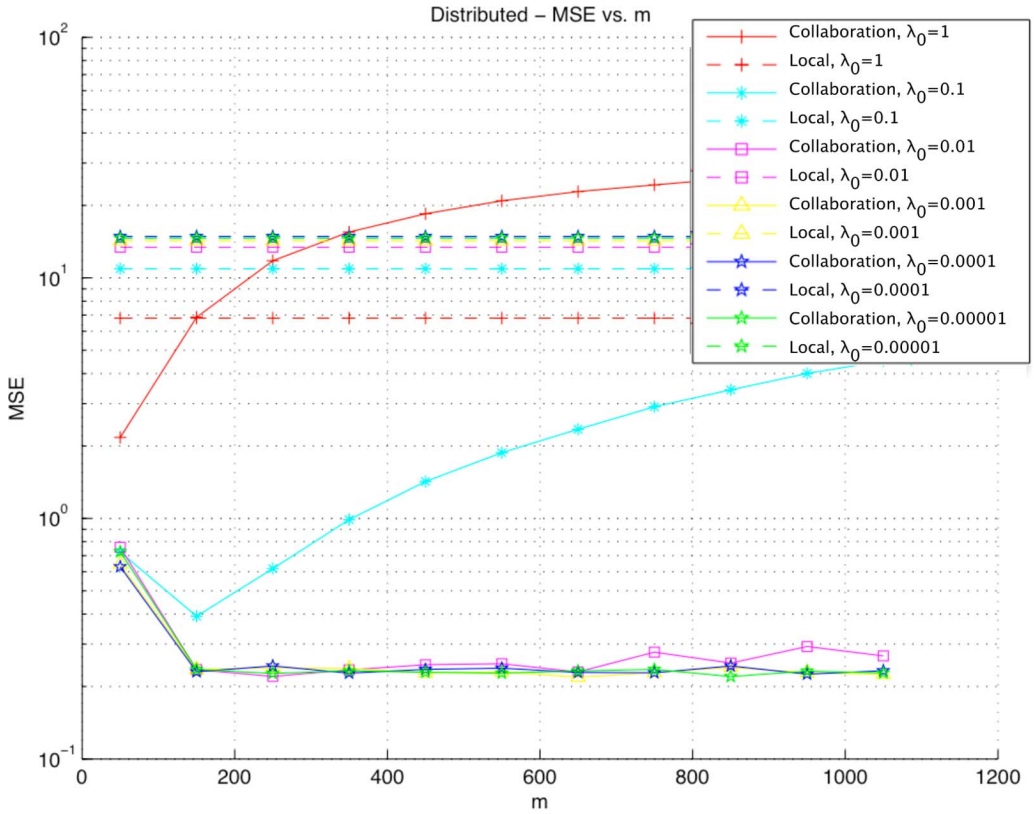

Fig. 9. Distributed, Case 1: MSE versus M.

solve a relaxation of the classical centralized kernel-linear least-squares regression problem. A statistical analysis has shown that the generalization error afforded agents by the collaborative training algorithm can be bounded in terms of the relationship between the network topology and the representational capacity of the relevant reproducing kernel Hilbert space. Numerical experiments have shown that the collaboration is effective at reducing noise. The algorithm was argued to be relevant to the problem of distributed learning in wireless sensor networks by virtue of its exploitation of local communication.

This paper has focused exclusively on the problem of kernel-linear least-squares regression. The collaborative training algorithm and all related results can be extended to more general convex loss functions and arbitrary convex function spaces. In particular, note that the coupling agreement constraints introduce a sparse set of linear constraints. In the least-squares context, successive orthogonal projection algorithms happen to be a useful tool for computing the estimators distributively. More generally, row-action algorithms [11] are applicable to minimizing convex functions over sparse sets of linear constraints. In particular, Bregman's algorithm [10] is expected to be relevant to constructing a more general collaborative training algorithm.

The formulation of distributed least-squares regression as a relaxation of the classical centralized least-squares under local agreement constraints bears resemblance to the generalized consensus formulation introduced in [2].[6] In the context of kernel-linear regression, the consensus formulation requires agents to agree with neighbors about the entire learned function (which by Lemma 1 is equivalent to global agreement); whereas in the present formulation, neighboring agents are merely required to agree on how the learned function evaluates on shared exemplars (local agreement). Both the algorithms developed here and in [2] require agents to communicate with neighbors in order to satisfy agreement constraints, and as discussed, our assumption is that the communication costs to meet global agreement constraints are infeasible in the context of learning with kernels.

Those familiar with the online learning framework may find our collaborative training algorithm reminiscent of the equations for additive gradient updates [25]. Though both algorithms may be interpreted in the context of successive orthogonal projection algorithms, it does not appear possible to specialize the bipartite graph model for distributed learning in a way that recovers the online learning framework (or vice versa).

Finally, those familiar with low-density parity check (LDPC) codes or Bayesian networks may find the current model and algorithm reminiscent of message-passing algorithms, such a belief propagation, which are frequently studied in those fields; variational interpretations of kernel methods in the context of Gaussian processes further suggests a relationship between these works. Formalizing such a connection would likely require one to interpret our relaxation in the context of dependency structures in Gaussian processes, and to connect alternating projection algorithms with the generalized distributive law [5].

The collaborative training algorithm developed in this paper was developed via a somewhat *ad hoc* relaxation of the classical least-squares regression estimator. However, its geometric interpretation and the statistical analysis support the algorithm

[6]We thank any anonymous referee for pointing out this connection.

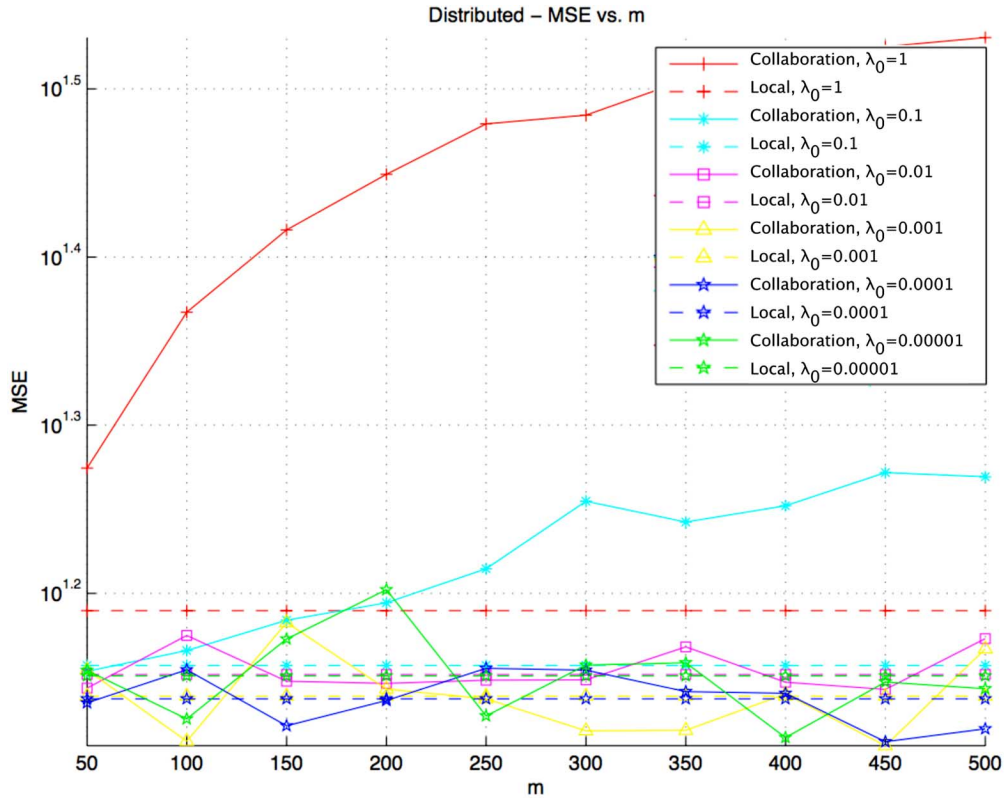

Fig. 10. Distributed, Case 2: MSE versus M.

as being reasonable. Nonetheless, the motivating association between distributed learning and local agreement inspires deeper questions for statistical learning theory. Classical statistical learning theory often considers the generalization error of learning algorithms that minimize empirical risk over a class of decision rules. The algorithm considered in this paper essentially attempts a similar minimization, under additional constraints on interagent agreement. Can generalization bounds be derived for networks of empirical risk minimizers that also strive for consensus? The statistical analysis in this paper suggests that such bounds can be derived, though the results rely on a strong notion of connectivity. Future work should seek general principles for learning under agreement, and generalization bounds for learning algorithms so derived. These directions are relevant to the study of distributed learning, since the local message-passing algorithm developed in this paper suggests that agreement can sometimes be achieved using only local communication. A related and seemingly relevant body of research concerns gossip or consensus algorithms.

APPENDIX

Proof of Lemma 1: If $(\boldsymbol{z}, g_n^1, \ldots, g_n^m)$ minimizes (9), then clearly g_n^j also minimizes

$$\begin{aligned} \min \quad & \|f\|_{\mathcal{H}_K}^2 \\ \text{s.t.} \quad & z_i = f(x_i) \quad \forall i \in \bar{S}_n \\ & f \in \mathcal{H}_K. \end{aligned} \tag{23}$$

for $j = 1, \ldots, m$. The solution to (23) is unique, and therefore $g_n^1 = \cdots = g_n^m$. After eliminating $\boldsymbol{z}$ from (9), we can rewrite (9) with a much stronger set of coupling constraints

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} (f_1(x_i) - y_i)^2 + \sum_{j=1}^{m} \lambda_j \|f_j\|_{\mathcal{H}_K}^2 \\ \text{s.t.} \quad & f_1 = \cdots = f_m \in \mathcal{H}_K. \end{aligned}$$

Now, it is clear that $g_n^j = g_n$ if $n\lambda = \sum_{j=1}^m \lambda_j$, for $j = 1 \ldots, m$. This completes the proof. □

Proof of Theorem 5: Suppose that $(\boldsymbol{z}, g_n^1, \ldots, g_n^m)$ minimizes (9), and that learning agents j and k are neighbors in the auxiliary graph constructed from $(\{S_n^j\}_{j=1}^m, K)$.

By the connectedness of $(\{S_n^j\}_{j=1}^m, K)$ and Theorem 4, there exists a $\boldsymbol{c}_n^j \in \Re^{|S_n^j \cap S_n^k|}$ and $\boldsymbol{c}_n^k \in \Re^{|S_n^j \cap S_n^k|}$ such that

$$g_n^j(\cdot) = \sum_{i \in \bar{S}_n^j \cap \bar{S}_n^k} c_{n,i}^j K(\cdot, x_i)$$

and

$$g_n^k(\cdot) = \sum_{i \in \bar{S}_n^j \cap \bar{S}_n^k} c_{n,i}^k K(\cdot, x_i).$$

Moreover

$$K^{jk} \boldsymbol{c}_n^j = K^{jk} \boldsymbol{c}_n^k = \boldsymbol{z}^{jk} \tag{24}$$

where $K^{jk} \in \Re^{|S_n^j \cap S_n^k| \times |S_n^j \cap S_n^k|}$ is the restriction of the kernel matrix K to the examples shared by agents j and k; $\boldsymbol{z}^{jk} =$

$(z_i)_{i\in\bar{S}_n^j\cap\bar{S}_n^k}$, i.e., the restriction of $\boldsymbol{z}$ to examples shared by agents j and k. Since the kernel is positive definite, it follows that $\boldsymbol{c}_n^j=\boldsymbol{c}_n^k=(K^{jk})^{-1}\boldsymbol{z}^{jk}$ and thus $g_n^j=g_n^k$.

The preceding argument holds for any pair of neighboring agents. Since the auxiliary graph is connected, it thereby follows that $g_n^j=g_n^k$ for *all* $j,k=1,\dots,m$. Thus, after eliminating $\boldsymbol{z}$ from (9), we can therefore rewrite (9) with a much stronger set of coupling constraints

$$\begin{aligned}\min\quad & \sum_{i=1}^{n}(f_1(x_i)-y_i)^2+\sum_{j=1}^{m}\lambda_j\|f_j\|_{\mathcal{H}_K}^2\\ \text{s.t.}\quad & f_1=\cdots=f_m\in\mathcal{H}_K.\end{aligned}$$

Now, it is clear that $g_n^j=g_n$ if $n\lambda=\sum_{j=1}^m\lambda_j$, for $j=1\dots,m$. The proof is complete. □

ACKNOWLEDGMENT

This research was completed while Joel B. Predd was a Ph.D. candidate at Princeton University.

REFERENCES

[1] T. Davis, "Sparse Matrix. From MathWorld—A Wolfram Web Resources," Aug. 16, 2008 [Online]. Available: http://mathworld.wolfram.com/SparseMatrix.html, created by Eric W. Weisstien

[2] M. G. Rabbat, R. D. Nowak, and J. A. Bucklew, "Generalized consensus computation in networked systems with erasure links," in *Proc. IEEE Workshop on Signal Processing Advances in Wireless Communications*, New York, Jun. 2005, pp. 1088–1092.

[3] T. Kailath, "An RKHS approach to detection and estimation problems. Pt. I: Deterministic signals in Gaussian noice," *IEEE Trans. Inf. Theory*, vol. IT-17, no. 5, pp. 530–549, Sep. 1971.

[4] N. Aronszajn, "Theory of reproducing kernels," *Trans. Amer. Math. Soc.*, vol. 68, no. 3, pp. 337–404, 1950.

[5] S. M. Aji and R. J. McEliece, "The generalized distributive law," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 325–343, Mar. 2000.

[6] S. Bandyopadhyay and E. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *Proc. 22nd Annu. Joint Conf. IEEE Computer and Communications Societies (Infocom)*, San Francisco, CA, Mar./Apr. 2003, vol. 3, pp. 1713–1723.

[7] H. H. Bauschke and J. M. Borwein, "On projection algorithms for solving convex feasibility problems," *SIAM Rev.*, vol. 38, no. 3, pp. 367–426, Sep. 1996.

[8] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Belmont, MA: Athena Scientific, 1997.

[9] O. Bousquet and A. Elisseeff, "Stability and generalization," *J. Mach. Learning Res.*, vol. 2, pp. 499–526, 2002.

[10] L. M. Bregman, "The relaxation method of finding a common point of convex sets andits application to the solution of problems in convex programming," *U. S. S. R. Comput. Math. Math. Phys.*, vol. 78, no. 384, pp. 200–217, 1967.

[11] Y. Censor and S. A. Zenios, *Parallel Optimization: Theory, Algorithms, and Applications*. New York: Oxford Univ. Press, 1997.

[12] M. Cetin, L. Chen, J. W. Fisher III, A. T. Ihler, R. L. Moses, M. J. Wainwright, and A. S. Willsky, "Distributed fusion in sensor networks: A graphical models perspective," *IEEE Signal Process. Mag. (Special Issue on Distributed Signal Processing in Wireless Sensor Networks)*, vol. 23, no. 4, pp. 42–55, Jul. 2006.

[13] V. Delouille, R. Neelamani, and R. Baraniuk, "Robust distributed estimation in sensor networks using the embedded polygons algorithm," in *Proc. 3rd Int. Symp. Information Processing in Sensor Networks*, Berkeley, CA, Apr. 2004, pp. 405–413.

[14] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*. New York: Springer-Verlag, 1996.

[15] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Comput. Syst. Scie.*, vol. 55, no. 1, pp. 119–139, 1997.

[16] S. Gambs, B. Kégl, and E. Aïmeur, "Privacy-Preserving Boosting," 2005, preprint.

[17] S. Gezici, Z. Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 70–84, Jul. 2005.

[18] G. Golub and C. Van Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins Univ. Press, 1989.

[19] C. Guestrin, P. Bodi, R. Thibau, M. Paskin, and S. Madde, "Distributed regression: An efficient framework for modeling sensor network data," in *Proc. 3rd Int. Symp. Information Processing in Sensor Networks*, Berkeley, CA, Apr. 2004, pp. 1–10.

[20] P. Gupta and P. R. Kumar, "Capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–401, Mar. 2000.

[21] L. Györfi, M. Kohler, A. Krzyżak, and H. Walk, *A Distribution-Free Theory of Nonparametric Regression*. New York: Springer-Verlag, 2002.

[22] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer-Verlag, 2001.

[23] , M. Jordan, Ed., *Learning in Graphical Models*. Cambridge, MA: MIT Press, 1999.

[24] G. Kimeldorf and G. Wahba, "Some results on Tchebycheffian spline functions," *J. Math. Anal. Applic.*, vol. 33, no. 1, pp. 82–95, 1971.

[25] J. Kivinen and M. K. Warmuth, "Additive versus exponentiated gradient updates for linear prediction," *Inf. Comput.*, vol. 132, no. 1, pp. 1–64, 1997.

[26] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[27] S. R. Kulkarni and P. Viswanath, "A deterministic approach to throughput scaling in wireless networks," *IEEE Trans. Inf. Theory*, vol. 50, no. 6, pp. 1041–1049, Jun. 2004.

[28] A. Lazarevic and Z. Obradovic, "The distributed boosting algorithm," in *Proc. 7th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, San Francisco, CA, 2001, pp. 311–316, published by ACM Press.

[29] H. A. Loeliger, "An introduction to factor graphs," *IEEE Signal Process. Mag.*, vol. 21, no. 1, pp. 28–41, Jan. 2004.

[30] C. C. Moallemi and B. Van Roy, "Distributed optimization in adaptive networks," in *Advances in Neural Information Processing Systems 16*, S. Thrun, L. Saul, and B. Schölkopf, Eds. Cambridge, MA: MIT Press, 2004.

[31] A. Nedic and D. Bertsekas, Incremental Subgradient Methods for Nondifferentiable Optimization MIT, Cambridge, MA, Tech. Rep. LIDS-P-2460, 1999.

[32] A. Nedic and D. Bertsekas, "Convergence rate of incremental subgradient algorithms," in *Stochastic Optimization: Algorithms and Applications*, S. Uryasev and P. M. Pardalos, Eds. Dordrecht, The Netherlands: Kluwer, 2000, pp. 263–304.

[33] X. Nguyen, M. I. Jordan, and B. Sinopoli, "A kernel-based learning approach to ad hoc sensor network localization," *ACM Trans. Sensor Networks*, vol. 1, no. 1, pp. 134–152, 2005.

[34] X. Nguyen, M. J. Wainwright, and M. I. Jordan, "Nonparametric decentralied detection using kernel methods," *IEEE Trans. Signal Process.*, vol. 53, no. 11, pp. 4053–4066, Nov. 2005.

[35] R. Nowak and U. Mitra, "Boundary estimation in sensor networks: Theory and methods," in *Proc. 2nd Int. Workshop on Information Processing in Sensor Networks*, Palo Alto, CA, Apr. 22–23, 2003, pp. 80–95.

[36] R. D. Nowak, "Distributed EM algorithms for density estimation and clustering in sensor networks," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2245–2253, Aug. 2003.

[37] M. A. Paskin and G. D. Lawrence, Junction Tree Algorithms for Solving Sparse Linear Systems EECS Dep., Univ.California, Berkeley, Tech. Rep. UCB/CSD-03-1271, 2003.

[38] M. A. Paskin, C. E. Guestrin, and J. McFadden, "A robust architecture for inference in sensor networks," in *Proc. 4rth Int. Symp. Information Processing in Sensor Networks*, UCLA, Los Angeles, CA, Apr. 2005, pp. 55–62.

[39] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, "Locating the nodes: Cooperative localization in wireless sensor networks," *IEEE Signal Process. Mag.* , vol. 22, no. 4, pp. 54–69, Jul. 2005.

[40] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA: Morgan Kaufmann, 1988.

[41] K. Plarre and P. R. Kumar, "Extended message passing algorithm for inference in loopy Gaussian graphical models," *Ad Hoc Networks*, vol. 2, no. 2, pp. 153–169, 2004.

[42] H. V. Poor, *An Introduction to Signal Detection and Estimation*, 2nd ed. New York: Springer-Verlag, 1994.
[43] J. B. Predd, S. R. Kulkarni, and H. V. Poor, "Regression in sensor networks: Training distributively with alternating projections," in *Proc. SPIE Conf. Advanced Signal Processing Algorithms, Architectures, and Implementations XV*, San Diego, CA, Jul./Aug. 2005, pp. 591006-1–591006-15, Invited Paper.
[44] J. B. Predd, S. R. Kulkarni, and H. V. Poor, "Distributed kernel regression: An algorithm for training collaboratively," in *Proc. 2006 IEEE Information Theory Workshop*, Punta del Este, Uruguay, Mar. 2006.
[45] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proc. 3rd Int. Symp. Information Processing in Sensor Networks*, Berkeley, CA, Apr. 2004, pp. 20–27.
[46] M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 798–808, Apr. 2005.
[47] J. B. Predd, S. R. Kulkarni, and H. V. Poor, "Consistency in models for distributed learning under communication constraints," *IEEE Trans. Inf. Theory*, vol. 52, no. 1, pp. 52–53, Jan. 2006.
[48] C. Rudin, Stability Analysis for Regularized Least Squares Regression 2005, arXiv: cs.LG/0502016.
[49] B. Schölkopf and A. Smola, *Learning with Kernels*, 1st ed. Cambridge, MA: MIT Press, 2002.
[50] S.-H. Son, M. Chiang, S. R. Kulkarni, and S. C. Schwartz, "The value of clustering in distributed estimation for sensor networks," in *Proc. IEEE Int. Conf. Wireless Networks, Communications, and Mobile Computing*, Maui, HI, Jun. 2005, vol. 2, pp. 969–974.
[51] J. von Neumann, *Function Operators II*. Princeton, NJ: Princeton Univ. Press, 1950.
[52] G. Wahba, *Spline Models for Observational Data*. Philadelphia, PA: SIAM, 1990.

Joel B. Predd (S'98–M'01–SM'02) received the B.S. degree in electrical engineering from Purdue University, West Lafayette, IN, in 2001 and the M.A. and Ph.D. degrees in electrical engineering from Princeton University, Princeton, NJ, in 2004 and 2006, respectively.

He is a policy researcher at the RAND Corporation. His primary research interests include information technology and information technology policy, methodologies for combining and eliciting expert opinion, and machine learning. Some of his recent projects at RAND have considered counter-IED operational analysis, the relation between avionics system complexity and cost, the implications of human decision-making for insider threats to information systems, and new methodologies for eliciting expert opinion. He spent the Summer of 2004 visiting National ICT Australia in Canberra.

Sanjeev R. Kulkarni (M'91–SM'96–F'04) received the B.S. degree in mathematics, the B.S. degree in electrical engineering, the M.S. degree in mathematics from Clarkson University, Potsdam, NY, in 1983, 1984, and 1985, respectively, the M.S. degree in electrical engineering from Stanford University, Stanford, CA, in 1985, and the Ph.D. degree in electrical engineering from the Massachusetts Institute of Technology (MIT), Cambridge, in 1991.

From 1985 to 1991, he was a Member of the Technical Staff at MIT Lincoln Laboratory, Lexington, MA. Since 1991, he has been with Princeton University, Princeton, NJ, where he is currently Professor of Electrical Engineering, and an affiliated faculty member in the Department of Operations Research and Financial Engineering and the Department of Philosophy. He spent January 1996 as a Research Fellow at the Australian National University, Canberra, 1998 with Susquehanna International Group, and Summer 2001 with Flarion Technologies, Bridgewater, NJ. His research interests include statistical pattern recognition, nonparametric statistics, learning and adaptive systems, information theory, wireless networks, and image/video processing.

Prof. Kulkarni received an ARO Young Investigator Award in 1992, an NSF Young Investigator Award in 1994, and several teaching awards at Princeton University. He has served as an Associate Editor for the IEEE TRANSACTIONS ON INFORMATION THEORY.

H. Vincent Poor (S'72–M'77–SM'82–F'87) received the Ph.D. degree in electrical engineering and computer science from Princeton University, Princeton, NJ, in 1977.

From 1977 until 1990, he was on the faculty of the University of Illinois at Urbana-Champaign, Urbana. Since 1990, he has been on the faculty at Princeton University, where he is the Dean of Engineering and Applied Science, and the Michael Henry Strater University Professor of Electrical Engineering. His research interests are in the areas of stochastic analysis, statistical signal processing and their applications in wireless networks, and related fields. Among his publications in these areas are the recent books *MIMO Wireless Communications* (Cambridge University Press, 2007), coauthored with Ezio Biglieri, *et al.*, and *Quickest Detection* (Cambridge University Press, 2009), coauthored with Olympia Hadjiliadis.

Dr. Poor is a member of the National Academy of Engineering, a Fellow of the American Academy of Arts and Sciences, and a former Guggenheim Fellow. He is also a Fellow of the Institute of Mathematical Statistics, the Optical Society of America, and other organizations. In 1990, he served as President of the IEEE Information Theory Society, and in 2004–2007 as the Editor-in-Chief of these TRANSACTIONS. He was the recipient of the 2005 IEEE Education Medal. Recent recognition of his work includes the 2007 IEEE Marconi Prize Paper Award, the 2007 Technical Achievement Award of the IEEE Signal Processing Society, and the 2008 Aaron D. Wyner Distinguished Service Award of the IEEE Information Theory Society.