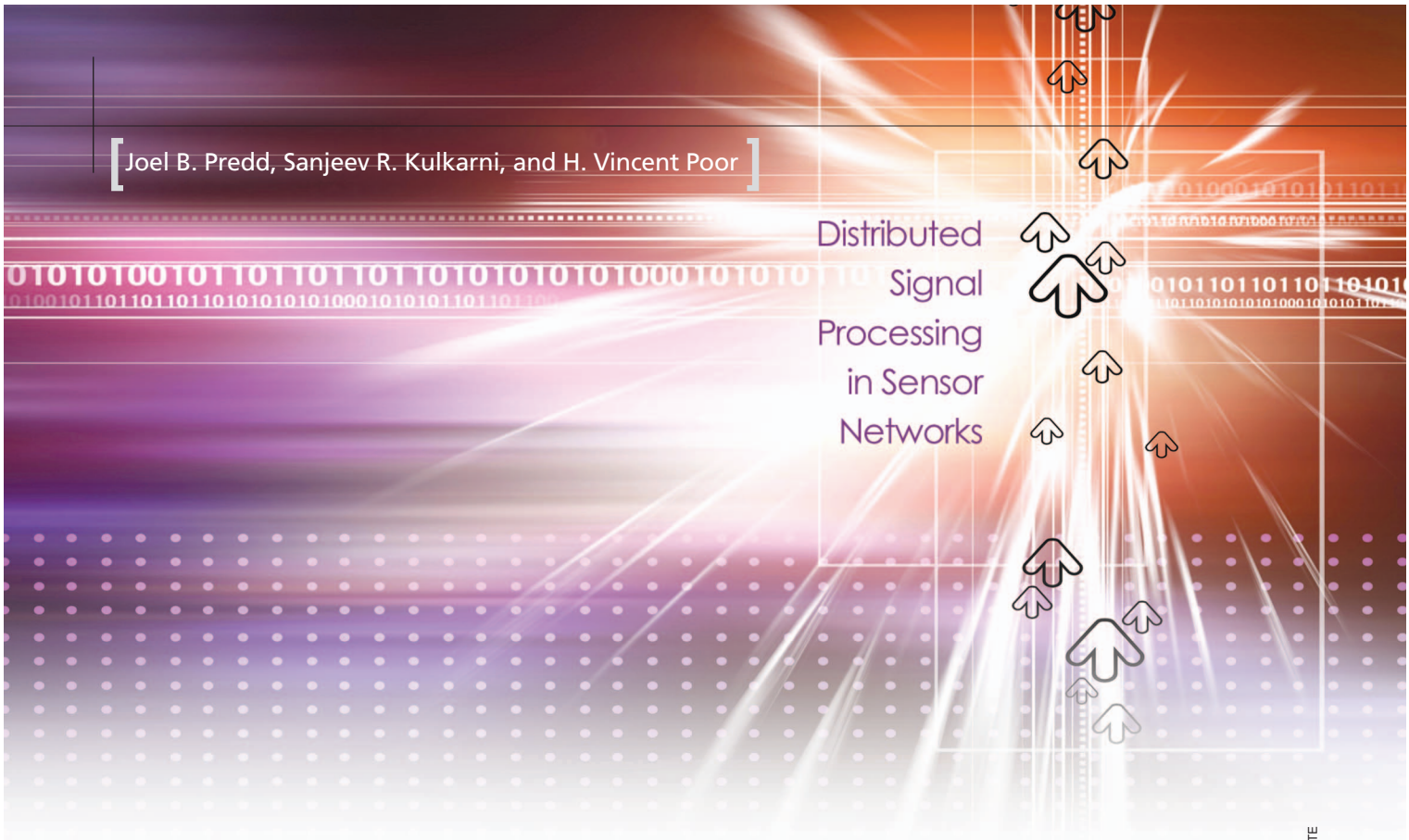
[Joel B. Predd, Sanjeev R. Kulkarni, and H. Vincent Poor]



Distributed Learning in Wireless Sensor Networks

[Applications issues and the problem of distributed inference]

Wireless sensor networks (WSNs) are a fortiori designed to make inferences about the environments that they are sensing, and they are typically characterized by limited communication capabilities due to tight energy and bandwidth constraints. As a result, WSNs have inspired a resurgence in research on decentralized inference. Decentralized detection and estimation have often been considered in the framework of parametric models, in which the statistics of phenomena under observation are assumed known to the system designer. Such assumptions are typically motivated by data or prior application-specific domain knowledge. However, when data is sparse or prior knowledge is vague, robust nonparametric methods are desirable. In this article, nonparametric *distributed learning* is discussed. After reviewing the classical learning model and highlighting the success of machine learning in centralized settings, the challenges that WSNs pose for distributed learning are discussed, and research aimed at addressing these challenges is surveyed.

INTRODUCTION

WSNs have attracted considerable attention in recent years [2]. Research in this area has focused on two separate aspects of such networks: networking issues, such as capacity, delay, and routing strategies;

1053-5888/06/$20.00©2006IEEE

and applications issues. This article is concerned with the second of these aspects of WSNs and in particular with the problem of distributed inference. WSNs are a fortiori designed for the purpose of making inferences about the environments that they are sensing, and they are typically characterized by limited communication capabilities due to tight energy and bandwidth limitations. Thus, decentralized inference is a major issue in the study of such networks.

Decentralized inference has a rich history within the information theory and signal processing communities, especially in the framework of parametric models. Recall that in parametric settings, the statistics of the phenomena under observation are assumed known to the system designer. Under such assumptions, research has typically focused on determining how the capacity of the sensor-to-fusion center channel fundamentally limits the quality of estimates (e.g., rate-distortion tradeoffs [7], [28], [36], [99]), on determining delay-sensitive optimal (under various criteria) sensor decision rules and fusion strategies under unreliable bandwidth constrained channels (e.g., [18], [96], [98]), on characterizing the performance of large networks relative to their centralized communication-unconstrained counterparts (e.g., [16]), or on developing message-passing algorithms through which globally optimal estimates are computed with only local intersensor communications (e.g., [22]). As this diverse yet nonexhaustive list of issues suggests, the literature on decentralized inference is massive and growing. See, for example, [9], [10], [20], [21], [44], [51], [69], [83], [87], [93], and references thereto and therein for entry points.

From a theoretical perspective, parametric models enable a rigorous examination of many fundamental questions for inference under communication constraints. Practically speaking, such strong assumptions should be motivated by data or prior application-specific domain knowledge. If, instead, data is sparse and prior knowledge is limited, robust *nonparametric methods* for decentralized inference are generally preferred.

The anticipated applications for WSNs range broadly from homeland security and surveillance to habitat and environmental monitoring. Indeed, advances in microelectronics and wireless communications have made WSNs the predicted panacea for attacking a host of large-scale decision and information-processing tasks. As the demand for these devices increases, one cannot expect that the necessary data or domain knowledge will always be available to support a parametric approach. Consequently, applications of WSNs provide an especially strong motivation for the study of nonparametric methods for decentralized inference.

Recognizing this demand, a variety of researchers have taken a nonparametric approach to study decentralized detection and estimation. For example, [60] and [97] consider decentralized detection schemes based on the Wilcoxon signed-rank test statistic, and [3] and [35] study the sign detector in a decentralized setting. References [6] and [38] address constant-false-alarm-rate detection in a distributed environment. Schemes for universal decentralized detection and estimation are surveyed in [102] and are studied in detail in [54], [55], [84], [100], and [101]. From a practical perspective, these approaches are attractive not only because they permit the design of robust networks with provable performance guarantees but also because in principle, they support the design of "isotropic" sensors, i.e., sensors that may be deployed for multiple applications without reprogramming.

In this article, our focus is on an alternative nonparametric approach, the learning-theoretic approach [95]. Frequently associated with pattern recognition [23], [24], nonparametric regression [34], and neural networks [4], learning-theoretic methods are aimed precisely at decision problems in which data is sparse and prior knowledge is limited. Researchers in computer science, statistics, electrical engineering, and other communities have been united in the field of machine learning, in which computationally tractable and statistically sound methods for nonparametric inference have been developed. Powerful tools such as boosting [25] and kernel methods [86] have been successfully employed in real-world applications ranging from handwritten digit recognition to function genomics and are quite well-understood statistically and computationally. A general research question arises: Can the power of these tools be tapped for inference in WSNs?

As we will discuss, the classical limits of and algorithms for nonparametric learning are not always applicable in WSNs, in part because the classical models from which they are derived have abstracted away the communication involved in data acquisition. This observation provides inspiration for distributed learning in WSNs and leads to a variety of fundamental questions. How is distributed learning in sensor networks different from centralized learning? In particular, what fundamental limits on learning are imposed by constraints on energy and bandwidth? In light of such limits, can existing learning algorithms be adapted? These questions are representative of a larger thrust within the sensor network community which invites engineers to consider signal processing and communications jointly.

Though the impetus for nonparametric distributed learning has been recognized in a variety of fields, the literature immediately relevant to sensor networks is small and is not united by a single model or paradigm. Indeed, distributed learning is a relatively young area, as compared to (parametric) decentralized detection and estimation, WSNs, and machine learning. Thus, an exhaustive literature review would necessarily focus on numerous disparate papers rather than aggregate results organized by model. In the interest of space, this article decomposes the literature on distributed learning according to two general research themes: distributed learning in WSNs with a fusion center, where the focus is on how learning is effected when communication constraints limit access to training data; and distributed learning in WSNs with in-network processing, where the focus is on how intersensor communications and local processing may be exploited to enable communication-efficient collaborative learning. We discuss these themes within the context of several papers from the field. Though the result is a survey unquestionably biased toward the authors' own interests, our hope is to provide the interested reader with an appreciation for a set of fundamental issues within distributed signal processing and an entree to a growing body of literature.

CLASSICAL LEARNING

In this section, we summarize the supervised learning model that is often studied in learning theory, nonparametric statistics and statistical pattern recognition. Then, we review kernel methods, a popular and well-studied class of algorithms for supervised learning. For a thorough introduction to classical learning models and algorithms, we refer the reader to the review paper [47] and references therein, and standard books [4], [23], [24], [34], [37], [56].

THE SUPERVISED LEARNING MODEL

Let X and Y be $\mathcal{X}$-valued 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, target, or parameter space. Attention in this article is restricted to detection and estimation, i.e., we consider two cases corresponding to binary classification ($\mathcal{Y} = \{0, 1\}$) and regression ($\mathcal{Y} = \mathbb{R}$). To ease exposition, we assume that $\mathcal{X} \subseteq \mathbb{R}^d$.

Given a loss function $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, we seek a decision rule mapping inputs to outputs that achieves minimal expected loss. In particular, we seek a function $g : \mathcal{X} \to \mathcal{Y}$ that minimizes

$$\mathrm{E}\{l(g(X), Y)\}. \tag{1}$$

In the binary classification setting, the criterion of interest is the probability of misclassification, which corresponds to the zero-one loss function $l(y, y') = 1_{\{y \neq y'\}}(y, y')$; in the context of estimation, the squared error $l(y, y') = |y - y'|^2$ is the metric of choice. In parametric settings, one assumes prior knowledge of a joint probability distribution P_{XY} that describes the stochastic relationship between inputs and outputs. Under this assumption, the structure of the loss minimizing decision rule $g^\star : \mathcal{X} \to \mathcal{Y}$ is well understood. In estimation, the regression function $g^\star(x) = \mathrm{E}\{Y | X = x\}$ achieves the minimal expected squared error; the maximum a posteriori (MAP) decision rule is Bayes optimal for binary classification [23]. In the sequel, we will use $L^\star = \mathrm{E}\{l(g^\star(X), Y)\}$ to denote the loss achieved by the loss-minimizing decision rule.

In the learning framework, prior knowledge of the joint distribution P_{XY} is not available, and thus computing the MAP decision rule or the regression function is not possible. Instead, one is provided a collection of *training data* $S_n = \{(\mathrm{x}, y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$, i.e., a set of exemplar input-output pairs. The learning problem is to use this data to infer decision rules with small loss, without making additional assumptions on the data generating probability distribution. In short, $g(X) = g(X, S_n)$ is dependent on the training set but independent of P_{XY}. Often, the training examples in S_n are assumed to be random variables generated from some stochastic process; for example, a standard assumption is that $S_n = \{(X_i, Y_i)\}_{i=1}^n$ is independent and identically distributed (i.i.d.) with $(X_i, Y_i) \sim \mathrm{P}_{XY}$ $\forall i \in \{1, \ldots, n\}$. Generally, such assumptions are introduced to analyze the limits of learning or to characterize the statistical behavior of specific learning rules and are not necessary to define the learning problem in general. To simplify the exposition, we use the notation $S_n = \{(X_i, Y_i)\}_{i=1}^n$ (i.e., with capital letters), which suggests that the data is randomly generated. However, unless otherwise noted, the discussion will be independent of assumptions on the data-generating stochastic process.

KERNEL METHODS

To aid the subsequent discussion, it will be helpful to have basic familiarity with kernel methods, a popular class of algorithms for supervised learning. The kernel approach to learning can be summarized as follows.

First, design a *kernel*, i.e., positive semidefinite function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, as a similarity measure for inputs. For example, one might take K to be the linear kernel $K(\mathrm{x}, \mathrm{x}') = \mathrm{x}^T\mathrm{x}'$, the naive kernel $K(\mathrm{x}, \mathrm{x}') = 1_{\{\|\mathrm{x}-\mathrm{x}'\| \leq r_n\}}(\mathrm{x}, \mathrm{x}')$, or perhaps the Gaussian kernel $K_\sigma(\mathrm{x}, \mathrm{x}') = \exp^{-(1/2\sigma^2)\|\mathrm{x}-\mathrm{x}'\|_2^2}$. Though kernel design is an active area of research, it is generally an art, typically guided by application-specific domain knowledge.

Given such a kernel, construct an estimate $g_n : \mathcal{X} \to \mathbb{R}$ of $\mathrm{E}\{Y | X\}$ as follows:

$$g_n(X) = g_n(X, S_n) = \begin{cases} \frac{\sum_{i=i}^n K(X, X_i) Y_i}{\sum_{i=i}^n K(X, X_i)} & \text{if } \sum_{i=i}^n K(X, X_i) > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

$g_n(X)$ associates with each input X a weighted average of the training data outputs, with the weights determined by how "similar" the corresponding inputs are to X. With the naive kernel, $g_n(X)$ is analogous to the Parzen-window rule for density estimation.

Though naive by the standards of state-of-the-art machine learning, the wisdom behind the kernel approach can verified by the following theorem proved originally by Stone [90] and described in detail in [23] and [34].

THEOREM 1 [90]

Assume that S_n is i.i.d. with $(X_i, Y_i) \sim \mathrm{P}_{XY}$. Define $L_n \triangleq \mathrm{E}\{l(g_n(X), Y) | S_n\}$ with g_n as in (2) with the naive kernel. If $r_n \to 0$ and $nr_n^d \to \infty$, then the kernel rule is universally consistent under the squared-error criterion. That is, with $l(y, y') = |y - y'|^2$, $\mathrm{E}\{L_n\} \to L^\star$ for *all* distributions P_{XY} with $\mathrm{E}\{Y^2\} < \infty$.

In the context of regression under the squared-error criterion, Theorem 1 says that in the limit of large amounts of data, the kernel decision rule will perform as well as could be expected if one had known P_{XY} in advance. Stated in full generality, Stone's Theorem [90] establishes that a large class of "weighted average" learning rules can be made universally consistent, including kernel rules with a Gaussian kernel, nearest neighbor rules, and histogram estimators. Interestingly, under identical assumptions on $\{r_n\}$, the decision rule induced by thresholding $g_n(X)$ at one-half is universally consistent for binary classification under the zero-one loss [23].

Though this seminal result is promising, there is a catch. It is well-known that without additional assumptions on P_{XY}, the

convergence rate of $\mathbf{E}\{L_n\}$ may be arbitrarily slow. Moreover, even with appropriate assumptions, the rate of convergence is typically exponentially slow in d, the dimensionality of the input space. These caveats have inspired the development of practical learning algorithms that recognize the finite-data reality and the so-called curse of dimensionality.

Many popular learning algorithms are based on the principle of (regularized) empirical risk minimization [95], which requires the learning algorithm to minimize a data-dependent approximation of the expected loss (1). For example, reproducing kernel methods constitute one popular approach in which the estimator (classifier) is taken as the solution to the following optimization problem:

$$\min_{f\in\mathcal{H}_K}\left[\frac{1}{n}\sum_{i=1}^{n} l(f(X_i), Y_i) + \lambda\|f\|^2_{\mathcal{H}_K}\right]. \tag{3}$$

The first term in the objective function (3) is the empirical loss of an estimator $f : \mathcal{X} \to \mathbb{R}$ and serves as measurement of how well f fits the data; the second term acts as a complexity control and regularizes the optimization. (In practice, for various statistical and computational reasons, the empirical loss is often measured using a convex loss function which may bound or otherwise approximate the loss criterion of interest. See [86] for a discussion and examples.) $\lambda \in \mathbb{R}_+$ is a constant parameter that governs the tradeoff between these two terms. The optimization variable (i.e., function) is f, which is constrained to be in $\mathcal{H}_K$, the reproducing kernel Hilbert space induced by the kernel $K(\cdot,\cdot)$; $\|\cdot\|_{\mathcal{H}_K}$ is the corresponding norm.

For those unfamiliar with reproducing kernel Hilbert spaces, it is sufficient for the subsequent discussion to note that $\mathcal{H}_K$ is a vector space of functions equipped with a particularly convenient inner-product. If $K(\cdot,\cdot)$ is the linear kernel, for example, then $\mathcal{H}_K$ is the space of linear functions on $\mathcal{X}$, i.e., $\mathcal{H}_K = \{f : \mathcal{X} \to \mathbb{R} : \exists \mathbf{w} \in \mathbb{R}^d \text{ s.t. } f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}\}$. Under minimal regularity [86], the inner-product structure implies that the minimizer $f_\lambda \in \mathcal{H}_K$ to (3) satisfies

$$f_\lambda(\cdot) = \sum_{i=1}^{n} c_{\lambda,i} K(\cdot, X_i), \tag{4}$$

for some $\mathbf{c}_\lambda \in \mathbb{R}^m$. This well-known fact is often termed the "representer theorem" [42]; it is significant because it highlights that while the optimization in (3) is defined over a potentially infinite dimensional Hilbert space, the minimizer must lie in a finite dimensional subspace. It also highlights a sense in which reproducing kernel methods generalize their more naive counterpart, since (2) can be expressed as (4) for a particular choice of $\mathbf{c}_\lambda$. To emphasize the significance of the representer theorem, note that in least-squares estimation it implies that $\mathbf{c}_\lambda$ is the solution to a system of n linear equations. In particular, it satisfies

$$\mathbf{c}_\lambda = (K + \lambda I)^{-1}\mathbf{y}, \tag{5}$$

where $K = (k_{ij})$ is the kernel matrix ($k_{ij} = K(X_i, X_j)$).

Intuitively, when n is large and λ is small, the objective function will closely approximate the expected loss; the hope is that the solution to (3) will then approximately minimize the expected loss. Rigorously, the statistical behavior of reproducing kernel methods is well understood under various assumptions on the stochastic process that generates the examples in S_n [34], [86]. This highly successful technique has been verified empirically in applications ranging from bioinformatics to handwritten digit recognition.

As the reader may well be aware, the scope of algorithms for supervised learning extends far beyond kernel methods and includes, for example, neural networks [4], nearest-neighbor rules [34], decision-trees [80], Bayesian and Markov networks [40], [73], and boosting [25]. Many of these algorithms are well understood computationally and statistically; together they form an indispensable toolbox for nonparametric learning. At this point, we leave classical learning in general and kernel methods in particular, referring the interested reader to previously cited references for additional information.

DISTRIBUTED LEARNING IN WSNS

To illustrate how learning is relevant to decentralized inference and to discuss the challenges that WSNs pose, it will be helpful to have a running example at hand. Suppose that the feature space $\mathcal{X}$ models the set of measurements observable by sensors in a wireless network. For example, the components of an element $\mathbf{x} \in \mathcal{X} = \mathbb{R}^3$ may model coordinates in a (planar) environment, and time. $\mathcal{Y} = \mathbb{R}$ may represent the space of temperature measurements. A fusion center, or the sensors themselves, may wish to know the temperature at some point in space-time; to reflect that these coordinates and the corresponding temperature are unknown prior to the network's deployment, let us model them with the random variable (X, Y). A joint distribution $\mathbf{P}_{XY}$ may model the spatiotemporal correlation structure of a temperature field. If the field's structure is well understood, i.e., if $\mathbf{P}_{XY}$ can be assumed known a priori, then an estimate may be designed within the standard parametric framework. However, if such prior information is unavailable, an alternative approach is necessary.

Suppose instead that sensors are randomly deployed about the environment, and collectively acquire a set of data $\{(X_i, Y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$, which represents the sensors' temperature measurements at various points in space-time. (A host of localization algorithms have been developed to enable sensors to measure their location; see, for example, [29], [63], and [73].) The set $\{(X_i, Y_i)\}_{i=1}^n$ of measurements is akin to the training data described earlier, and the theory and methods described seem naturally applicable. However, the supervised learning model has abstracted away the process of data acquisition and generally does not incorporate communication constraints that may limit a learner's access to the data. Indeed, the theory and methods discussed depend critically on the assumption that the training data is entirely available to a single processor.

In WSNs, of course, the energy and bandwidth required to pool the sensors' measurements may be prohibitively large.

Thus, employing centralized learning strategies may limit the sensors' battery life and may ultimately preclude one from realizing the potential of WSNs. Are there more communication-efficient methods for *distributed learning*? In particular, can we design learning algorithms that respect constraints on energy and bandwidth?

Before proceeding, note that the simplicity of the running example should not mask the generality of the model. Indeed, $\mathcal{X}$ may model more than position and may represent a space of multimodal sensor measurements that commonly occur in WSN applications. Moreover, $\mathcal{Y}$ may model any number of quantities of interest, for example the strength of a signal emitted from a target, a force measured by a strain gauge, or an intensity level assessed by an acoustic sensor. In general, each sensor or the fusion center seeks a decision rule $f : \mathcal{X} \to \mathcal{Y}$ that predicts output measurements using input measurements. And they wish to do so using only the data observed by the sensor network.

A GENERAL MODEL FOR DISTRIBUTED LEARNING

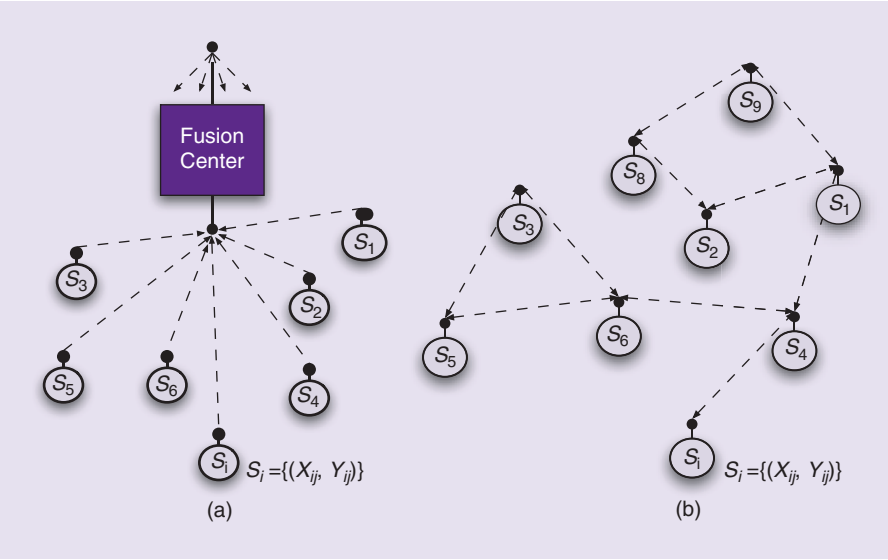
Now let us pose a general model for distributed learning that will aid in formulating the problem and categorizing work within the field. Suppose that in a network of m sensors, sensor i has acquired a set of measurements, i.e., training data, $S_i \subset \mathcal{X} \times \mathcal{Y}$. In the running example above, S_i may represent a stationary sensor's measurements of temperature over the course of a day or a mobile sensor's readings at various points in space-time. Suppose further that the sensors form a wireless network, whose topology is specified by a graph. For example, consider the models depicted pictorially in Figure 1.

Each node in the graph represents a sensor and its locally observed data; an edge in the graph posits the existence of a wireless link between sensors. Note that the fusion center can be modeled as an additional node in the graph, perhaps with larger capacity links between itself and the sensors, to reflect its larger energy supply and computing power. A priori, this model makes no assumptions on the topology of the network (e.g., the graph is not necessarily connected); however, the success of distributed learning may in fact depend on such properties.

Much of the work in distributed learning differs in the way that the capacity of the links is modeled. Given the complexity of nonparametric learning, simple application-layer abstractions are typically preferred over detailed physical layer models. As we shall see, often the links are assumed to support the exchange of "simple" real-valued messages, where simplicity is defined relative to the application (e.g., sensors communicate summary statistics rather than entire data sets). Note that this general model is consistent with a pervasive intuition within the WSN community, which views WSNs as distributed sampling devices with a wireless interface.

Generally speaking, research in distributed learning can be categorized within two classes of networks. Depicted in Figure 1(a), the parallel network supposes a network of sensors that communicate directly with a fusion center. In this setting, the question is: How is the fusion center's ability to learn fundamentally effected when communication constraints limit its access to the data? This architecture is relevant to WSNs whose primary purpose is data collection. A second class of networks may be modeled with an ad hoc structure such as the network in Figure 1(b). The typical assumption is that the topology of these networks is dynamic and perhaps unknown prior to deployment; a fusion center may exist, but the sensors are largely autonomous and may make decisions independently of the fusion center (e.g., to track a moving object). Here, we might ask how can local intersensor communication and in-network processing be exploited to enable communication-efficient collaborative learning?

In the next few sections, we review recent work aimed at addressing these questions in the context of WSNs. Though statistical and machine learning are rife with results relevant to distributed learning in general, to our knowledge suprisingly little research has addressed learning in WSNs in particular. Thus, before proceeding, let us highlight several areas of machine learning research that are relevant to distributed learning, if not WSNs, and that may bear on future studies in distributed signal processing.



[FIG1] (a) A parallel network with a fusion center. (b) An ad hoc network with in-network processing.

RELATED WORK

Within the context of WSNs, [64] developed a nonparametric kernel-based methodology for designing decentralized detection systems. As in centralized learning, a training set was assumed available offline to a single processor. The data was used to find a decision rule that solved an

optimization problem similar to (3), with the additional constraint that the rule lie within a restricted class of estimators that were deployable across a sensor network; the powerful notion of a marginal kernel was 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.

THE ANTICIPATED APPLICATIONS FOR WIRELESS SENSOR NETWORKS RANGE BROADLY FROM HOMELAND SECURITY AND SURVEILLANCE TO HABITAT AND ENVIRONMENTAL MONITORING.

Ensemble methods have attracted considerable attention within machine learning. Examples of these techniques include bagging, boosting, mixtures of experts, and others [12], [25], [26], [39], [43]. Intuitively, these methods allocate portions of the training database to different classifiers that are independently trained. The individual classifiers are subsequently aggregated, sometimes through a different training algorithm and sometimes using feedback from the training phase of the individual classifiers. One might cast these algorithms within the framework of distributed learning in a parallel network, but ensemble methods are generally designed within the classical model for supervised learning and fundamentally assume that the training set S_n is available to a single coordinating processor. In general, the focus of ensemble learning is on the statistical and algorithmic advantages of learning with an ensemble and not on the nature of learning under communication constraints. Nevertheless, many fundamental insights into learning have arisen from ensemble methods; future research in distributed signal processing stands to benefit.

Inspired by the availability of increasingly large data sets, an active area of machine learning research focuses on "scaling up" existing learning algorithms to handle massive training databases; see, for example, [11], [17], [31], and [79] and references thereto and therein. One approach is to decompose the training set into smaller "chunks" and subsequently parallelize the learning process by assigning distinct processors/agents to each of the chunks. In this setting, sometimes termed parallel learning, the communication constraints arise as parameters to be tweaked, rather than from resources to be conserved; this difference in perspective often limits the applicability of the underlying communication model to applications like sensor networks. However, in principle, algorithms for parallelizing learning may be useful for distributed learning and vice-versa.

Population learning is an early model for distributed learning [41], [59], [105]. In that setting, a parallel network is considered; it is assumed that the "sensors" locally train estimators before transmitting a complete description of their learned rules to the fusion center. The fusion center's task is to observe the response of the network to infer a more accurate rule. The original model [41] was parametric (i.e., "distribution specific" learning) and was constructed in the spirit of the "probably approximately correct" (PAC) framework [94]. Generalizations such as [59] relaxed such assumptions, but the results ultimately depend on strong assumptions about a class of hypotheses that generate the data. The utility of these results to WSNs may be limited by these strong assumptions or by the fact that sensors must communicate a complete description of the rule. Nevertheless, population learning appears to have motivated a host of other studies in distributed learning, and may provide insights for distributed signal processing.

The online learning framework also appears relevant to distributed learning in WSNs with a fusion center [14], [26], [52]. In that setting, a panel of experts (i.e., a network of sensors) provides predictions (one can imagine that predictions arose from independently trained estimators, but such assumptions are unnecessary). A central agent (i.e., a fusion center) receives these forecasts and bases its own prediction on a weighted average of the experts' predictions. Upon learning the "truth" (i.e., Y), the agent suffers a loss (e.g., squared error). In repeated trials, the agent updates the weights of its weighted average by taking into account the performance of each expert. Under minimal assumptions on the evolution of these trials, bounds are derived that compare the trial-averaged performance of the central agent with that of the best (weighted combination of) expert(s). This framework may be relevant to aggregation problems that arise in WSNs, however to our knowledge such applications have not been made.

Finally, the field of data mining has explored distributed learning in the context of distributed databases. Here, various agents have access to training databases and wish to collaborate with each other to maximize the accuracy of their decision rules. Consider, for example, the fraud detection application, where corporations have access to large databases of consumer transactions that they wish to use to identify fraudulent interactions. In this setting, communication constraints between agents arise due to security, privacy, or legal concerns, not from limitations on energy or bandwidth. Nonetheless, the problem bears a striking resemblance to the ad hoc structure of distributed learning in sensor networks. In the data mining context, a distributed boosting algorithm is studied in reference [50]; a similar algorithm is analyzed in the framework of secure multiparty computation in [27].

DISTRIBUTED LEARNING IN WSNS WITH A FUSION CENTER

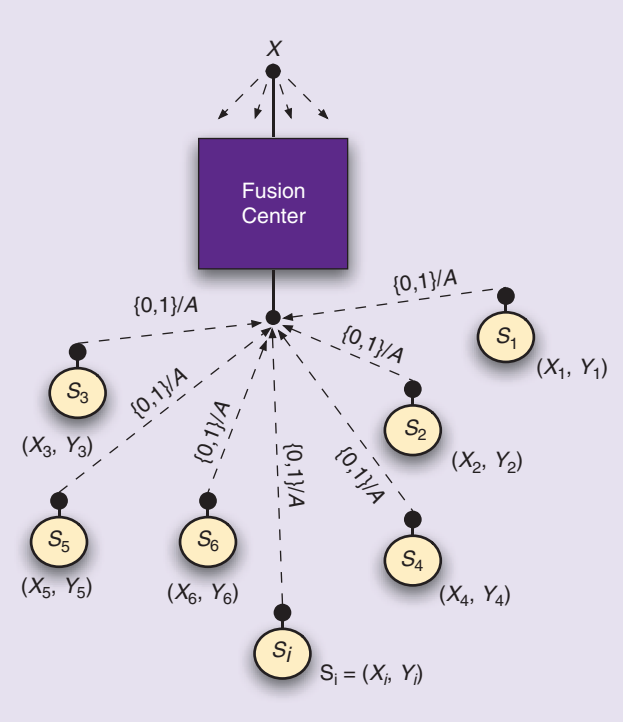
In this section, we discuss distributed learning in WSNs with a fusion center, which focuses on the parallel network depicted in Figure 1(a). Recall that in this setting, each sensor in the network acquires a set of data. In the running example, the data may constitute the sensors' temperature measurements at discrete points in space-time. The fusion center would like to use the locally observed data to construct a global estimate of the continuously varying temperature field.

A CLUSTERED APPROACH

The naive approach in this setting would require the sensors to send all of their data to the fusion center. As has been discussed, this approach would be costly in terms of energy and bandwidth. A more principled methodology might designate a small subset of nodes to send data. If the number of nodes is small, and the data (or the nodes) are wisely chosen, then such a strategy may be effective in optimizing learning performance while keeping communication costs to a minimum.

For example, one may partition the sensors into subgroups, and assign each a "cluster head." (Distributed clustering algorithms have been developed with such applications in mind; see [5], for example.) Cluster heads may retrieve the data from sensors within its group; since the sensors within a cluster are nearby, this exchange may be inexpensive since communication occur over short distances can be done efficiently. Then, the cluster head may filter this data and send the fusion center a summary, which might include a locally learned rule or data that is particularly informative (e.g., support vectors). Clustered approaches has been considered frequently within parametric frameworks for detection and estimation [20].

Reference [63] considered a clustered approach to address sensor network localization. There, the feature space $\mathcal{X} = \mathbb{R}^2$ models points in a planar terrain, and the output space $\mathcal{Y} = \{0, 1\}$ models whether or not a point belonged to (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., nonbase 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. We highlight this as an example of the clustered approach to distributed learning with a fusion center, a methodology which is broadly applicable.



[FIG2] The model studied in [75] and [77].

STATISTICAL LIMITS OF DISTRIBUTED LEARNING

Stone's theorem is a seminal result in statistical pattern recognition which established the existence of universally consistent learning rules (see Theorem 1). Many efforts have extended this result to address the consistency of Stone-type learning rules under various sampling processes; for example, [23], [34] and references therein, [19], [32], [45], [48], [49], [58], [65], [66], [67], [85], [90], [103], and [104]. These results extend Theorem 1 by considering various dependency structures within the training data (e.g., Markovian data). However, all of these works are in the centralized setting and assume that the training database is available to a single processor.

Reference [77] attempted to characterize the limits of distributed learning with a fusion center, by overlaying several simple communication models onto the classical model for supervised learning. In particular, [77] sought to extend Stone's theorem by addressing the following question: with sensors that have each acquired a small set of training data and that have some limited ability to communicate with the fusion center, can enough information be exchanged to enable universally consistent learning?

To address this question, [77] supposes that each sensor acquires just one training example, i.e., $S_i = \{(X_i.Y_i)\}$. Communication was modeled as follows: when the fusion center observes a new observation $X \sim \mathrm{P}_X$, it broadcasts the observation to the network in a request for information. At this time, bandwidth constraints limit each sensor to responding with at most 1 b. That is, each sensor chooses whether or not to respond to the fusion center's request for information; if it chooses to respond, a sensor sends either a 1 or a 0 based on its local decision algorithm. Upon observing the response of the network, the fusion center combines the information to create an estimate of Y.

A refined depiction of the architecture of this model is depicted in Figure 2. To emphasize its structure, note that the fusion center has a broadcast channel back to the sensor (for requesting information on X), and each sensor has a point-to-point wireless uplink channel over which they can send 1 b. Since each sensor may *abstain* from voting altogether, the sensors' uplink channels have a slightly larger capacity than is suggested by this mere 1 b that we have allowed them to physically transmit to the fusion center. Indeed, sensor-to-fusion center communication occurs even when a sensor abstains from voting.

Despite the simplicity of the model, fundamental questions arise. In particular, can the sensors communicate enough information to the fusion center to enable universally consistent learning? Proved in [77], the following theorem settles the question in this model for distributed learning with abstention.

THEOREM 2 [77] (CLASSIFICATION AND ESTIMATION WITH ABSTENTION)

Suppose that

1) the sensors' data $\cup_{i=1}^{m} S_i$ are i.i.d. with $(X_i, Y_i) \sim \mathbf{P}_{XY} \forall i \in \{1, \ldots, m\}$

2) each sensor knows m, the size of the sensor network.

Then, in binary classification under the zero-one loss, and in estimation under the squared-error criterion, there exist sensor decision rules and a fusion rule that enable universally consistent distributed learning with abstention.

In this model, each sensor decision rule can be viewed as a selection of one of *three* states: abstain, vote and send 0, and vote and send 1. With this observation, Theorem 2 can be interpreted as follows: $\log_2(3)$ bits per sensor per decision is sufficient to enable universally consistent learning in this model for distributed learning with abstention. In this view, it is natural to ask whether these $\log_2(3)$ bits are necessary. That is, can consistency be achieved by communicating at lower bit rates?

THIS ARTICLE SURVEYS THE PROBLEM OF DISTRIBUTED LEARNING IN WIRELESS SENSOR NETWORKS.

To answer this question, [77] considered a revised model, precisely the same as above, except that in response to the fusion center's request for information, each sensor must respond with 1 or 0; abstention is not an option and thus, each sensor responds with exactly 1 b per decision. Can the sensors communicate enough information to the fusion center to enable universally consistent distributed learning *without abstention*? Also proved in [77], the following theorems settle this question.

THEOREM 3 [77] (CLASSIFICATION WITHOUT ABSTENTION)

Suppose that

1) the sensors' data $\cup_{i=1}^{m} S_i$ are i.i.d. with $(X_i, Y_i) \sim \mathbf{P}_{XY} \forall i \in \{1, \ldots, m\}$

2) each sensor knows m, the size of the sensor network.

Then, in binary classification under the zero-one loss, there exist sensor decision rules and a fusion rule that enable universally consistent distributed learning without abstention.

THEOREM 4 [77] (ESTIMATION WITHOUT ABSTENTION)

Suppose that

1) the sensor's data $\cup_{i=1}^{m} S_i$ is i.i.d. with $(X_i, Y_i) \sim \mathbf{P}_{XY} \forall i \in \{1, \ldots, m\}$

2) each sensor knows m

3) the fusion rule satisfies a set of regularity conditions. (Reference [77] assumes that the fusion rule is invariant to the order of bits received from the sensor network and Lipschitz continuous in the average Hamming distance.)

Then, for any sensor decision rule that obeys the constraints of distributed learning without abstention, there does *not* exist a regular fusion rule that is consistent for *every* distribution $\mathbf{P}_{XY}$ with $\mathbf{E}\{Y^2\} < \infty$ under the squared-error criterion.

In [77], Theorems 2 and 3 are proved by construction; sensor decision rules and fusion rules are specified that simultaneously satisfy the communication constraints of the respective models and are provably universally consistent. Theorem 4 is proved via a counterexample and thereby establishes the impossibility of universal consistency in distributed regression without abstention for a restricted, but reasonable, class of WSNs.

Theorems 2–4 establish fundamental limits for distributed learning in WSNs by addressing the issue of whether or not the guarantees provided by Stone's theorem in centralized environments hold in distributed settings. However, the applicability of these results may be limited by the appropriateness of the model. For example, in practice, the training data observed by a sensor network may not be i.i.d; in the field estimation problem, data may be corrupted by correlated noise [89], [91], [92]. Moreover, the process by which sensors acquire data may differ from the process observed by the fusion center; for example, sensors may be deployed uniformly about a city, despite the fusion center's interest in a particular district. In the context of binary classification, [75] established the achievability of universally consistent distributed learning with abstention under a class of sampling processes which model such an asymmetry. In general, extending the above results to realistic sampling processes is of practical importance.

In these models, the assumption that each sensor acquires only one training example appears restrictive. However, the results hold for training sets of any finite (and fixed) size. Thus, these results have examined an asymptote not often consider in machine learning, corresponding to the limit of the number of learning agents. One can argue that if the number of examples per sensor grows large, then universally consistent learning is possible within most reasonable communication models. Thus, communication-constrained sensor networks with finite training sets is an interesting case.

Finally, note that these models generalize, in a sense, models recently considered in universal decentralized detection and estimation [54], [55], [84], [100], [101]. The communication and network models in that setting are nearly identical to those considered here. However, there the fusion center is interested in making a binary decision or in estimating a real-valued parameter, whereas in the present setting, the fusion center estimates a function.

DISTRIBUTED LEARNING IN AD HOC WSNS WITH IN-NETWORK PROCESSING

In this section, we turn our attention to distributed learning in WSNs with in-network processing, considering networks with the ad hoc structure depicted in Figure 1(b). One should note that in doing so, we do not exclude the possibility of there being a fusion center. Our shift represents a change in focus. We consider how in-network processing and local intersensor communication may enable communication-efficient collaborative learning.

Much of the work in distributed learning differs in the way that the capacity of the links is modeled. Given the complexity of

nonparametric learning, simple application-layer abstractions are typically preferred over detailed physical layer models. The links are typically assumed to support the exchange of "simple" real-valued messages, where simplicity is defined relative to the application (e.g., sensors communicate 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. The key intuition, which generally requires a formal justification, is that local communication is more efficient since it requires less energy and bandwidth than communicating globally.

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 [1], [40], [46], [53], [70], [73], [74]. These tools are often applicable to signal processing in WSNs, 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 intersensor communications are envisioned to occur over similar graphical structures. Indeed, graphical models form a broad topic in their own right, and applications to sensor networks are deserving of a separate article (e.g., [15]). Here, our focus is specifically on how message-passing algorithms, broadly construed, may be applied to address distributed learning in WSNs. The learning formalism aside, various connections may exist between the work we now discuss and the previously cited studies.

MESSAGE-PASSING ALGORITHMS FOR LEAST-SQUARES REGRESSION

To simplify the exposition, let us restrict ourselves to a least-squares estimation problem and consider the reproducing kernel estimator discussed earlier. Also to simplify our discussion, assume that each sensor measures a single training example, i.e., $S_i = (X_i, Y_i)$. Finally, assume that each sensor has been preprogrammed with the same kernel K.

Recall, reproducing kernel methods take as input a training set $\cup_{i=1}^{m} S_i = S = \{(X_i, Y_i)\}_{i=1}^{m}$ and in the least-squares regression setting output a function $f_\lambda : \mathcal{X} \to \mathcal{Y}$ which solves the optimization problem

$$\min_{f \in \mathcal{H}_K} \left[\frac{1}{m} \sum_{i=1}^{m} (f(X_i) - Y_i)^2 + \lambda \| f \|_{\mathcal{H}_K}^2 \right]. \tag{6}$$

As discussed earlier, solving (6) is infeasible in WSNs, since the data in S is distributed about the network of sensors.

Through (6) (and other implementations of the principle of empirical risk minimiziation), learning has been reduced to solving an optimization problem. Thus, distributed and parallel optimization, fields with rich histories in their own right [8], [13], have an immediate bearing on problems of distributed learning. Indeed, tools from distributed and parallel optimization have recently been considered in the context of WSNs [22], [57], [76], [78], [81], [82], [88]. Here, we discuss three approaches that differ by the structure that they exploit and by the messages that sensors exchange.

TRAINING DISTRIBUTIVELY BY EXPLOITING SPARSITY

The first method that we consider exploits an intuition that is upheld in many sensor network applications: in WSNs, nearby sensors can communicate efficiently and are expected to have correlated measurements. In the running example, for example, the temperature field may be slowly varying and thus it may be reasonable to assume that nearby sensors have similar temperature measurements. This intuition implicitly assumes a relationship between topology of the wireless network and the correlation structure of the field; note that the network topology arises from the same notion of "nearby." Many algorithms for distributed estimation using graphical models rely on formalizations of this powerful intuition, e.g., [22].

To illustrate how such structure may be useful for solving (6), recall from (5) that in least-squares kernel estimation, the solution to (6) is implied by the solution to the following system of linear equations

$$(K + \lambda I)\mathbf{c}_\lambda = \mathbf{y}, \tag{7}$$

where $K = (K_{ij})$ is the kernel matrix with $K_{ij} = K(X_i, X_j)$. If each sensor acquires a single training datum, then K represents a matrix of sensor-to-sensor similarity measurements. For many kernels, K is sparse. Various algorithms are available for efficiently solving sparse systems of linear equations, some of which permit message-passing implementations [30], [70]. When the sparsity "corresponds" with the topology of the network—as the intuition suggests—often these messages are passed between neighboring nodes; in that event, the algorithms imply a distributed, often energy-efficient, training algorithm.

Along these lines, [33] developed a distributed algorithm based on a distributed Gaussian elimination algorithm executed on a cleverly engineered junction tree. A detailed description of this algorithm requires familiarity with the junction tree formalism and knowledge of a distributed Gaussian elimination algorithm, which unfortunately are outside the scope of the present article. Notably, the algorithm has provable finite-time convergence guarantees and arrives at the globally optimal solution to (6). Because their system in [33] is developed within a very general framework for distributed inference in sensor networks [71], this approach is applicable in many cases when the intuition we have described fails (e.g., when sparsity is prevalent but may not "correspond" in an intuitive way the network topology). Nevertheless, the approach appears maximally efficient from an energy and bandwidth perspective when the intuition bears credibility. We refer the reader to [33] for additional detail and a description of several interesting experiments.

TRAINING DISTRIBUTIVELY USING INCREMENTAL SUBGRADIENT METHODS

A second approach to distributively solving (6) exploits the fact that the sensor measurements are decoupled in the additive

objective function. For reasons that will soon become clear, let us rewrite (6) as

$$\min_{f\in\mathcal{H}_K}\left[\frac{1}{m}\sum_{i=1}^{m}(f(X_i)-Y_i)^2+\sum_{i=1}^{m}\lambda_i\|f\|^2_{\mathcal{H}_K}\right]. \tag{8}$$

When $\sum_{i=1}^{m}\lambda_i=\lambda$, (8) is clearly equivalent to (6).

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 (6) defines a sequence of estimates

$$\hat{f}^{(k+1)}=\hat{f}^{(k)}-\alpha_k\frac{\partial F}{\partial f}\left(\hat{f}^{(k)}\right), \tag{9}$$

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

$$j=k \mod m \tag{10}$$

$$\hat{f}^{(k+1)}=\hat{f}^{(k)}-\alpha_k\frac{\partial G_j}{\partial f}\left(\hat{f}^{(k)}\right), \tag{11}$$

where $G_j=(f(X_j)-Y_j)^2+\lambda_j\|f\|^2_{\mathcal{H}_K}$. In short, the update equations iterate over the m terms in F. Incremental subgradient algorithms have been studied in detail in [61] and [62]. Under reasonable regularity (e.g., $\|\partial G_j/\partial f\|$ must be bounded), one can show that if $\alpha_k\to 0$, then $\|\hat{f}^{(k+1)}-f_\lambda\|_{\mathcal{H}_K}\to 0$; with a constant step size (i.e., $\alpha_k=\alpha$), one can bound the number of iterations required to make $\|\hat{f}^{(k)}-f_\lambda\|_{\mathcal{H}_K}\le\epsilon$.

These facts were exploited in [81] and [82] to derive a message-passing algorithm for distributed parameter estimation. In particular, they note that the update equation at iteration k depends only on the data observed by sensor $k \mod m$. With this insight, they propose a two-step process to distributed estimation. First, establish a path through the network 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 depends 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 $\|f-f_\lambda\|_{\mathcal{H}_K}\le\epsilon$. The algorithm is depicted pictorially in Figure 3.

Notably, the present setting is slightly different than the one originally conceived in [81] and [82]. First, [81] and [82] consider more general nonquadratic objective functions. Second, there the optimization variable was a real valued (i.e., real-valued parameter estimation); here we estimate a function. From a theoretical perspective, the differences are primarily technical. However, practically speaking the second difference is important. In particular, one can show that the functional derivative is given by

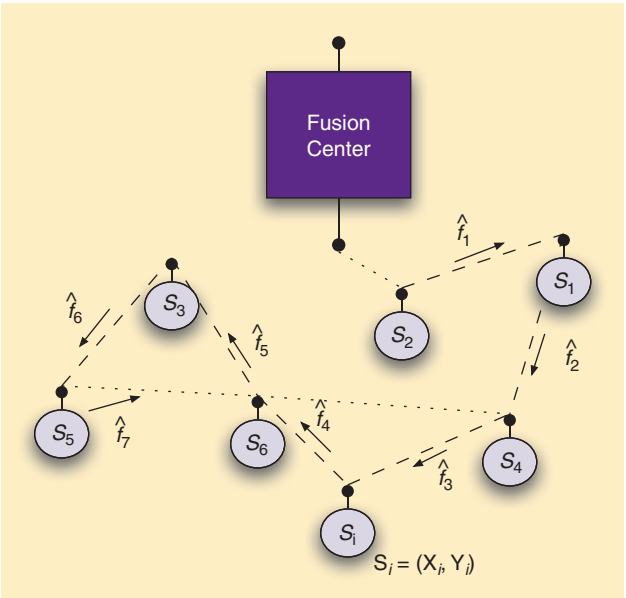
$$\frac{\partial G_j}{\partial f}=\frac{2}{m}(f(X_j)-Y_j)K(\cdot,X_j)+2\lambda_j f(\cdot). \tag{12}$$

In consequence, all the data will ultimately propagate to all the sensors, since exchanging (X_j, Y_j) is necessary to compute $\partial G_j/\partial f$ and hence to share $\hat{f}^{(k)}$ (assuming that the sensors are preprogrammed with the kernel). This is precisely what we were trying to avoid in the first place. Thus, in the general case, the incremental subgradient approach may have limited use for reproducing kernel methods. However, often $\mathcal{H}_\kappa$ admits a lower dimensional parameterization; for example, this is the case for the linear kernel when $\mathcal{H}_\kappa$ is the space of linear functions on $\mathcal{X}=\mathbb{R}^d$. In that case, messages may be communicated more efficiently to the tune of considerable energy savings. The energy-accuracy tradeoff is discussed in [81].

Note that unlike the sparsity-driven approach, the algorithm is independent of modeling assumptions that link the kernel to the topology of the network. Indeed, the distributed training algorithms depend only on there being a path through the network; the kernel and the network are distinct objects. Finally, note that [88] addressed a generalization of the incremental subgradient message-passing methodology by considering a clustered network topology.

TRAINING DISTRIBUTIVELY USING ALTERNATING PROJECTION ALGORITHMS

A final approach to solving (6) distributively relies on sensors to locally (and iteratively) share data, not entire functions, and thereby addresses the practical weakness that sometimes limits



[FIG3] An incremental subgradient approach to training distributively [81], [82].

[TABLE 1] TRAINING DISTRIBUTIVELY WITH ALTERNATING PROJECTIONS.

INITIALIZATION: NEIGHBORING SENSORS SHARE TRAINING DATA *INPUTS*:
SENSOR s STORES $\{X_j\}_{j\in N_s}$
SENSOR s INITIALIZES $z_s = y_s$, $f_{s,0} = 0 \in \mathcal{H}_K$

TRAIN: FOR $t = 1, \ldots, T$
FOR $s = 1, \ldots, m$
SENSOR s
QUERIES z_j $\forall j \in N_s$
$f_{s,t} := \arg\min_{f\in\mathcal{H}_K}$
$\left[\sum_{j\in N_s}(f(X_j) - z_j)^2 + \lambda_s \|f - f_{s,t-1}\|^2_{\mathcal{H}_K}\right]$
UPDATES $z_j \leftarrow f_{s,t}(X_j)$ $\forall j \in N_s$
END
END

the incremental subgradient approach. To construct the algorithm, assume that sensor i can query its neighbors' data (X_j, Y_j) for all $j \in N_i$ (where $N_i \subseteq \{1, \ldots, m\}$ denotes the neighbors of sensor i), and may use this local data to compute a global estimate for the field by solving

$$\min_{f\in\mathcal{H}_K}\left[\sum_{j\in N_i}(f(\mathrm{x}_j) - y_j)^2 + \lambda_i \|f\|^2_{\mathcal{H}_K}\right]. \tag{13}$$

Presumably, each sensor can compute such an estimate; thus, in principle, one could iterate through the network allowing each sensor to compute a global estimate using only local data. The key idea behind an algorithm presented in [76] and [78] is to couple this iterative process using a set of message variables. Specifically, sensor i maintains an auxiliary message variable $z_i \in \mathbb{R}$, which is interpreted as an estimate of the field at X_i. Each sensor initializes its message variable according to its initial field measurement, i.e., $z_i = Y_i$ to start.

Subsequently, the sensors perform a local computation in sequential order. At its turn, sensor i *queries* its neighbors' message variables and computes $f_i \in \mathcal{H}_K$ as the solution to (13) using $\{(X_j, z_j)\}_{j\in N_i}$ as training data. Then, sensor i *updates* its neighbors' message variables, setting $z_j = f_i(X_j)$ for all $j \in N_i$. Since sensor i's neighbors may pass along their newly updated message variables to other sensors, the algorithm allows local information to propagate globally.

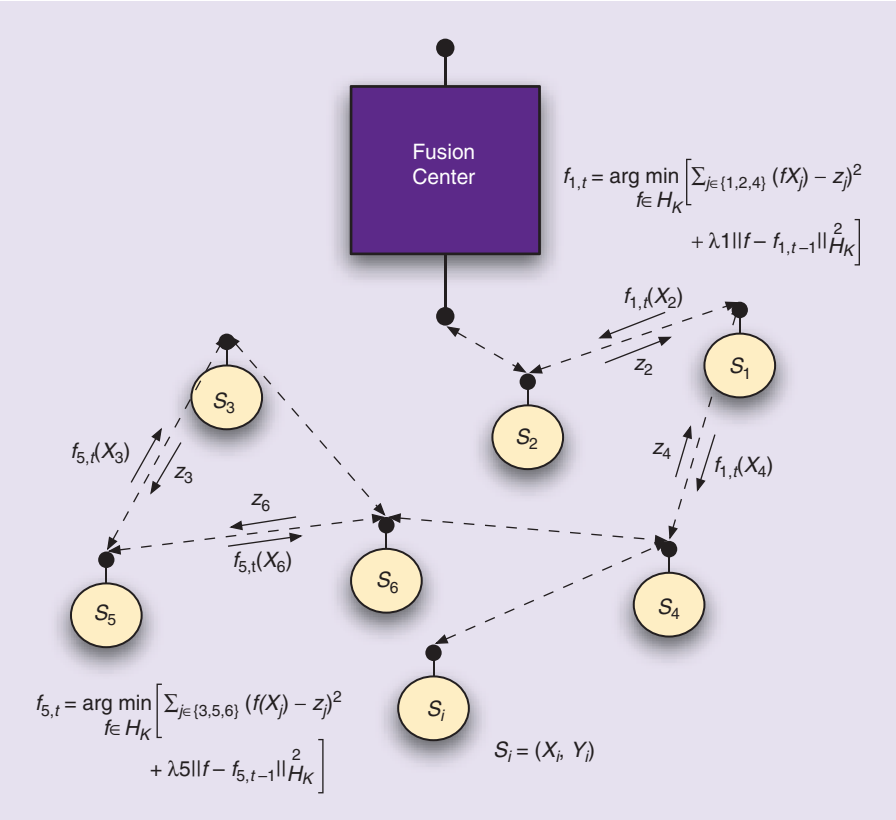
Two additional modifications are needed to fully specify the algorithm. First, multiple passes (in fact, T iterations) through the network are made; for convenience, denote sensor i's global estimate at iteration t by $f_{i,t} \in \mathcal{H}_K$. Secondly, each sensor controls the "intertia" of the algorithm, by modifying the complexity term in (13). Specifically, at iteration t, $f_{i,t} \in \mathcal{H}_K$ is found to minimize

$$\min_{f\in\mathcal{H}_K}\left[\sum_{j\in N_i}(f(X_j) - z_j)^2 + \lambda_i \|f - f_{i,t-1}\|^2_{\mathcal{H}_K}\right]. \tag{14}$$

The resulting algorithm is summarized more concisely in Table I and depicted pictorially in Figure 4.

Here, the algorithm has been derived through an intuitive argument. However, [76], [78] introduces this approach as an application of powerful successive orthogonal projection (SOP) algorithms [13] applied to a geometric topology-dependent relaxation of the centralized kernel estimator (14). Using standard analysis of SOP algorithms, [76] proves that the algorithm converges in the limit of the number of passes through the network (i.e., as $T \to \infty$) and characterizes the point of convergence as an *approximation* to the globally optimal solution to the centralized problem (6). In [78], the performance of this algorithm is discussed in a simplified statistical setting. For additional detail on this general approach, we refer the interested reader to the full paper [76], [78].

A few comments are in order. First, note that as was the case for the incremental subgradient approach, this algorithm is independent of assumptions that couple the kernel matrix K with the network topology. Thus, prior domain



[FIG4] Training distributively with alternating projections [76], [78].

knowledge about $\mathbf{P}_{XY}$ can be encoded in the kernel; the training algorithm approximates the centralized estimator as well as the communication constraints allow. Second, in contrast to the previous approach, sensors share data, i.e., real-valued evaluations of functions, and not the functions themselves. This significantly broadens the scope of problems where the approach is applicable.

DECENTRALIZED INFERENCE HAS A RICH HISTORY WITHIN THE INFORMATION THEORY AND SIGNAL PROCESSING COMMUNITIES, ESPECIALLY IN THE FRAMEWORK OF PARAMETRIC MODELS.

Notably, just as in the incremental approach, each sensor derives a global estimate, despite having access to only local data; this is useful when the sensors are autonomous (e.g., mobile) and may make predictions on their own independent of a fusion center. Next, as demonstrated in [76] and [78], sensor i can compute $f_{i,t} \in \mathcal{H}_K$ in a manner similar to (5); the calculation requires solving an $|N_i|$-dimensional system of linear equations. As stated in Table 1, the algorithm assumes that the sensors perform their local computations in sequence. As discussed in the full paper, the computations can be parallelized, insofar as none of the message variables is updated by multiple sensors simultaneously. Finally, experiments in [76] demonstrate that the algorithm converges quickly in practice; this is promising since for energy efficiency, the number of iterations (i.e., T) must be bounded. Additional experiments suggest that this approach to passing data considerably enhances the accuracy of the individual sensors' estimates.

OTHER WORK

Many other learning algorithms implicitly solve (or approximately solve) an optimization problem similar to (6), perhaps with a different loss function and perhaps over a different class of functions. Thus, though the discussion has focused exclusively on least-squares kernel regression, the key ideas are more broadly applicable, increasing their relevance to distributed learning in sensor networks.

In the context of boundary estimation in WSNs, [68] derived a hierarchical processing strategy by 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 (6)] and enables an interesting energy-accuracy analysis. Reference [69] derives a distributed 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.

CONCLUSIONS

This article has surveyed the problem of distributed learning in WSNs. Motivated by the anticipated breadth of applications of WSNs, we first discussed how parametric methods for distributed signal processing may be inappropriate in those applications where data is sparse and prior knowledge is limited. Then, inspired by the success of machine learning in classical, centralized signal processing applications, we sought to understand whether and how the power of existing learning models and algorithms could be leveraged for nonparametric distributed signal processing in wireless sensors networks. After identifying the challenges that bandwidth and energy constraints impose on learning and posing a general model distributed learning, we considered two general themes of existing and future research: distributed learning in networks with a fusion center, and distributed learning in ad hoc networks with in-network processing. Subsequently, we discussed recent research within these themes. In doing so, we hope that this article has usefully described a set of fundamental issues for nonparametric distributed signal processing and provided an entry point to a larger body of literature.

ACKNOWLEDGEMENTS

This research was supported in part by the Army Research Office under Grant DAAD19-00-1-0466, in part by Draper Laboratory under IR&D 6002 Grant DL-H-546263, in part by the National Science Foundation under Grants CCR-02055214 and CCR-0312413, and in part by the U.S. Army Pantheon Project.

AUTHORS

Joel B. Predd (jpredd@princeton.edu) received a B.S. from Purdue University in 2001 and an M.A. from Princeton University in 2003, both in electrical engineering. Currently, he is a Ph.D. candidate in the information sciences and systems group at Princeton University. He spent the summer of 2004 visiting the Statistical Machine Learning Group at National ICT Australia in Canberra; he was a summer associate at the RAND Corp. in summer 2005. His research interests include nonparametric statistics, machine learning, and the psychology of human decision making, with applications to distributed decision making and signal processing. He is a Student Member of the IEEE.

Sanjeev R. Kulkarni (kulkarni@princeton.edu) received his Ph.D. from the Massachusetts Institute of Technology (MIT) in 1991. From 1985 to 1991 he was a member of the technical staff at MIT Lincoln Laboratory. Since 1991, he has been with Princeton University where he is currently a professor of electrical engineering. He received a 1992 ARO Young Investigator Award, a 1994 NSF Young Investigator Award, and several teaching awards at Princeton University. He was an associate editor for *IEEE Transactions on Information Theory*. His research interests include statistical pattern recognition, nonparametric estimation, learning and adaptive systems, information theory, wireless networks, and image/video processing. He is a Fellow of the IEEE.

H. Vincent Poor (poor@princeton.edu) received the Ph.D. degree in electrical engineering and computer science from Princeton University in 1977. From 1977 until 1990, he was on the faculty of the University of Illinois at Urbana-Champaign. Since 1990 he has been with Princeton University, where he is the Michael Henry Strater University Professor of Electrical Engineering and Applied Science. His research interests are in the areas of stochastic analysis and statistical signal processing and their applications in wireless networks and related fields. Among his publications is the book *Wireless Networks: Multiuser Detection in Cross-Layer Design* (Springer, 2005). He is a member of the National Academy of Engineering and a Fellow of the IEEE, the American Academy of Arts and Sciences, the Institute of Mathematical Statistics, and the Optical Society of America. In 1990, he was president of the IEEE Information Theory Society, and he is currently the editor-in-chief of *IEEE Transactions on Information Theory*. Recent recognition of his work include a Guggenheim Fellowship (2002–2003) and the 2005 IEEE Education Medal.

REFERENCES

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

[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, 2002.

[3] M.M. Al-Ibrahim and P.K. Varshney, "Nonparametric sequential detection based on multisensor data," in *Proc. 23rd Annu. Conf. Information Science Systems*, Johns Hopkins Univ., Baltimore, MD, Mar. 1989, pp. 157–162.

[4] M. Anthony and P. Bartlett, *Neural Network Learning: Theoretical Foundations*. Cambridge, U.K.: Cambridge Univ. Press, 1999.

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

[6] M. Barkat and P.K. Varshney, "Decentralized CFAR signal detection," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 25, no. 2, pp. 141–149, 1989.

[7] T. Berger, Z. Zhang, and H. Vishwanathan, "The CEO problem," *IEEE Trans. Inform. Theory*, vol. 42, no. 3, pp. 887–902, 1996.

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

[9] D. Blatt and A. Hero, "Distributed maximum likelihood estimation for sensor networks," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, Montreal, Quebec, Canada, May 2004, pp. 929–932.

[10] R. Blum, S. Kassam, and H.V. Poor, "Distributed detection with multiple sensors: Part II—Advanced topics," *Proc. IEEE*, vol. 85, pp. 64–79, 1997.

[11] A. Bordes, S. Ertekin, J. Weston, and L. Bottou, "Fast kernel classifiers with online and active learning," *J. Mach. Learning Res.*, vol. 6, pp. 1579–1619, 2005.

[12] L. Breiman, "Bagging predictors," *Mach. Learning*, vol. 26, no. 2, pp. 123–140, 1996.

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

[14] N. Cesa-Bianchi, Y. Freund, D. Haussler, D.P. Helmbold, R.E. Schapire, and M.K. Warmuth, "How to use expert advice," *J. Assoc. Comput. Mach.*, vol. 44, no. 3, pp. 427–485, 1997.

[15] 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 Processing Mag.,* vol. 23, no. 4, pp. 42–55, June 2006.

[16] J.-F. Chamberland and V.V. Veeravalli, "Asymptotic results for decentralized detection in power constrained wireless sensor networks," *IEEE J. Select. Areas Commun.*, vol. 22, no. 6, pp. 1007–1015, 2004.

[17] N.V. Chawla, L.O. Hall, K.W. Bowyer, and W.P. Kegelmeyer, "Learning ensembles from bites: A scalable and accurate approach," *J. Mach. Learning Res.*, vol. 5, pp. 421–451, 2004.

[18] B. Chen, L. Tong, and P.K. Varshney, "Channel aware distributed detection in wireless sensor networks," *IEEE Signal Processing Mag.,* vol. 23, no. 4, pp. 16–25, June 2006.

[19] T.M. Cover, "Rates of convergence for nearest neighbor procedures," in *Proc. Hawaii Int. Conf. Systems Sciences*, Honolulu, Jan. 1968, pp. 413–415.

[20] A. D'Costa, V. Ramachadran, and A.M. Sayeed, "Distributed classification of Gaussian space-time sources in wireless sensor networks," *IEEE J. Select. Areas Commun,* vol. 22, no. 6, pp. 1026–1036, 2004.

[21] A. D'Costa and A.M. Sayeed, "Collaborative signal processing for distributed classification in sensor networks," in *Proc. 2nd Int. Workshop Information Processing Sensor Networks*, Palo Alto, CA, Apr. 2003, pp. 193–208.

[22] 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 Sensor Networks*, Berkeley, CA, Apr. 2004.

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

[24] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. New York: Wiley-Interscience, 2001.

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

[26] Y. Freund, R.E. Schapire, Y. Singer, and M.K. Warmuth, "Using and combining predictors that specialize," in *Proc. 29th Annu. ACM Symp. Theory Computing*, El Paso, Texas, pp. 334–343, May 1997.

[27] S. Gambs, B. Kégl, and E. Aïmeur, "Privacy-preserving boosting," submitted for publication.

[28] M. Gastpar, M. Vetterli, and P.L. Dragotti, "Sensing reality and communicating bits: A dangerous liaison," *IEEE Signal Processing Mag.,* vol. 23, no. 4, pp. 70–83, June 2006.

[29] 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 Processing Mag.*, vol. 22, no. 4, pp. 70–84, 2005.

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

[31] H.P. Graf, E. Cosatto, L. Bottou, I. Dourdanovic, and V. Vapnik, "Parallel support vector machines: The Cascade SVM," in *Advances in Neural Information Processing Systems*, vol. 17, L. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA: MIT Press, 2005, pp. 521–528.

[32] W. Greblicki and M. Pawlak, "Necessary and sufficient conditions for Bayes risk consistency of recursive kernel classification rule," *IEEE Trans. Inform. Theory*, vol. 33, no. 3, pp. 408–412, 1987.

[33] 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 Sensor Networks*, Berkeley, CA, Apr. 2004, pp. 1–10.

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

[35] J. Han, P.K. Varshney, and V.C. Vannicola, "Some results on distributed nonparametric detection," in *Proc. 29th Conf. Decision Control*, Honolulu, Hawaii, Dec. 1990, pp. 2698–2703.

[36] T.S. Han and S. Amari, "Statistical inference under multiterminal data compression," *IEEE Trans. Inform. Theory*, vol. 44, no. 6, pp. 2300–2324, 1998.

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

[38] E.K. Hussaini, A.A.M. Al-Bassiouni, and Y.A. El-Far, "Decentralized CFAR signal detection," *Signal Process.*, vol. 44, no. 3, pp. 299–307, 1995.

[39] R. Jacobs, M.I. Jordan, S. Nowlan, and G.E. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, no. 1, pp. 125–130, 1991.

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

[41] M. Kearns and H.S. Seung, "Learning from a population of hypotheses," *Mach. Learning*, vol. 18, no. 2-3, pp. 255–276, 1995.

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

[43] J. Kittler, M. Hatef, P.W. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 3, pp. 226–239, 1998.

[44] J.H. Kotecha, V. Ramachandran, and A. Sayeed, "Distributed multi-target classification in wireless sensor networks," *IEEE J. Select. Areas Commun.,* vol. 23, no. 4, pp. 703–713, 2005.

[45] A. Krzyżak, "The rates of convergence of kernel regression estimates and classification rules," *IEEE Trans. Inform. Theory*, vol. 32, no. 5, pp. 668–679, 1986.

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

[47] S.R. Kulkarni, G. Lugosi, and S.S. Venkatesh, "Learning pattern classification: A survey," *IEEE Trans. Inform. Theory*, vol. 44, no. 6, pp. 2178–2206, 1998.

[48] S.R. Kulkarni and S.E. Posner, "Rates of convergence of nearest neighbor estimation under arbitrary sampling," *IEEE Trans. Inform. Theory*, vol. 41, no. 4, pp. 1028–1039, 1995.

[49] S.R. Kulkarni, S.E. Posner, and S. Sandilya, "Data-dependent k_n-nn and kernel estimators consistent for arbitrary processes," *IEEE Trans. Inform. Theory*, vol. 48, no. 10, pp. 2785–2788, 2002.

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

[51] D. Li, K. Wong, Y.H. Hu, and A. Sayeed, "Detection, classification, and tracking of targets," *IEEE Signal Processing Mag.*, vol. 19, no. 2, pp. 17–29, 2002.

[52] N. Littlestone and M. Warmuth, "The weighted majority algorithm," *Inform. Computation*, vol. 108, no. 2, pp. 212–261, 1994.

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

[54] Z.-Q. Luo, "An isotropic universal decentralied estimation scheme for a bandwidth constrained ad hoc sensor network," *IEEE J. Select. Areas Commun.,* vol. 23, no. 4, pp. 735–744, 2005.

[55] Z.Q. Luo, "Universal decentralized estimation in a bandwidth constraint sensor network," *IEEE Trans. Inform. Theory*, vol. 51, no. 6, pp. 2210–2219, 2005.

[56] T. Mitchell, *Machine Learning*. New York: McGraw Hill, 1997.

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

[58] G. Morvai, S.R. Kulkarni, and A.B. Nobel, "Regression estimation from an individual stable sequence," *Statistics*, vol. 33, pp. 99–118, 1999.

[59] A. Nakamura, J. Takeuchi, and M. Abe, "Efficient distribution-free learning of simple concepts," *Ann. Math. Artificial Intell.*, vol. 23, no. 1–2, pp. 53–82, 1998.

[60] A. Nasipuri and S. Tantaratana, "Nonparametric distributed detection using Wilcoxin statistics," *Signal Process.*, vol. 57, no. 2, pp. 139–146, 1997.

[61] A. Nedic and D. Bertsekas, "Incremental subgradient methods for nondifferentiable optimization," MIT, Cambridge, MA, Tech. Rep. LIDS-P-2460, 1999.

[62] "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.

[63] 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, pp. 134–152, 2005.

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

[65] A.B. Nobel, "Limits to classification and regression estimation from ergodic processes," *Ann. Statist.*, vol. 27, no. 1, pp. 262–273, 1999.

[66] A.B. Nobel and T.M. Adams, "Estimating a function from ergodic samples with additive noise," *IEEE Trans. Inform. Theory*, vol. 47, no. 7, pp. 2895–2902, 2001.

[67] A.B. Nobel, G. Morvai, and S. Kulkarni, "Density estimation from an individual sequence," *IEEE Trans. Inform. Theory*, vol. 44, no. 2, pp. 537–541, 1998.

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

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

[70] M.A. Paskin and G.D. Lawrence, "Junction tree algorithms for solving sparse linear systems," EECS Dept., UC, Berkeley, Tech. Rep. UCB/CSD-03-1271, 2003.

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

[72] 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 Processing Mag.*, vol. 22, no. 4, pp. 54–69, 2005.

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

[74] K. Plarre and P.R. Kumar, "Extended message passing algorithm for inference in loopy Gaussian graphical models," *Ad Hoc Networks*, vol. 2, pp. 153–169, 2004.

[75] J.B. Predd, S.R. Kulkarni, and H.V. Poor, "Consistency in a model for distributed learning with specialists," in *Proc. 2004 IEEE Int. Symp. Information Theory*, Chicago, IL, June 2004, p. 465.

[76] 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 (invited)*, San Diego, CA, July–Aug. 2005.

[77] J.B. Predd, S.R. Kulkarni, and H.V. Poor, "Consistency in models for distributed learning under communication constraints," *IEEE Trans. Inform. Theory*, vol. 52, no. 1, pp. 52–63, 2006.

[78] 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.

[79] F. Provost and D.N. Hennessy, "Scaling up: Distributed machine learning with cooperation," in *Proc. 13th Nat. Conf. Artificial Intelligence*, Portland, OR, 1996, pp. 74–79.

[80] J.R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kauffman, 1992.

[81] M.G. Rabbat and R.D. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE J. Special Areas Commun.*, vol. 23, no. 4, pp. 798–808, 2006.

[82] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proc. 3rd Int. Symp. Information Processing Sensor Networks*, Berkeley, CA, Apr. 2004.

[83] A. Ribeiro and G.B. Giannakis, "Bandwidth-constrained distributed estimation for wireless sensor networks, part I: Gaussian PDF," *IEEE Trans. Signal Processing*, vol. 54, no. 3, pp. 1131– 1143, 2006.

[84] A. Ribeiro and G.B. Giannakis, "Bandwidth-constrained distributed estimation for wireless sensor networks, part II: Unknown PDF," *IEEE Trans. Signal Processing*, to be published.

[85] G. Roussas, "Nonparametric estimation in Markov processes," *Ann. Inst. Statist. Math.*, vol. 21, pp. 73–87, 1967.

[86] B. Schölkopf and A. Smola, *Learning with Kernels*, 1st ed. Cambridge, MA: MIT Press, 2002.

[87] S.D. Servetto, "On the feasibility of large scale wireless sensor networks," in *Proc. 40th Annu. Allerton Conf. Communication, Control, Computing*, Monticello, Illinois, Oct. 2002.

[88] 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, Mobile Computing*, Maui, Hawaii, June 2005, vol. 2, pp. 969–974.

[89] S.-H. Son, S.R. Kulkarni, S.C. Schwartz, and M. Roan, "Communication-estimation tradeoffs in wireless sensor networks," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Philadelphia, PA, Mar. 2005, vol. 5, pp. 1065–1068.

[90] C.J. Stone, "Consistent nonparametric regression," *Ann. Statist.*, vol. 5, no. 4, pp. 595–645, 1977.

[91] Y. Sung, L. Tong, and H.V. Poor, "Neyman-Pearson detection of Gauss-Markov signals in noise: Closed-form error exponent and properties," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1354–1365, 2006.

[92] Y. Sung, L. Tong, and H.V. Poor, "A large deviations approach to sensor scheduling for detection of correlated random fields," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Philadelphia, PA, 19–23 Mar. 2005.

[93] J.N. Tsitsiklis, "Decentralized detection," in *Advances in Statistical Signal Processing*, H.V. Poor and J.B. Thomas, Eds. Greenwich, CT: JAI Press, 1993, pp. 297–344.

[94] L.G. Valiant, "A theory of the learnable," *Commun. ACM*, vol. 27, no. 11, pp. 1134–1142, 1984.

[95] V.N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1991.

[96] P.K. Varshney, *Distributed Detection and Data Fusion*. New York: Springer-Verlag, 1996.

[97] R. Vishwanathan and A. Ansari, "Distributed detection of a signal in generalized Gaussian noise," *IEEE Trans. Acoustic, Speech, Signal Processing*, vol. 37, no. 5, pp. 775–778, 1989.

[98] R. Vishwanathan and P.K. Varshney, "Distributed detection with multiple sensors: Part I—Fundamentals," *Proc. IEEE*, vol. 85, no. 1, pp. 54–63, 1997.

[99] H. Viswanathan and T. Berger, "The quadratic Gaussian CEO problem," *IEEE Trans. Inform. Theory*, vol. 43, no. 5, pp. 1549–1559, 1997.

[100] J.-J. Xiao and Z.-Q. Luo, "Universal decentralized detection in a bandwidth constraint sensor network," *IEEE Trans. Signal Processing*, vol. 53, no. 8, pp. 2617–2624, 2005.

[101] J.-J. Xiao and Z.-Q. Luo, "Universal decentralized estimation in an inhomogeneous sensing environment," *IEEE Trans. Inform. Theory*, vol. 51, no. 10, pp. 3564–3575, 2005.

[102] J.-J. Xiao, A. Ribeiro, Z.-Q. Luo, and G.B. Giannakis, "Distributed compression-estimation using wireless sensor networks," *IEEE Signal Processing Mag.,* vol. 23, no. 4, pp. 26–41, June 2006.

[103] S. Yakowitz, "Nonparametric density and regression estimation from Markov sequences without mixing assumptions," *J. Multivariate Anal.*, vol. 30, no. 1, pp. 124–136, 1989.

[104] S. Yakowitz, "Nearest neighbor regression estimation for null-recurrent Markov time series," *Stochastic Processes Their Applicat.*, vol. 48, no. 2, pp. 311–318, 1993.

[105] K. Yamanishi, "Distributed cooperative Bayesian learning," in *Proc. Conf. Learning Theory*, Nashville, Tennessee, July 1997, pp. 250–262. SP