



MIT Open Access Articles

Linear Dimensionality Reduction for Margin-Based Classification: High-Dimensional Data and Sensor Networks

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

Citation	Varshney, Kush R., and Alan S. Willsky. Linear Dimensionality Reduction for Margin-Based Classification: High-Dimensional Data and Sensor Networks. IEEE Transactions on Signal Processing 59, no. 6 (June 2011): 2496-2512.
As Published	http://dx.doi.org/10.1109/tsp.2011.2123891
Publisher	Institute of Electrical and Electronics Engineers
Version	Author's final manuscript
Citable link	http://hdl.handle.net/1721.1/81207
Terms of Use	Creative Commons Attribution-Noncommercial-Share Alike 3.0
Detailed Terms	http://creativecommons.org/licenses/by-nc-sa/3.0/

Linear Dimensionality Reduction for Margin-Based Classification: High-Dimensional Data and Sensor Networks

Kush R. Varshney, *Member, IEEE*, and Alan S. Willsky, *Fellow, IEEE*

Abstract—Low-dimensional statistics of measurements play an important role in detection problems, including those encountered in sensor networks. In this work, we focus on learning low-dimensional linear statistics of high-dimensional measurement data along with decision rules defined in the low-dimensional space in the case when the probability density of the measurements and class labels is not given, but a training set of samples from this distribution is given. We pose a joint optimization problem for linear dimensionality reduction and margin-based classification, and develop a coordinate descent algorithm on the Stiefel manifold for its solution. Although the coordinate descent is not guaranteed to find the globally optimal solution, crucially, its alternating structure enables us to extend it for sensor networks with a message-passing approach requiring little communication. Linear dimensionality reduction prevents overfitting when learning from finite training data. In the sensor network setting, dimensionality reduction not only prevents overfitting, but also reduces power consumption due to communication. The learned reduced-dimensional space and decision rule is shown to be consistent, and its Rademacher complexity is characterized. Experimental results are presented for a variety of datasets, including those from existing sensor networks, demonstrating the potential of our methodology in comparison with other dimensionality reduction approaches.

Index Terms—supervised classification, linear dimensionality reduction, Stiefel manifold, sensor networks

I. INTRODUCTION

SENSOR networks are systems used for distributed detection and data fusion that operate with severe resource limitations; consequently, minimizing complexity in terms of communication and computation is critical [3]. A current interest is in deploying wireless sensor networks with nodes that take measurements using many heterogeneous modalities such as acoustic, infrared, and seismic to monitor volcanoes [4], detect intruders [5], [6], and perform many other classification tasks. Sensor measurements may contain much redundancy,

both within the measurement dimensions of a single sensor and between measurement dimensions of different sensors due to spatial correlation.

Resources can be conserved if sensors do not transmit irrelevant or redundant data, but it is usually not known in advance which measurement dimensions or combination of dimensions are most useful for the detection or classification task. The transmission of irrelevant and redundant data can be avoided through dimensionality reduction; specifically, a low-dimensional representative form of measurements may be transmitted by sensors to a fusion center, which then detects or classifies based on those low-dimensional measurement representations. As measurements or low-dimensional measurement representations are transmitted from sensor to sensor, eventually reaching the fusion center, dimensionality reduction at the parent node eliminates redundancy between parent and child node measurements. Even a reduction from two-dimensional measurements to one-dimensional features is significant in many hostile-environment monitoring and surveillance applications.

Decision rules in detection problems, both in the sensor network setting and not, are often simplified through sufficient statistics such as the likelihood ratio [7]. Calculation of a sufficient statistic losslessly reduces the dimensionality of high-dimensional measurements before applying a decision rule defined in the reduced-dimensional space, but requires knowledge of the probability distribution of the measurements. The statistical learning problem *supervised classification* deals with the case when this distribution is unknown, but a set of labeled samples from it, known as the training dataset, is available. For the most part, however, supervised classification methods (not adorned with feature selection) produce decision rules defined in the full high-dimensional measurement space rather than in a reduced-dimensional space, motivating feature selection or dimensionality reduction for classification.

In this paper, we propose a method for simultaneously learning both a dimensionality reduction mapping and a classifier defined in the reduced-dimensional space. Not only does dimensionality reduction simplify decision rules, but it also decreases the probability of classification error by preventing overfitting when learning from a finite training dataset [8]–[11]. We focus on *linear* dimensionality reduction mappings represented by matrices on the Stiefel manifold [12] and on *margin-based* classifiers, a popular and effective class of classifiers that includes logistic regression, the support vector

Manuscript received July 11, 2010; revised December 30, 2010. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Anna Scaglione. Portions of the material in this paper were first presented in [1], [2]. This work was supported in part by a National Science Foundation Graduate Research Fellowship, by a MURI funded through ARO Grant W911NF-06-1-0076, by the Air Force Office of Scientific Research under Award No. FA9550-06-1-0324, and by Shell International Exploration and Production, Inc. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the Air Force.

K. R. Varshney was with and A. S. Willsky is with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: krvarshn@us.ibm.com; willsky@mit.edu).

machine (SVM), and the geometric level set (GLS) classifier [13]–[15]. The importance of the Stiefel manifold is its role as the set of all linear subspaces with basis specified and hence it provides precisely the right object for exploring different subspaces on which to project measurements.

Many methods for linear dimensionality reduction, including the popular principal component analysis (PCA) and Fisher discriminant analysis (FDA), can be posed as optimization problems on the Stiefel or Grassmann manifold with different objectives [12]. In this paper, we propose an optimization problem on the Stiefel manifold whose objective is that of margin-based classification and develop an iterative coordinate descent algorithm for its solution. PCA, FDA, and other methods do not have margin-based classification as their objective and are consequently suboptimal with respect to that objective. Coordinate descent is not guaranteed to find the global optimum; however, as seen later in the paper, an advantage of coordinate descent is that it is readily implemented in distributed settings and tends to find good solutions in practice. We successfully demonstrate the learning procedure on several real datasets from different applications.

The idea of learning linear dimensionality reduction mappings from labeled training data specifically for the purpose of classification is not new. For example, the goal of FDA is classification, but it assumes that the class-conditional distributions generating the data are Gaussian with identical covariances; it is also not well suited to datasets of small cardinality [16]. We reserve discussion of several such methods until Section I-A.¹ Our work fits into the general category of learning data representations that have traditionally been learned in an unsupervised manner, appended with known class labels and consequently supervision. Examples from this category include learning undirected graphical models [20], sparse signal representations [21], [22], directed topic models [23], [24], quantizer codebooks [25], and linear dimensionality reduction matrices, which is the topic of this paper and others described in Section I-A.

Statistical learning theory characterizes the phenomenon of overfitting when there is finite training data. The generalization error of a classifier—the probability of misclassification on new unseen measurements (the quantity we would ideally like to minimize)—can be bounded by the sum of two terms [8]: the classification error on the training set, and a complexity term, e.g. the Rademacher complexity [26], [27]. We analytically characterize the Rademacher complexity as a function of the dimension of the reduced-dimensional space in this work. Finding it to be an increasing function of the dimension, we can conclude that dimensionality reduction does in fact prevent overfitting and that there exists some optimal reduced dimension.

As the cardinality of the training dataset grows, the generalization error of a *consistent* classifier converges to the Bayes optimal probability of error, i.e., the error probability had the joint probability distribution been known. We show

that our proposed joint linear dimensionality reduction and margin-based classification method is consistent.

The problem of *distributed* detection has been an object of study during the last thirty years [28]–[31], but the majority of the work has focused on the situation when either the joint probability distribution of the measurements and labels or the likelihood functions of the measurements given the labels are assumed known. Recently, there has been some work on supervised classification for distributed settings [32]–[34], but in that work sensors take scalar-valued measurements and dimensionality reduction is not involved. Previous work on the linear dimensionality reduction of sensor measurements in distributed settings, including [35]–[37] and references therein, have estimation rather than detection or classification as the objective.

In this paper, we show how the linear dimensionality reduction of heterogeneous data specifically for margin-based classification may be distributed in a tree-structured multisensor data fusion network with a fusion center via individual Stiefel manifold matrices at each sensor. The proposed coordinate descent learning algorithm is amenable to distributed implementation. In particular, we extend the coordinate descent procedure so that it can be implemented in tree-structured sensor networks through a message-passing approach with the amount of communication related to the reduced dimension rather than the full measurement dimension. The ability to be distributed is a key strength of the coordinate descent optimization approach.

Multisensor networks lead to issues that do not typically arise in statistical learning, where generalization error is the only criterion. In sensor networks, resource usage presents an additional criterion to be considered, and the architecture of the network presents additional design freedom. In wireless sensor networks, the distance between nodes affects energy usage in communication, and must therefore be considered in selecting network architecture. We give classification results on real datasets for different network architectures and touch on these issues empirically.

A. Relationship to Prior Work

The most popular method of linear dimensionality reduction for data analysis is PCA. PCA and several other methods only make use of the measurement vectors, not the class labels, in finding a dimensionality reduction mapping. If the dimensionality reduction is to be done in the context of supervised classification, the class labels should also be used. Several *supervised* linear dimensionality reduction methods exist in the literature. We can group these methods into three broad categories: those that separate likelihood functions according to some distance or divergence [38]–[44], those that try to make the probability of the labels given the measurements and the probability of the labels given the dimensionality-reduced measurements equal [45]–[50], and those that attempt to minimize a specific classification or regression objective [12], [51]–[54].

As mentioned previously in the section, FDA assumes that the likelihood functions are Gaussian with the same covariance

¹Our paper focuses on general linear dimensionality reduction and not on feature subset selection, which is a separate topic in its own right, e.g. see [17]–[19].

and different means. It returns a dimensionality reduction matrix on the Stiefel manifold that maximally separates (in Euclidean distance) the clusters of the different labels [12]. The method of [39] also assumes Gaussian likelihoods with the same covariance and different means, but with an even stronger assumption that the covariance matrix is a scalar multiple of the identity. The probability of error is explicitly minimized using gradient descent; the gradient updates to the dimensionality reduction matrix do not enforce the Stiefel manifold constraint, but the Gram-Schmidt orthonormalization procedure is performed after every step to obtain a matrix that does meet the constraint. With a weaker assumption only that the likelihood functions are Gaussian, but without restriction on the covariances, other methods maximize Bhattacharyya divergence or Chernoff divergence, which are surrogates for minimizing the probability of error [43].

The method of [38], like FDA, maximally separates the clusters of the different labels but does not make the strong Gaussian assumption. Instead, it performs kernel density estimation of the likelihoods and separates those estimates. The optimization is gradient ascent and orthonormalization is performed after every step. Similarly, information preserving component analysis also performs kernel density estimation and maximizes Hellinger distance, another surrogate for minimizing the probability of error, with optimization through gradient ascent and the Stiefel manifold constraint maintained in the gradient steps [44]. Other approaches with information-theoretic criteria include [40]–[42].

Like [38], [44], the method of [49] also estimates probability density functions for use in the criterion for linear dimensionality reduction. The particular criterion, however, is based on the idea that the dimensionality reduction mapping should be such that the probability of the class labels conditioned on the unreduced measurements equal the probability conditioned on the reduced measurements. The same criterion appears in [45], [46], [48], [50] and many references given in [47]. These papers describe various methods of finding dimensionality reduction mappings to optimize the criterion with different assumptions.

Some supervised dimensionality reduction methods explicitly optimize a classification or regression objective. A linear regression objective and a regression parameter/Stiefel manifold coordinate descent algorithm is developed in [53]. The support vector singular value decomposition machine of [52] has a joint objective for dimensionality reduction and classification with the hinge loss function. However, the matrix it produces is not guaranteed to be on the Stiefel manifold, and the space in which the classifier is defined is not exactly the dimensionality-reduced image of the high-dimensional space. It also changes the regularization term from what is standardly used for the SVM. Maximum margin discriminant analysis is another method based on the SVM; it finds the reduced-dimensional features one by one instead of giving a complete matrix at once and it does not simultaneously give a classifier [54]. The method of [12], [51] is based on the nearest neighbor classifier.

The objective function and optimization procedure we propose in Section II has some similarities to many of the methods

discussed, but also some key differences. First of all, we do not make *any* assumption, and indeed do not explicitly make use of any assumptions on the statistics of likelihood functions (e.g., no assumption of Gaussianity is employed). Moreover, our method does not require nor involve estimation of the probability density functions under the two hypotheses nor of the likelihood ratio. Indeed, we are directly interested only in learning decision boundaries and using margin-based loss functions to guide both this learning *and* the optimization over the Stiefel manifold to determine the reduced-dimensional space in which decision making is to be performed. Density estimation is a harder problem than finding classifier decision boundaries and it is well known that when learning from finite data, it is best to only solve the problem of interest and nothing more. Similarly, the desire that the conditional distributions of the class label given the high-dimensional and reduced-dimensional measurements be equal is more involved than wanting good classification performance in the reduced-dimensional space.

Rather than nearest neighbor classification or linear regression, the objective in the method we propose is margin-based classification. Our method finds all reduced-dimensional features in a joint manner, and gives both the dimensionality reduction mapping and the classifier as output. Unlike in [52], the classifier is defined exactly without approximation in the reduced-dimensional subspace resulting from applying the dimensionality reduction matrix that is found. Additionally, the regularization term and consequently inductive bias of the classifier is left unchanged.

The preceding represent the major conceptual differences between our framework and that considered in previous work. We use coordinate descent optimization procedures in Section II, which are also employed in other works, e.g. [52], [53], but the setting in which we use these are new. Our framework also allows us to develop some new theoretical results on consistency and Rademacher complexity. Moreover, as developed in Section III, our framework allows a natural generalization to distributed dimensionality reduction for classification in sensor networks, a problem that has not been considered previously.

Ji and Ye presented an approach to linear dimensionality reduction for classification with linear decision boundaries [55] after the initial presentation of this work [1], which is similar to our formulation as well as the formulation of [53]. Ji and Ye restrict themselves to the regularization term of the SVM and either a regression objective like [53], or the hinge loss. In our formulation, any regularization term and any margin-based loss function may be used, and the decision boundaries are generally nonlinear. With the hinge loss, the optimization in [55] is through coordinate descent similar to ours, but the dimensionality reduction matrix optimization step is carried out via a convex-concave relaxation (which is not guaranteed to find the optimum of the true unrelaxed problem) rather than gradient descent along Stiefel manifold geodesics that we do. The work of Ji and Ye also considers the learning problem when training samples may have either zero, one, or more than one assigned class label, which is known as multilabel classification [56] and is not the focus of our work.

B. Organization of Paper

The paper is organized as follows. Section II combines the ideas of margin-based classification and optimization on the Stiefel manifold to give a joint linear dimensionality reduction and classification objective as well as an iterative algorithm. An analysis of Rademacher complexity and consistency is also presented in the section. Section III shows how the basic method of Section II extends to multisensor data fusion networks, including wireless sensor networks. In Section IV, an illustrative example and results on several real datasets are given. Also given are experimental results of classification performance as a function of transmission power in wireless sensor networks. Section V concludes.

II. LINEAR DIMENSIONALITY REDUCTION FOR MARGIN-BASED CLASSIFICATION

In this section, we formulate a problem for composite dimensionality reduction and margin-based classification. We develop a coordinate descent minimization procedure for this formulation, characterize the complexity of the formulation from a statistical learning theory perspective, and show the consistency of the formulation.

A. Formulation

Consider the binary detection or classification problem with measurement vectors $\mathbf{X} \in \Omega \subset \mathbb{R}^D$ and class labels $Y \in \{+1, -1\}$ drawn according to the probability density function $p_{\mathbf{X},Y}(\mathbf{x}, y)$. We would like to find the classifier $\hat{y} : \Omega \rightarrow \{+1, -1\}$ that minimizes the error probability $\Pr[Y \neq \hat{y}(\mathbf{X})]$. We do not have access to $p_{\mathbf{X},Y}(\mathbf{x}, y)$, but instead are given training data $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$. The true objective we would like to minimize in learning \hat{y} is the generalization error $\Pr[Y \neq \hat{y}(\mathbf{X})]$, but a direct minimization is not possible since the joint distribution of \mathbf{X} and Y is not known. In practice, the classifier \hat{y} is selected from a function class \mathcal{F} to minimize a loss function of the training data.

Margin-based classifiers take the form $\hat{y}(\cdot) = \text{sign}(\varphi(\cdot))$, where φ is a decision function whose specifics are tied to the specific margin-based classifier. The decision function is chosen to minimize the functional:

$$L(\varphi) = \sum_{j=1}^n \ell(y_j \varphi(\mathbf{x}_j)) + \lambda J(\varphi), \quad (1)$$

where the value $y\varphi(\mathbf{x})$ is known as the margin; it is related to the distance between \mathbf{x} and the classifier decision boundary $\varphi(\mathbf{x}) = 0$. The function ℓ is known as a margin-based loss function. Examples of such functions are the logistic loss function:

$$\ell_{\text{logistic}}(z) = \log(1 + e^{-z})$$

and the hinge loss function:

$$\ell_{\text{hinge}}(z) = \max\{0, 1 - z\}.$$

The second term on the right side of (1), with non-negative weight λ , represents a regularization term that penalizes the complexity of the decision function [13], [14]. In the kernel SVM, ℓ is the hinge loss, the decision functions φ are in

a reproducing kernel Hilbert space \mathcal{H} , and J is the squared norm in that space $\|\varphi\|_{\mathcal{H}}^2$ [13], [14]. In the GLS classifier, any margin-based loss function may be used and the decision functions are in the space of signed distance functions [2], [15]. The magnitude of $\varphi(\mathbf{x})$ equals the Euclidean distance of \mathbf{x} to the decision boundary. The regularization term J is the surface area of the zero level set of φ , i.e., $J(\varphi) = \oint_{\varphi=0} ds$, where ds is an infinitesimal surface area element on the decision boundary.

The new contribution of this section is the formulation of a joint linear dimensionality reduction and classification minimization problem by extension of the margin-based functional (1). The decision function φ is defined in the reduced d -dimensional space and a linear dimensionality reduction mapping appears in its argument, but otherwise, the classification objective is left unchanged. In particular, the regularization term J is not altered, thereby allowing any regularized margin-based classifier to be extended for dimensionality reduction.

The margin-based classification objective is extended to include a matrix $\mathbf{A} \in \mathbb{R}^{D \times d}$ with elements a_{ij} as follows:

$$L(\varphi, \mathbf{A}) = \sum_{j=1}^n \ell(y_j \varphi(\mathbf{A}^T \mathbf{x}_j)) + \lambda J(\varphi), \quad (2)$$

with the constraint that \mathbf{A} lie on the *Stiefel manifold* of $D \times d$ matrices, i.e. $\mathbf{A} \in \mathcal{V}(D, d)$, where

$$\mathcal{V}(D, d) = \{\mathbf{A} \in \mathbb{R}^{D \times d}, d \leq D | \mathbf{A}^T \mathbf{A} = \mathbf{I}\}. \quad (3)$$

With a data vector $\mathbf{x} \in \mathbb{R}^D$, $\mathbf{A}^T \mathbf{x}$ is in d dimensions. Typically—and especially in our framework—we are uninterested in scalings of the reduced-dimensional data $\mathbf{A}^T \mathbf{x}$, so we limit the set of possible matrices to those which involve orthogonal projection, i.e., to the Stiefel manifold.

The formulation as presented is for a fixed value of d . If generalization error is the only criterion, then any popular model selection method from the machine learning literature, including those based on cross-validation, bootstrapping, and information criteria, can be used to find a good value for the reduced dimension d . However, other criteria besides generalization error become important in various settings, including sensor networks. System resource usage is one such criterion; it is not typically statistical in nature and is often a deterministic increasing function of d . As such, it may be used as an additional cost with information criteria or as a component in modified cross-validation and bootstrapping. If different types of errors such as false alarms and missed detections incur different costs, then the criterion is not strictly generalization error, but cross-validation and bootstrapping may be modified accordingly.

B. Coordinate Descent Minimization

An option for performing the minimization of $L(\varphi, \mathbf{A})$ given in (2) is coordinate descent: alternating minimizations with fixed \mathbf{A} and with fixed φ . The problem is conceptually similar to level set image segmentation along with pose estimation for a shape prior [57]. With \mathbf{A} fixed, we are left with a

standard margin-based classification problem in the reduced-dimensional space. The optimization step may be performed using standard methods for margin-based classifiers.

With φ fixed, we have a problem of minimizing a function of \mathbf{A} lying on the Stiefel manifold. For differentiable functions, several iterative minimization algorithms exist [58]–[60]. The function $L(\mathbf{A}) = \sum_{j=1}^n \ell(y_j \varphi(\mathbf{A}^T \mathbf{x}_j))$ is differentiable with respect to \mathbf{A} for differentiable loss functions. Using $\mathbf{L}_\mathbf{A}$ to denote the $D \times d$ matrix with elements $\partial L / \partial a_{ij}$, the first derivative is:

$$\mathbf{L}_\mathbf{A} = \sum_{j=1}^n y_j \ell'(y_j \varphi(\mathbf{A}^T \mathbf{x}_j)) \times \mathbf{x}_j [\varphi_1(\mathbf{A}^T \mathbf{x}_j) \cdots \varphi_d(\mathbf{A}^T \mathbf{x}_j)]. \quad (4)$$

Note that \mathbf{x}_j is a $D \times 1$ vector and that $[\varphi_1(\mathbf{A}^T \mathbf{x}_j) \cdots \varphi_d(\mathbf{A}^T \mathbf{x}_j)]$ is a $1 \times d$ vector, where $\varphi_k(\cdot)$ is the partial derivative of the decision function with respect to dimension k . For the logistic loss function:

$$\ell'_{\text{logistic}}(z) = -\frac{e^{-z}}{1 + e^{-z}}$$

and for the hinge loss function:

$$\ell'_{\text{hinge}}(z) = -\text{step}(1 - z),$$

where $\text{step}(\cdot)$ is the Heaviside step function.

We perform gradient descent along geodesics of the Stiefel manifold [58]. The gradient is:

$$\mathbf{G} = \mathbf{L}_\mathbf{A} - \mathbf{A} \mathbf{L}_\mathbf{A}^T \mathbf{A}. \quad (5)$$

Starting at an initial $\mathbf{A}(0)$, a step of length τ in the direction $-\mathbf{G}$ to $\mathbf{A}(\tau)$ is:

$$\mathbf{A}(\tau) = \mathbf{A}(0)\mathbf{M}(\tau) + \mathbf{Q}\mathbf{N}(\tau), \quad (6)$$

where $\mathbf{Q}\mathbf{R}$ is the QR decomposition of $(\mathbf{A}\mathbf{A}^T \mathbf{G} - \mathbf{G})$, and

$$\begin{bmatrix} \mathbf{M}(\tau) \\ \mathbf{N}(\tau) \end{bmatrix} = \exp \left\{ \tau \begin{bmatrix} -\mathbf{A}^T \mathbf{G} & -\mathbf{R}^T \\ \mathbf{R} & \mathbf{0} \end{bmatrix} \right\} \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix}.$$

The step size τ may be optimized by a line search.

The coordinate descent is not guaranteed to find the global optimum, only a local optimum; however, as seen in the illustrative example in Section IV-A, even poor initializations lead to the globally optimal solution in practice. For the results given in Section IV-B, \mathbf{A} is initialized by making use of estimates of the mutual informations between the label y and individual data dimensions x_k , $k = 1, \dots, D$. Mutual information provides an indication of whether a measurement dimension is individually relevant for classification, and thus projection onto dimensions with high mutual information is a good starting point. Of course, these dimensions may be correlated, and that is precisely what the Stiefel manifold optimization iterations uncover. The first column of \mathbf{A} is taken to be the canonical unit vector corresponding to the dimension with the largest mutual information. The second column of \mathbf{A} is taken to be the canonical unit vector corresponding to the dimension with the second largest mutual information, and so on. The last, i.e. d th, column of \mathbf{A} is zero in the rows already containing ones in the first $(d - 1)$ columns, and

nonzero in the remaining rows with values proportional to the mutual informations of the remaining dimensions. Kernel density estimation is used in estimating mutual information.

C. Rademacher Complexity

The generalization error can be bounded by the sum of the error of \hat{y} on the training set, and a penalty that is larger for more complex \mathcal{F} . One such penalty is the Rademacher complexity $\hat{R}_n(\mathcal{F})$ [26], [27]. A classifier with good generalizability balances training error and complexity; this is known as the *structural risk minimization principle* [8].

With probability greater than or equal to $1 - \delta$, Bartlett and Mendelson give the following bound on the generalization error for a specified decision rule \hat{y} [27]:

$$\Pr[Y \neq \hat{y}(\mathbf{X})] \leq \frac{1}{n} \sum_{j=1}^n \mathbb{I}(y_j \neq \hat{y}(\mathbf{x}_j)) + \frac{\mathbb{E}[\hat{R}_n(\mathcal{F})]}{2} + \sqrt{\frac{\ln(1/\delta)}{2n}}, \quad (7)$$

where \mathbb{I} is an indicator function. The first term on the right hand side is the training error and the second term is complexity. As discussed in [9]–[11], dimensionality reduction reduces classifier complexity and thus prevents overfitting. Here, we analytically characterize the Rademacher complexity term $\hat{R}_n(\mathcal{F})$ for the joint linear dimensionality reduction and margin-based classification method proposed in this paper. It is shown in [61] that the Rademacher average of a function class \mathcal{F} satisfies:

$$\hat{R}_n(\mathcal{F}) \leq 2\epsilon + \frac{4\sqrt{2}}{\sqrt{n}} \int_{\frac{\epsilon}{4}}^{\infty} \sqrt{H_{\rho_{\infty}, \epsilon'}(\mathcal{F})} d\epsilon', \quad (8)$$

where $H_{\rho_{\infty}, \epsilon}(\mathcal{F})$ is the ϵ -entropy of \mathcal{F} with respect to the L_{∞} metric.²

In classification, it is always possible to scale and shift the data and this is often done in practice. Forgoing some bookkeeping and without losing much generality, we consider the domain of the unreduced measurement vectors to be the unit hypercube, that is $\mathbf{x} \in \Omega = [0, 1]^D$. The reduced-dimensional domain is then the zonotope³ $Z = \mathbf{A}^T \Omega \subset \mathbb{R}^d$, where \mathbf{A} is on the Stiefel manifold. We denote the set of decision functions φ defined on Ω as \mathcal{F}_{Ω} and those defined on Z as \mathcal{F}_Z .

Given the generalization bound based on Rademacher complexity (7) and the Rademacher complexity term (8), we

²The ϵ -covering number of a metric space is the minimal number of sets with radius not exceeding ϵ required to cover that space; the ϵ -entropy is the base-two logarithm of the ϵ -covering number [62]. The L_{∞} metric is $\rho_{\infty}(\varphi_1, \varphi_2) = \sup |\varphi_1(\mathbf{x}) - \varphi_2(\mathbf{x})|$.

³The set $Z = \mathbf{A}^T [0, 1]^D \subset \mathbb{R}^d$, the orthogonal shadow cast by $[0, 1]^D$ due to the projection $\mathbf{A} \in \mathcal{V}(D, d)$, is a zonotope, a particular type of polytope that is convex, centrally-symmetric, and whose faces are also centrally-symmetric in all lower dimensions [63], [64]. For reference, Fig. 1 shows several zonotopes for $D = 4$ and $d = 2$. The matrix \mathbf{A}^T is known as the generator of the zonotope Z ; we use the notation $Z(\mathbf{A})$ to denote the zonotope generated by \mathbf{A}^T . Also, let

$$\mathcal{Z}(D, d) = \{Z(\mathbf{A}) | \mathbf{A} \in \mathcal{V}(D, d)\}. \quad (9)$$

Although the relationship between zonotopes and their generators is not bijective, zonotopes provide a good means of visualizing Stiefel manifold matrices, especially when $d = 2$.

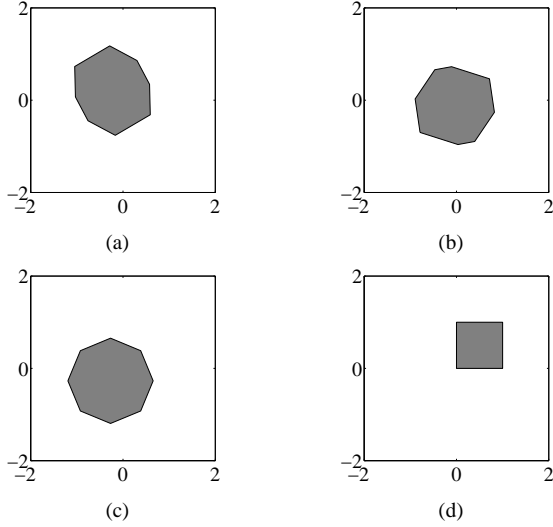


Fig. 1. Several zonotopes in $\mathcal{Z}(4, 2)$.

must find an expression for $H_{\rho_{\infty}, \epsilon}(\mathcal{F}_Z)$ to characterize the prevention of overfitting by linear dimensionality reduction. The function class \mathcal{F}_Z is tied to the specific margin-based classification method employed. In order to make concrete statements, we select the GLS classifier; similar analysis may be performed for other margin-based classifiers such as the kernel SVM. Such analysis would also be similar to [11]. As mentioned in Section II-A, the decision function φ in the GLS classifier is a signed distance function and \mathcal{F}_Z is the set of all signed distance functions whose domain is the zonotope Z .

For classification without dimensionality reduction, it is shown in [15] that

$$H_{\rho_{\infty}, \epsilon}(\mathcal{F}_\Omega) \leq \left\lceil \frac{1}{\epsilon} \right\rceil^D. \quad (10)$$

This result follows from the fact that $\lceil 1/\epsilon \rceil^D$ D -dimensional hypercubes with side of length ϵ fit as a Cartesian grid into $\Omega = [0, 1]^D$. To find an expression for the ϵ -entropy of the dimensionality-reduced GLS classifier, the same analysis applies and consequently, we need to determine how many d -dimensional hypercubes with side of length ϵ fit into Z . The number of small hypercubes that fit inside Z is related to its content $V(Z)$.

An upper bound for $V(Z)$ is developed in [63] that is asymptotically of the correct order of magnitude for fixed d as D goes to infinity. Specifically,

$$V(Z) \leq \omega_d \left(\frac{\omega_{d-1}}{\omega_d} \sqrt{\frac{D}{d}} \right)^d, \quad (11)$$

where $\omega_d = \sqrt{\pi^d} / \Gamma(1 + d/2)$ is the content of the d -dimensional unit hypersphere and $\Gamma(\cdot)$ is Legendre's gamma function. Based on (11), we find that

$$H_{\rho_{\infty}, \epsilon}(\mathcal{F}_Z) \leq V(Z) \left\lceil \frac{1}{\epsilon} \right\rceil^d \leq \omega_d \left(\frac{\omega_{d-1}}{\omega_d} \left\lceil \frac{1}{\epsilon} \right\rceil \sqrt{\frac{D}{d}} \right)^d. \quad (12)$$

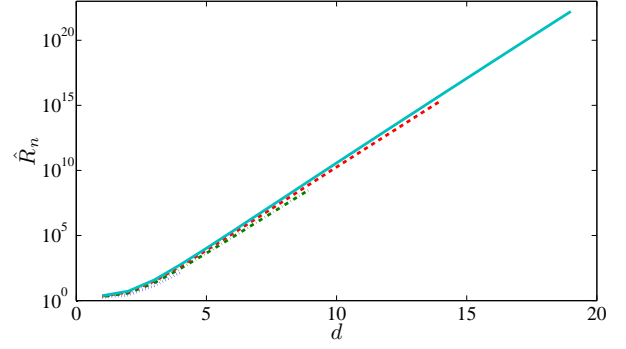


Fig. 2. Rademacher average as a function of the reduced dimension d for $D = 5$ (dotted blue line), $D = 10$ (dashed and dotted green line), $D = 15$ (dashed red line), and $D = 20$ (solid cyan line) for $\epsilon = 0.01$ and $n = 1000$.

For fixed reduced dimension d , $H_{\rho_{\infty}, \epsilon}(\mathcal{F}_Z)$ increases as a function of the measurement dimension D , i.e., the classifier function class is richer for larger measurement dimension with the same reduced-dimension. Importantly, $H_{\rho_{\infty}, \epsilon}(\mathcal{F}_Z)$ increases as a function of d for fixed D .

Substituting the $H_{\rho_{\infty}, \epsilon}(\mathcal{F}_Z)$ expression (12) into (8), we find that for a fixed measurement dimension D , the more the dimensionality is reduced, that is the smaller the value of d , the smaller the Rademacher complexity. This is shown in Fig. 2, a plot of the complexity value as a function of d for different values of D . Although larger measurement dimension D does result in larger complexity, the effect is minor in comparison to the effect of d .

Since training error increases as d decreases, and the generalization error is related to the sum of the Rademacher complexity and the training error: the more we reduce the dimension, the more we prevent overfitting. However, if we reduce the dimension too much, we end up underfitting the data; the training error component of the generalization error becomes large. There is an optimal reduced dimension that balances the training error and the complexity components of the generalization error.⁴

D. Consistency

With a training dataset of cardinality n drawn from $p_{\mathbf{X}, Y}(\mathbf{x}, y)$, a consistent classifier is one whose probability of error converges in the limit as n goes to infinity to the probability of error of the Bayes risk optimal decision rule \hat{y}^* when both types of classification errors have equal cost.⁵ For

⁴Note the purpose of generalization bounds in statistical learning theory as stated by Bousquet [65]: “one should not be concerned about the quantitative value of the bound or even about its fundamental form but rather about the terms that appear in the bound. In that respect a useful bound is one which allows to understand which quantities are involved in the learning process. As a result, performance bounds should be used for what they are good for. They should not be used to actually predict the value of the expected error. Indeed, they usually contain prohibitive constants or extra terms that are mostly mathematical artifacts. They should not be used directly as a criterion to optimize since their precise functional form may also be a mathematical artifact. However, they should be used to modify the design of the learning algorithms or to build new algorithms.”

⁵The Bayes optimal decision rule is a likelihood ratio test involving $p_{\mathbf{X}|Y}(\mathbf{x}|y = -1)$ and $p_{\mathbf{X}|Y}(\mathbf{x}|y = +1)$ with threshold equal to the ratio of the class prior probabilities.

consistency to be at all meaningful, we assume in this analysis that there is a reduced-dimensional statistic $\mathbf{A}^T \mathbf{x}$ so that the optimal Bayes decision rule based on this statistic achieves the same performance as the optimal decision rule based on the complete data \mathbf{x} , that is, we assume that there exists at least one $\mathbf{A}^* \in \mathcal{V}(D, d)$ such that $\Pr[Y \neq \hat{y}^*(\mathbf{A}^{*T} \mathbf{X})] = \Pr[Y \neq \hat{y}^*(\mathbf{X})]$, where \hat{y}^* takes the appropriate dimensional argument, and d is known. We also assume that the optimization method used in training finds the global optimum. The question is whether for a sequence of classifiers learned from training data $\hat{y}^{(n)}(\mathbf{x}) = \text{sign}(\varphi^{(n)}(\mathbf{A}^{(n)T} \mathbf{x}))$, where

$$(\mathbf{A}^{(n)}, \varphi^{(n)}) = \arg \min_{\mathbf{A} \in \mathcal{V}(D, d)} \min_{\varphi \in \mathcal{F}_Z(\mathbf{A})} \frac{1}{n} \sum_{j=1}^n \ell(y_j \varphi(\mathbf{A}^T \mathbf{x}_j)),$$

does $\Pr[Y \neq \hat{y}^{(n)}] - \Pr[Y \neq \hat{y}^*]$ converge in probability to zero. Note that $\Pr[Y \neq \hat{y}^{(n)}]$ is a random variable that depends on the data.

The properties of $\Pr[Y \neq \hat{y}^{(n)}]$ are affected by both the margin-based loss function ℓ and by the classifier function space \mathcal{F}_Z . Conditions on the loss function necessary for a margin-based classifier to be consistent are given in [13], [14], [66]. A loss function that meets the necessary conditions is termed *Fisher consistent* in [13]. Common margin-based loss functions including the logistic loss and hinge loss are Fisher consistent.⁶ Fisher consistency of the loss function is not enough, however, to imply consistency of the classifier overall; the function class must also be analyzed.

We apply Theorem 4.1 of [13], which is in turn an application of Theorem 1 of [67] to show consistency. The theorem is based on $H_{\rho_{\infty}, \epsilon}(\mathcal{F}_Z)$. In order to apply this theorem, we need to note three things. First, that ℓ is a Fisher consistent loss function. Second, that signed distance functions on Z are bounded in the L_{∞} norm. Third, that there exists a constant $B > 0$ such that $H_{\rho_{\infty}, \epsilon}(\mathcal{F}_Z) \leq B\epsilon^{-d}$, which follows from (12). Then, from [13] we have that⁷

$$\Pr[Y \neq \hat{y}^{(n)}] - \Pr[Y \neq \hat{y}^*] = O_P(n^{-\tau}), \quad (13)$$

where

$$\tau = \begin{cases} \frac{1}{3}, & d = 1 \\ \frac{1}{4} - \frac{\log \log n}{2 \log n}, & d = 2 \\ \frac{1}{2d}, & d \geq 3 \end{cases}$$

The dimensionality reduction and classification method is consistent: $\Pr[Y \neq \hat{y}^{(n)}] - \Pr[Y \neq \hat{y}^*]$ goes to zero as n goes to infinity because $n^{-\tau}$ goes to zero.

III. DIMENSIONALITY REDUCTION IN TREE-STRUCTURED NETWORKS

As discussed in Section I, a classification paradigm that intelligently reduces the dimensionality of measurements locally at sensors before transmitting them is critical in sensor network

settings. In this section, we make use of and appropriately extend the formulation of joint linear dimensionality reduction and classification presented in Section II for this task. For ease of exposition, we begin the discussion by first considering a setup with a single sensor, and then come to the general setting with m sensors networked according to a tree graph with a fusion center at the root of the tree. Also for simplicity of exposition, we assume that the fusion center does not take measurements, that it is not also a sensor; this assumption is by no means necessary. We make the assumption, as in [32]–[34], that the class labels y_j of the training set are available at the fusion center.

A. Network with Fusion Center and Single Sensor

Consider a network with a single sensor and a fusion center. The sensor measures data vector $\mathbf{x} \in \mathbb{R}^D$ and reduces its dimensionality using \mathbf{A} . The sensor transmits $\tilde{\mathbf{x}}_{s \rightarrow \text{fc}} = \mathbf{A}^T \mathbf{x} \in \mathbb{R}^d$ to the fusion center, which applies decision rule $\text{sign}(\varphi(\tilde{\mathbf{x}}_{s \rightarrow \text{fc}}))$ to obtain a classification for \mathbf{x} . Clearly in its operational phase, the linear dimensionality reduction reduces the amount of transmission required from the sensor to the fusion center.

Moreover, the communication required in training depends on the reduced dimension d rather than the dimension of the measurements D . The coordinate descent procedure described in Section II-B is naturally implemented in this distributed setting. With \mathbf{A} fixed, the optimization for φ occurs at the fusion center. The information needed by the fusion center to perform the optimization for φ are the $\tilde{\mathbf{x}}_{s \rightarrow \text{fc}, j}$, the dimensionality-reduced training examples. With φ fixed, the optimization for \mathbf{A} occurs at the sensor. Looking at (4), we see that the information required by the sensor from the fusion center to optimize \mathbf{A} includes only the scalar value $y_j \ell'(y_j \varphi(\tilde{\mathbf{x}}_{s \rightarrow \text{fc}, j}))$ and the column vector $[\varphi_1(\tilde{\mathbf{x}}_{s \rightarrow \text{fc}, j}) \cdots \varphi_d(\tilde{\mathbf{x}}_{s \rightarrow \text{fc}, j})]^T$, which we denote $\tilde{\varphi}'_{\text{fc} \rightarrow s, j} \in \mathbb{R}^d$, for $j = 1, \dots, n$.

Thus the alternating minimizations of the coordinate descent are accompanied by the alternating communication of messages $\tilde{\mathbf{x}}_{s \rightarrow \text{fc}, j}$ and $\tilde{\varphi}'_{\text{fc} \rightarrow s, j}$. The more computationally demanding optimization for φ (the application of a margin-based classification algorithm) takes place at the fusion center. A computationally simple Stiefel manifold gradient update occurs at the sensor.⁸ One may ask whether it is more efficient to perform training by just transmitting the full-dimensional measurements to the fusion center. The total communication involved in that case is $D(n + d)$ scalar values, whereas with the distributed implementation, this total is $(2d + 1)n$ multiplied by the number of coordinate descent iterations. Frequently D is much larger than d (an example in Section IV-B has $D = 10000$ and optimal $d = 20$), and the number of iterations is typically small (usually less than ten or twelve). In such cases, the distributed implementation provides quite a bit of savings. This scheme extends to the more interesting case of

⁶The conditions on ℓ for it to be Fisher consistent are mainly related to it being such that incorrect classifications incur more loss than correct classifications.

⁷The notation $\Psi_n = O_P(\psi_n)$ means that the random variable $\Psi_n = \psi_n \Xi_n$, where Ξ_n is a random variable bounded in probability [68]. Thus, if ψ_n converges to zero, then Ψ_n converges to zero in probability.

⁸The Stiefel manifold constraint requires QR factorization or other orthonormalization which may be prohibitive on certain existing sensor nodes, but as is demonstrated in [69] and references therein, efficient FPGA implementations of QR factorization have been developed and could be integrated into existing or new sensor nodes.

multisensor networks, as we describe next. The transmission savings of training with distributed implementation are further magnified in the multisensor network case.

B. Multisensor Networks

We now consider networks with m sensors connected in a tree topology with the fusion center at the root. We denote the χ_{fc} children of the fusion center as $\text{child}_1(fc), \dots, \text{child}_{\chi_{fc}}(fc)$; we also denote the χ_i children of sensor i as $\text{child}_1(i), \dots, \text{child}_{\chi_i}(i)$, and we denote the parent of sensor i as $\text{parent}(i)$. Training data vector $\mathbf{x}_{i,j} \in \mathbb{R}^{D_i}$ is measured by sensor i .⁹ The sensor receives dimensionality-reduced measurements from its children, combines them with its own measurements, and transmits a dimensionality-reduced version of this combination to its parent. Mathematically, the transmission from sensor i to its parent is:

$$\tilde{\mathbf{x}}_{i \rightarrow \text{parent}(i),j} = \mathbf{A}_i^T \begin{bmatrix} \mathbf{x}_{i,j} \\ \tilde{\mathbf{x}}_{\text{child}_1(i) \rightarrow i,j} \\ \vdots \\ \tilde{\mathbf{x}}_{\text{child}_{\chi_i}(i) \rightarrow i,j} \end{bmatrix}, \quad (14)$$

where $\mathbf{A}_i \in \mathcal{V}(D_i + \sum_{k=1}^{\chi_i} d_{\text{child}_k(i)}, d_i)$.

As an extension to the margin-based classification and linear dimensionality reduction objective (2), we propose the following objective for sensor networks:

$$\begin{aligned} L(\varphi, \mathbf{A}_1, \dots, \mathbf{A}_m) = \\ \sum_{j=1}^n \ell \left(y_j \varphi \left(\begin{bmatrix} \tilde{\mathbf{x}}_{\text{child}_1(fc) \rightarrow fc,j} \\ \vdots \\ \tilde{\mathbf{x}}_{\text{child}_{\chi_{fc}}(fc) \rightarrow fc,j} \end{bmatrix} \right) \right) + \lambda J(\varphi). \end{aligned} \quad (15)$$

Just as in the single sensor network in which the fusion center needed to receive the message $\tilde{\mathbf{x}}_{s \rightarrow fc,j}$ from its child in order to optimize φ , in the multisensor network the fusion center needs to receive the messages $\tilde{\mathbf{x}}_{\text{child}_1(fc) \rightarrow fc,j}, \dots, \tilde{\mathbf{x}}_{\text{child}_{\chi_{fc}}(fc) \rightarrow fc,j}$ from all of its children in order to optimize φ . The messages coming from the children of the fusion center are themselves simple linear functions of the messages coming from their children, as given in (14). The same holds down the tree to the leaf sensors. Thus, to gather the information required by the fusion center to optimize φ , a message-passing sweep occurs from the leaf nodes in the tree up to the root.

For fixed φ and optimization of the \mathbf{A}_i , we also see message-passing, this time sweeping back from the fusion center toward the leaves that generalizes what occurs in the single sensor network. Before finding the partial derivative of $L(\varphi, \mathbf{A}_1, \dots, \mathbf{A}_m)$ with respect to \mathbf{A}_i , let us first introduce further notation. We slice \mathbf{A}_i into blocks as follows:

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{A}_{i,\text{self}} \\ \mathbf{A}_{i,\text{child}_1} \\ \vdots \\ \mathbf{A}_{i,\text{child}_{\chi_i}} \end{bmatrix},$$

⁹In real-world situations, there is no reason to expect underlying likelihood functions for different sensors $p_{\mathbf{x}_i|Y}$, $i = 1, \dots, m$ to be identical. Different sensors will certainly be in different locations and may even be measuring different modalities of different dimensions with different amounts of noise.

where $\mathbf{A}_{i,\text{self}} \in \mathbb{R}^{D_i \times d_i}$ and $\mathbf{A}_{i,\text{child}_k} \in \mathbb{R}^{d_{\text{child}_k(i)} \times d_i}$. Also,

$$\tilde{\varphi}'_{fc \rightarrow \text{child}_k(fc),j} = \begin{bmatrix} \varphi_{\sum_{\kappa=1}^{k-1} d_{\text{child}_{\kappa}(fc)} + 1} \left(\begin{bmatrix} \tilde{\mathbf{x}}_{\text{child}_1(fc) \rightarrow fc,j} \\ \vdots \\ \tilde{\mathbf{x}}_{\text{child}_{\chi_{fc}}(fc) \rightarrow fc,j} \end{bmatrix} \right) \\ \vdots \\ \varphi_{\sum_{\kappa=1}^k d_{\text{child}_{\kappa}(fc)}} \left(\begin{bmatrix} \tilde{\mathbf{x}}_{\text{child}_1(fc) \rightarrow fc,j} \\ \vdots \\ \tilde{\mathbf{x}}_{\text{child}_{\chi_{fc}}(fc) \rightarrow fc,j} \end{bmatrix} \right) \end{bmatrix}$$

is the slice of the decision function gradient corresponding to the dimensions transmitted by $\text{child}_k(fc)$ to the fusion center. Additionally, let:

$$\tilde{\varphi}'_{i \rightarrow \text{child}_k(i),j} = \mathbf{A}_{i,\text{child}_k} \tilde{\varphi}'_{\text{parent}(i) \rightarrow i,j}. \quad (16)$$

Then, the matrix partial derivative of the objective function (15) with respect to \mathbf{A}_i is:

$$\begin{aligned} \mathbf{L}_{\mathbf{A}_i} = \sum_{j=1}^n y_j \ell' \left(y_j \varphi \left(\begin{bmatrix} \tilde{\mathbf{x}}_{\text{child}_1(fc) \rightarrow fc,j} \\ \vdots \\ \tilde{\mathbf{x}}_{\text{child}_{\chi_{fc}}(fc) \rightarrow fc,j} \end{bmatrix} \right) \right) \\ \times \begin{bmatrix} \mathbf{x}_{i,j} \\ \tilde{\mathbf{x}}_{\text{child}_1(i) \rightarrow i,j} \\ \vdots \\ \tilde{\mathbf{x}}_{\text{child}_{\chi_i}(i) \rightarrow i,j} \end{bmatrix} \tilde{\varphi}'_{\text{parent}(i) \rightarrow i,j}^T. \end{aligned} \quad (17)$$

Like in the single sensor network, the information required at sensor i to optimize \mathbf{A}_i that it does not already have consists of a scalar and a vector. The scalar value $y_j \ell'(y_j \varphi)$ is common throughout the network. The vector message $\tilde{\varphi}'_{\text{parent}(i) \rightarrow i,j}$ has length d_i and is received from $\text{parent}(i)$. As seen in (16), the message a sensor passes onto its child is a simple linear function of the message received from its parent. To optimize all of the \mathbf{A}_i , a message-passing sweep starting from the fusion center and going down to the leaves is required. Simple gradient descent along Stiefel manifold geodesics is then performed locally at each sensor. Overall, the coordinate descent training proceeds along with the passing of messages $\tilde{\mathbf{x}}_{i \rightarrow \text{parent}(i),j}$ and $\tilde{\varphi}'_{i \rightarrow \text{child}_k(i),j}$, which are functions of incoming messages as seen in (14) and (16).

C. Consistency and Complexity

The data vector that is received by the fusion center is reduced from $\sum_{i=1}^m D_i$ dimensions to $\sum_{k=1}^{\chi_{fc}} d_{\text{child}_k(fc)}$ dimensions. The fact that the composition of linear dimensionality reduction by two matrices on the Stiefel manifold can be represented by a single matrix on the Stiefel manifold leads to the observation that the dimensionality reduction performed by the sensor network has an equivalent matrix $\mathbf{A} \in \mathcal{V}(\sum_{i=1}^m D_i, \sum_{k=1}^{\chi_{fc}} d_{\text{child}_k(fc)})$. However, \mathbf{A} has further constraints than just the Stiefel manifold constraint due to the topology of the network. For example, the equivalent \mathbf{A} of the network in which the fusion center has two child sensors must be block-diagonal with two blocks.

Thus in the tree-structured sensor network, there is an equivalent matrix $\mathbf{A} \in \mathcal{T}(\sum_{i=1}^m D_i, \sum_{k=1}^{\chi_{fc}} d_{\text{child}_k(\text{fc})}) \subset \mathcal{V}(\sum_{i=1}^m D_i, \sum_{k=1}^{\chi_{fc}} d_{\text{child}_k(\text{fc})})$, where \mathcal{T} is a subset determined by the tree topology. The consistency analysis of Section II-D holds under the assumption that there exists an $\mathbf{A}^* \in \mathcal{T}(\sum_{i=1}^m D_i, \sum_{k=1}^{\chi_{fc}} d_{\text{child}_k(\text{fc})})$ such that $\Pr[Y \neq \hat{y}^*(\mathbf{A}^{*T} \mathbf{X})] = \Pr[Y \neq \hat{y}^*(\mathbf{X})]$.

The constrained set of dimensionality reduction matrices \mathcal{T} may have a smaller maximum zonotope content $V(Z)$ than the full Stiefel manifold, which would in turn mean a smaller Rademacher complexity. The fusion center receives the χ_{fc} -ary Cartesian product of dimensionality-reduced data from its children. The content of the Cartesian product is the product of the individual contents, and thus:

$$V(Z) \leq \prod_{k=1}^{\chi_{fc}} \omega_{d_{\text{child}_k(\text{fc})}} \left(\frac{\omega_{d_{\text{child}_k(\text{fc})}} - 1}{\omega_{d_{\text{child}_k(\text{fc})}}} \sqrt{\frac{D_k}{d_{\text{child}_k(\text{fc})}}} \right)^{d_{\text{child}_k(\text{fc})}},$$

which is less than or equal to the bound (11) for $\mathcal{Z}(\sum_{i=1}^m D_i, \sum_{k=1}^{\chi_{fc}} d_{\text{child}_k(\text{fc})})$. A more refined upper bound may be developed based on the specifics of the tree topology.

The tree-structured network has smaller Rademacher complexity than a dimensionality-reduced margin-based classifier of the same overall dimensions due to further constraints to the classifier function space resulting from the network structure. However, similar to D having a minor effect on complexity seen in Fig. 2, this smaller complexity for $\mathcal{T}(\sum_{i=1}^m D_i, \sum_{k=1}^{\chi_{fc}} d_{\text{child}_k(\text{fc})})$ is not much less than the complexity for the system without network constraints $\mathcal{V}(\sum_{i=1}^m D_i, \sum_{k=1}^{\chi_{fc}} d_{\text{child}_k(\text{fc})})$. The network constraints, however, may increase the training error. The generalization error expression (7), being composed of both the training error and the complexity, increases with network constraints due to increases in training error that are not offset by decreases in complexity, resulting in worse classification performance. However, for sensor networks, the performance criterion of interest is generally a combination of generalization error and power expenditure in communication.

D. Wireless Sensor Network Physical Model

Thus far in the section, we describe linear dimensionality reduction for margin-based classification in sensor networks abstractly, without considering the physical implementation or specific tree topologies. Here we set forth a specific physical model for wireless sensor networks that is used in Section IV-C. Consider m sensors and a fusion center in the plane that communicate wirelessly. The distance between sensor i and its parent is $r_{i \leftrightarrow \text{parent}(i)}$, and the power required for communication from i to its parent is $d_i r_{i \leftrightarrow \text{parent}(i)}^2$, where as before, d_i is the reduced dimension output by the sensor. The model arises by the common assumption of signal attenuation according to the square of the distance [70].¹⁰ The total transmission power used by the network is then:

$$\text{transmission power} = \sum_{i=1}^m d_i r_{i \leftrightarrow \text{parent}(i)}^2. \quad (18)$$

¹⁰The model $r_{i \leftrightarrow \text{parent}(i)}^\alpha$ for values of α other than two could also be considered.

We consider three network structures: parallel architecture, serial or tandem architecture, and binary tree architecture. In the parallel architecture, all m sensors are direct children of the fusion center. In the serial architecture, the fusion center has a single child, which in turn has a single child, and so on. In the binary tree architecture, the fusion center has two children, each of whom have two children on down the tree. When the number of sensors is such that a perfect binary tree is not produced, i.e., $m+2$ is not a power of two, the bottom level of the tree remains partially filled.

The sensor and fusion center locations are modeled as follows. The fusion center is fixed at the center of a circle with unit area and the m sensor locations are uniformly distributed over that circle. Given the sensor node locations and desired network topology, we assume that parent-child links and corresponding $r_{i \leftrightarrow \text{parent}(i)}$ are chosen to minimize (18). In a parallel network, the links are fixed with the fusion center as the parent of all sensors, and thus there is no parent-child link optimization to be performed. Exact minimization of (18) for the other architectures may not be tractable in deployed ad hoc wireless sensor networks because it involves solving a version of the traveling salesman problem for the serial architecture and a version of the minimum spanning tree problem for the binary tree architecture. Nevertheless, we assume that the minimization has been performed; we comment on this assumption later in the paper. For the parallel architecture, the distances are [71]:

$$r_{i \leftrightarrow \text{fc}}^{(\text{parallel})} = \frac{\Gamma(i + \frac{1}{2}) \Gamma(m+1)}{\sqrt{\pi} \Gamma(i) \Gamma(m + \frac{3}{2})}, \quad (19)$$

where sensor i is the i th closest sensor to the fusion center. There is no closed form expression for the $r_{i \leftrightarrow \text{parent}(i)}$ in the serial or binary tree architectures, but we estimate it through Monte Carlo simulation.

To fully specify the network, we must also set the reduced dimensions of the sensors d_i . The choice we make is to set d_i proportional to the number of descendants of sensor i plus one for itself. This choice implies that all d_i are equal in the parallel network, and that d_i is proportional to $m - i + 1$ in the serial network so that the number of dimensions passed up the chain to the fusion center increases the closer one gets to the fusion center. We will see that with this choice of d_i , all three topologies have essentially the same classification performance. This is not, however, generally true for different d_i assignments; for example, if we take all d_i to be equal in the serial network, the classification performance is quite poor. The imbalance in d_i values among different nodes is a shortcoming of our approach because nodes closer to the fusion center consume energy more quickly; future work may consider adapting aggregation services with balanced d_i [72], which have been used for distributed PCA, to our problem formulation.

IV. EXAMPLES AND RESULTS

With high-dimensional data, dimensionality reduction aids in visualization and human interpretation, allows the identification of important data components, and reduces the

computational and memory requirements of further analysis. An illustrative example is presented in this section, which shows the proposed dimensionality reduction and margin-based classification method. The key motivation of dimensionality reduction is that it prevents overfitting, which is shown in this section on several datasets.

Also in this section, we consider wireless sensor networks and look at classification performance as a function of transmission power expended. The phenomenon of overfitting seen in the centralized case has an important counterpart and implication for wireless sensor networks: increasing the total allowed transmission power—manifested either by increases in the number of sensors or increases in the number of transmitted dimensions per sensor—does not necessarily result in improved classification performance. The examples in this section illustrate several tradeoffs and suggest further lines of research.

A. Illustrative Example

We now present an illustrative example showing the operation of the classification–linear dimensionality reduction coordinate descent for training from a synthetic dataset. The dataset contains $n = 1000$ measurement vectors, of which 502 have label $y_j = -1$ and 498 have label $y_j = +1$. The dimensionality of the measurements is $D = 8$. The first two dimensions of the data, x_1 and x_2 , are informative for classification and the remaining six are completely uninformative. In particular, an ellipse in the x_1 – x_2 plane separates the two classes as shown in Fig. 3(a). The values in the other six dimensions are independent samples from an identical Gaussian distribution without regard for class label. Linear dimensionality reduction to $d = 2$ dimensions is sought. Note that the two class-conditional distributions have the same mean and are not Gaussians, and thus not very amenable to FDA. Fig. 4 shows the \mathbf{A} matrices obtained using PCA and FDA, visualized using the zonotope $Z(\mathbf{A})$. Neither PCA nor FDA is successful at recovering the informative subspace: the x_1 – x_2 plane.

We run our coordinate descent minimization of (2) to find both an \mathbf{A} matrix and decision boundary using two different margin-based classifiers: the SVM with radial basis function kernel and the geometric level set classifier with the logistic loss function. The matrix \mathbf{A} is randomly initialized. At convergence, the optimization procedure ought to give an \mathbf{A} matrix with all zeroes in the bottom six rows, corresponding to a zonotope that is a possibly rotated square, and an elliptical decision boundary. Fig. 3(b) shows the decision boundary resulting from the first optimization for φ using the GLS classifier with the random initialization for \mathbf{A} , before the first gradient descent step on the Stiefel manifold. Fig. 3(c)–(e) show intermediate iterations and Fig. 3(f) shows the final learned classifier and linear dimensionality reduction matrix. As the coordinate descent progresses, the zonotope becomes more like a square, i.e., \mathbf{A} aligns with the x_1 – x_2 plane, and the decision boundary becomes more like an ellipse. Fig. 5 shows the operation of the coordinate descent with the SVM. Here also, the zonotope becomes more like a square and the

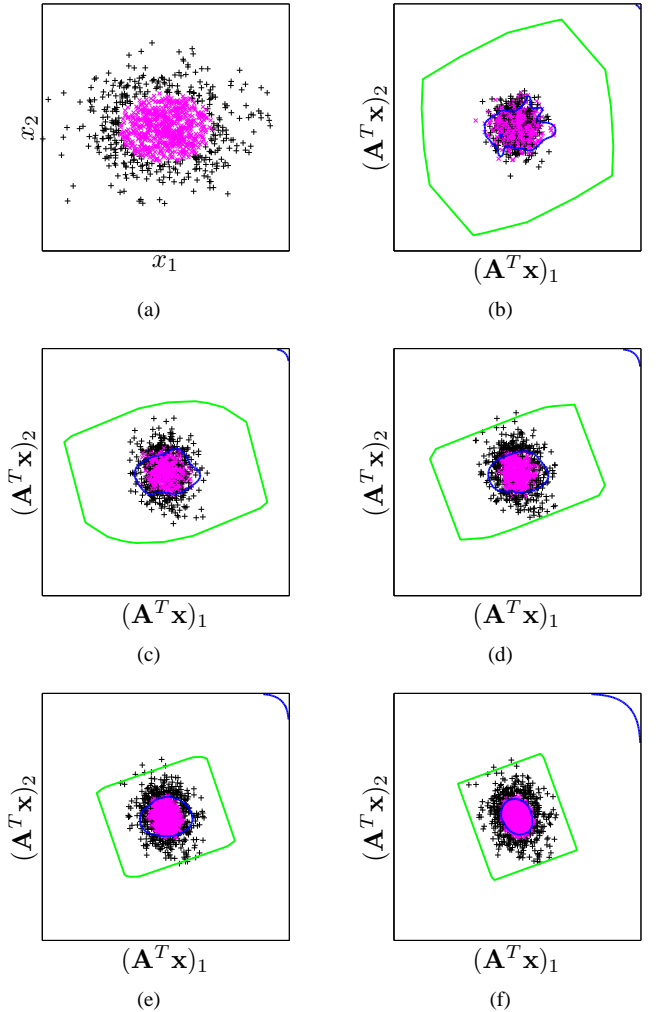


Fig. 3. Illustrative example. Magenta \times markers indicate label -1 . Black $+$ markers indicate label $+1$. The blue line is the classifier decision boundary. The green line outlines a zonotope generated by \mathbf{A}^T . (a) The first two measurement dimensions. (b) Random initialization for \mathbf{A} and first φ from GLS classifier. (c)–(e) Intermediate iterations. (f) Final \mathbf{A} and φ from GLS classifier.

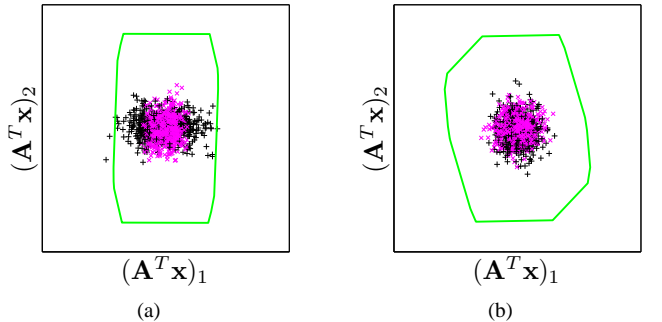


Fig. 4. Illustrative example. Magenta \times markers indicate label -1 . Black $+$ markers indicate label $+1$. The green line outlines a zonotope generated by \mathbf{A}^T from (a) PCA, and (b) FDA.

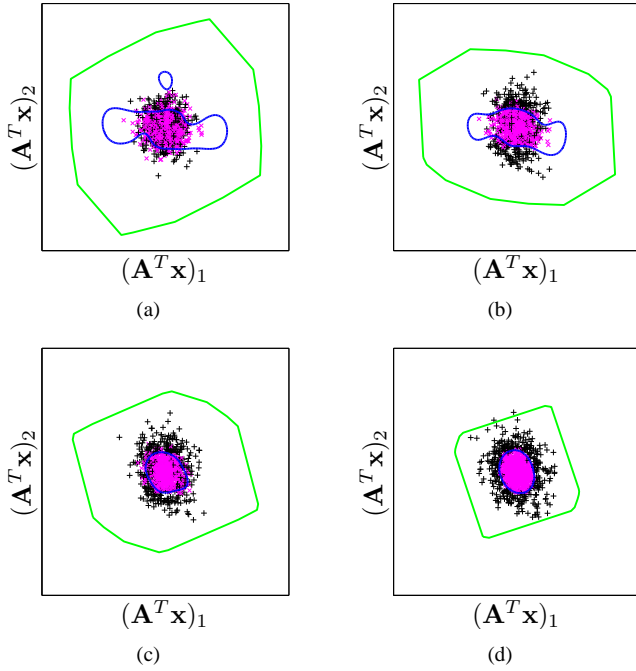


Fig. 5. Illustrative example. Magenta \times markers indicate label -1 . Black $+$ markers indicate label $+1$. The blue line is the classifier decision boundary. The green line outlines a zonotope generated by \mathbf{A}^T . (a) Random initialization for \mathbf{A} and first φ from SVM. (b)–(c) Intermediate iterations. (d) Final \mathbf{A} and φ from SVM.

TABLE I
INITIAL AND FINAL \mathbf{A} MATRICES IN ILLUSTRATIVE EXAMPLE

Random Initialization	GLS Solution	SVM Solution
$\begin{bmatrix} 0.0274 & -0.4639 \\ 0.4275 & 0.2572 \\ 0.4848 & 0.1231 \\ -0.0644 & 0.4170 \\ 0.0138 & 0.3373 \\ 0.5523 & 0.2793 \\ 0.1333 & 0.0283 \\ 0.5043 & -0.5805 \end{bmatrix}$	$\begin{bmatrix} 0.3386 & -0.9355 \\ 0.9401 & 0.3406 \\ 0.0118 & -0.0110 \\ 0.0103 & -0.0196 \\ 0.0246 & -0.0675 \\ -0.0172 & 0.0181 \\ 0.0186 & -0.0580 \\ -0.0108 & -0.0027 \end{bmatrix}$	$\begin{bmatrix} 0.3155 & -0.9425 \\ 0.9446 & 0.3098 \\ 0.0334 & 0.0936 \\ 0.0037 & 0.0356 \\ 0.0061 & -0.0318 \\ -0.0716 & 0.0121 \\ -0.0411 & -0.0410 \\ -0.0151 & -0.0537 \end{bmatrix}$

decision boundary becomes more like an ellipse throughout the minimization.

The random initial \mathbf{A} matrix and the final \mathbf{A} matrix solutions for the GLS classifier and the SVM are given in Table I. What we would want for this example is that the correct two-dimensional projection is identified and, assuming that it is, that the decision boundary is essentially elliptical. First, note that if the correct projection is identified, we expect the last six rows of the final \mathbf{A} matrix to be small compared to the first two rows and the corresponding zonotopes to be nearly square. Since rotations and reflections of the space onto which we project are inconsequential, we do not necessarily expect the first two rows of \mathbf{A} to be the identity matrix, nor do we expect the orientation of the nearly square zonotopes in Fig. 3(f) and Fig. 5(d) to line up with the coordinate axes. The results shown in Fig. 3(f), Fig. 5(d), and Table I reflect these desired characteristics. Given these final projections, we

see that the resulting decision boundaries are indeed nearly elliptical.¹¹ As this example indicates, the procedure is capable of making large changes to \mathbf{A} .

B. Classification Error For Different Reduced Dimensions

We present experimental classification results in this section on several datasets from the UCI machine learning repository [73]. The joint linear dimensionality reduction and margin-based classification method proposed in Section II is run for different values of the reduced dimension d , showing that performing dimensionality reduction does in fact improve classification performance in comparison to not performing dimensionality reduction. The margin-based classifier that is used is the SVM with radial basis function kernel and default parameter settings from the Matlab bioinformatics toolbox.

First, we look at training error and test error¹² as a function of the reduced dimension on five different datasets from varied application domains: Wisconsin diagnostic breast cancer ($D = 30$), ionosphere ($D = 34$), sonar ($D = 60$), arrhythmia ($D = 274$ after preprocessing to remove dimensions containing missing values), and arcene ($D = 10000$). On the first four datasets, we look at the tenfold cross-validation training and test errors. The arcene dataset has separate training and validation sets which we employ for these purposes.

The tenfold cross-validation training error is shown with blue triangle markers and the tenfold cross-validation test error is shown with red circle markers for the ionosphere dataset in Fig. 6(a). The plot also contains error bars showing one standard deviation above and below the average error over the ten folds. In Fig. 6(b), the test error for the joint minimization is compared to the test error if the linear dimensionality reduction is first performed using PCA, FDA, information preserving component analysis [44], or sufficient dimension reduction (structured principal fitted components [74]), followed by classification with the kernel SVM. Fig. 7 shows tenfold cross-validation training and test error for other datasets. Fig. 8 gives the training and test performance for the arcene dataset. For the Wisconsin diagnostic breast cancer, ionosphere, and sonar datasets, we show classification performance for all possible reduced dimensions. For the arrhythmia and arcene datasets, we show reduced dimensions up to $d = 50$ and $d = 100$, respectively.

The first thing to notice in the plots is that the training error quickly converges to zero with an increase in the reduced dimension d . The margin-based classifier with linear dimensionality reduction perfectly separates the training set when the reduced dimension is sufficiently large. However, this perfect separation does not carry over to the test error—the error in which we are most interested. In all of the datasets, the test error first decreases as we increase the reduced dimension, but then starts increasing. There is an intermediate optimal value

¹¹The curved piece of the decision boundary in the top right corner of the domain in Fig. 3(f) is an artifact of geometric level sets and does not affect classification performance.

¹²Training error is the misclassification associated with the data used to learn the Stiefel manifold matrix and decision function. Test error is the misclassification associated with data samples that were not used in training and is a surrogate for generalization error.

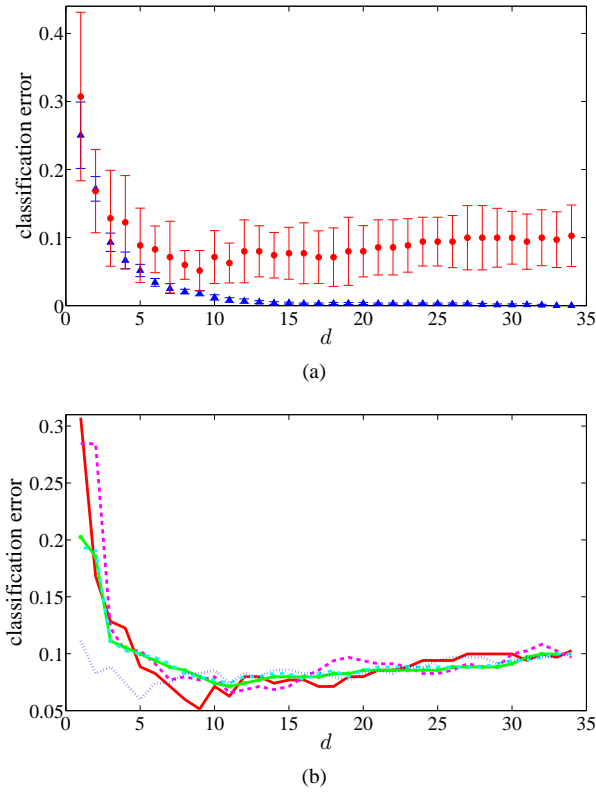


Fig. 6. (a) Tenfold cross-validation training error (blue triangle markers) and test error (red circle markers) on ionosphere dataset. Error bars indicate standard deviation over the ten folds. (b) Tenfold cross-validation test error on ionosphere dataset using PCA (dashed and dotted cyan line), FDA (dashed magenta line), information preserving component analysis (dotted blue line), sufficient dimension reduction (green line with markers), and joint minimization (solid red line). Error bars are not included because they would make the plot unreadable, but note that standard deviations for all five methods are approximately the same.

for the reduced dimension. For the five datasets, these values are $d = 3$, $d = 9$, $d = 16$, $d = 10$, and $d = 20$, respectively. This test error behavior is evidence of overfitting if d is too large. Dimensionality reduction improves classification performance on unseen samples by preventing overfitting. Remarkably, even the ten thousand-dimensional measurements in the arcene dataset can be linearly reduced to twenty dimensions. In the ionosphere dataset test error comparison plot, it can be seen that the minimum test error is smaller with the joint minimization than when doing dimensionality reduction separately with PCA, FDA, information preserving component analysis, or sufficient dimension reduction. Moreover, this minimum test error occurs at a smaller reduced dimensionality than the minima for PCA, FDA, and sufficient dimension reduction. Comparisons on other datasets are similar.

The classification error as a function of d using our new joint linear dimensionality reduction and margin-based classification method matches the structural risk minimization principle. Rademacher complexity analysis supporting these empirical findings is presented in Section II-C.

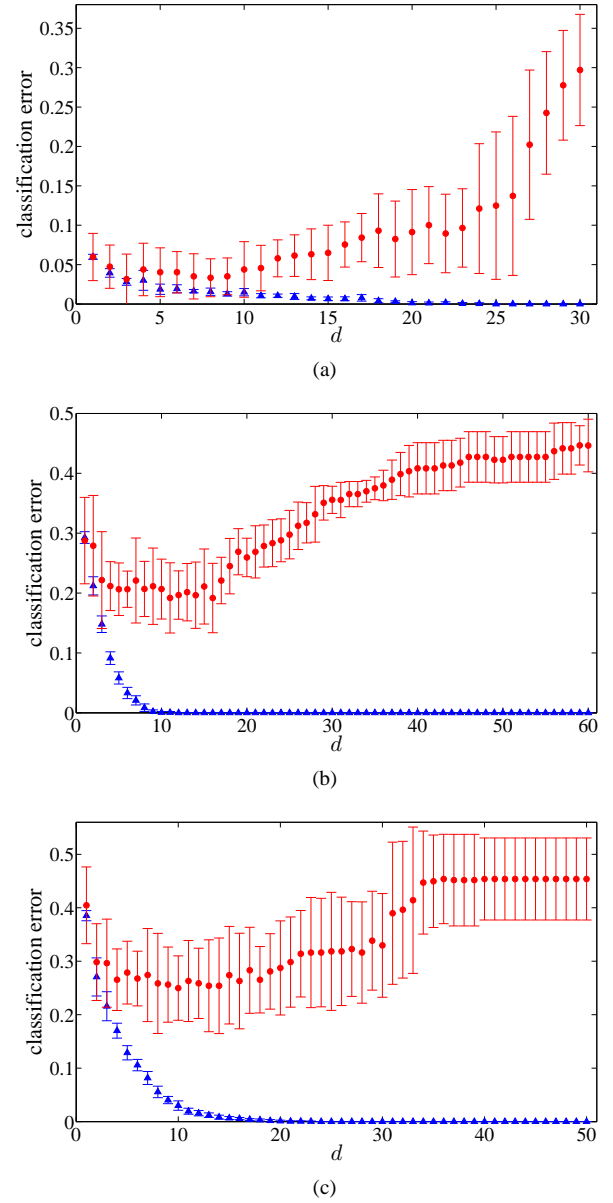


Fig. 7. Tenfold cross-validation training error (blue triangle markers) and test error (red circle markers) on (a) Wisconsin diagnostic breast cancer, (b) sonar, and (c) arrhythmia datasets. Error bars indicate standard deviation over the ten folds.

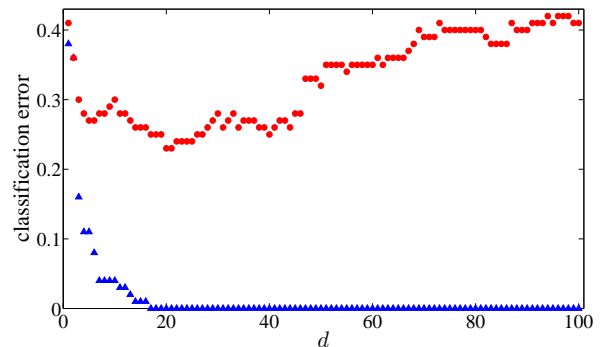


Fig. 8. Training error (blue triangle markers) and test error (red circle markers) on arcene dataset.

C. Classification Error For Different Networks

Given the sensor network model of Section III-D, we look at classification performance for the three different network architectures with different amounts of transmission power. Different transmission powers are obtained by varying the number of sensors and scaling the d_i values. We emulate data coming from a sensor network by slicing the dimensions of the ionosphere, sonar, and arcene datasets and assigning the different dimensions to different sensors. With $D_i = 5$ for all sensors in the network for the ionosphere and sonar datasets and $D_i = 50$ for the arcene dataset, we assign the dimensions in the order given in the UCI machine learning repository, so the first sensor ‘measures’ the first D_i dimensions listed, the second sensor ‘measures’ dimensions $D_i + 1$ through $2D_i$, and so on. The dimensions are not ordered according to relevance for classification in any way.

We plot results for the ionosphere dataset in Fig. 9. In Fig. 9(a), we plot tenfold cross-validation training and test error obtained from the algorithm described in Section III-B with the parallel network as a function of transmission power. Each training and test error pair corresponds to a different value of $m = 1, 2, \dots, 6$ and $d_i = 1, 2, \dots, 5$. In Section IV-B, we plotted classification performance as a function of the reduced dimension, but here the horizontal axis is transmission power, taking the distance between sensor nodes into account. As in Section IV-B, the phenomenon of overfitting is quite apparent.

In Fig. 9(b), classification error is plotted as a function of transmission power for the serial architecture. The points in the plot are for different numbers of sensors $m = 1, 2, \dots, 6$ and different scalings of the reduced dimension $d_i = (m - i + 1), 2(m - i + 1), \dots, 5(m - i + 1)$. The classification error values in Fig. 9(b) are quite similar to the ones for the parallel case.¹³ The plot for the parallel architecture appearing to be a horizontally compressed version of the serial architecture plot indicates that to achieve those similar classification performances, more transmission power is required by the serial architecture. Although the distances between parents and children tends to be smaller in the serial architecture, the chosen d_i are larger closer to the fusion center leading to higher transmission power.

The binary tree architecture’s classification error plot is given in Fig. 9(c). The training and test error values are similar to the other two architectures.¹⁴ The transmission power needed to achieve the given classification errors is similar to that of the parallel architecture and less than the serial architecture. Among the three architectures with the d_i assigned as described in Section III-D, all have approximately the same classification performance, but the serial network uses more power.

The same experiments are repeated for the sonar and arcene datasets with plots given in Fig. 10 and Fig. 11. For the sonar dataset, m varies from one to eleven, and d_i of leaf nodes from

¹³In fact, they are the same for the five pairs of points when $m = 1$ because the parallel and serial networks are the same when there is a single sensor.

¹⁴The binary tree is the same as the parallel network for $m = 1, 2$ and the serial network for $m = 1$.

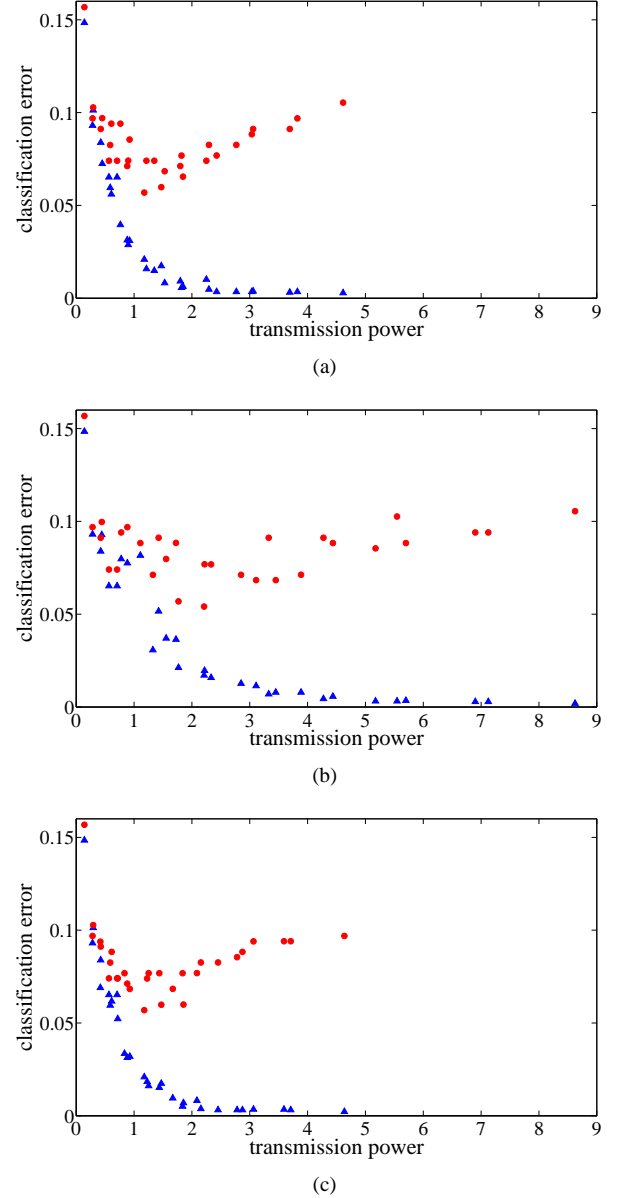
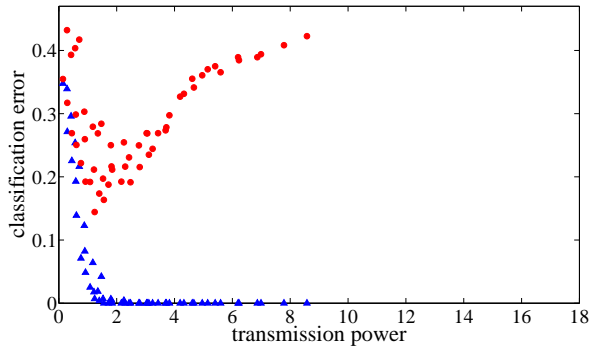


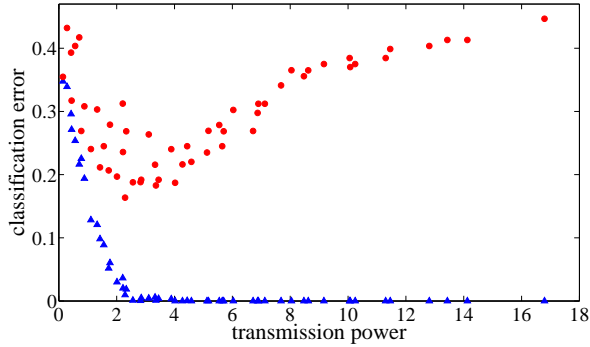
Fig. 9. Tenfold cross-validation training error (blue triangle markers) and test error (red circle markers) on ionosphere dataset for (a) parallel, (b) serial, and (c) binary tree network architectures.

one to five. For the arcene dataset, m varies from one to ten, and d_i of leaf nodes from one to fifteen. The same trends can be observed as in the ionosphere dataset; similar plots are produced for other datasets such as Wisconsin diagnostic breast cancer and arrhythmia. All three network topologies produce similar classification errors, but the serial network uses more power.

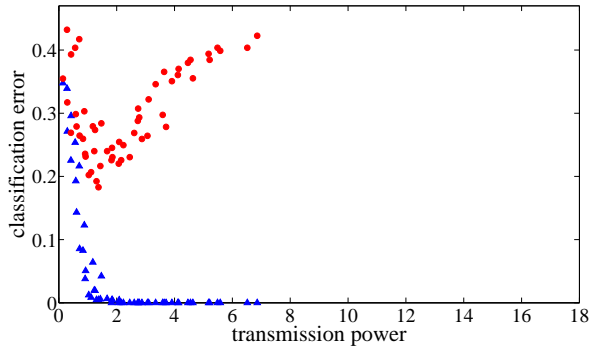
Some overall observations for wireless sensor networks are the following. There exist some optimal parameters of the network with a finite number of sensors and some dimensionality reduction. One may be tempted to think that deploying more sensors always helps classification performance since the total number of measured dimensions increases, but we find that this is not generally true. For a fixed number of samples n , once



(a)



(b)

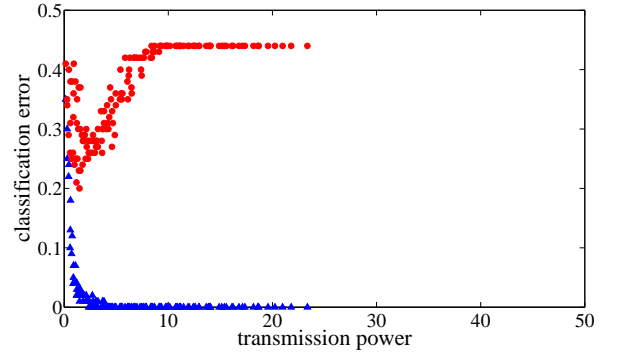


(c)

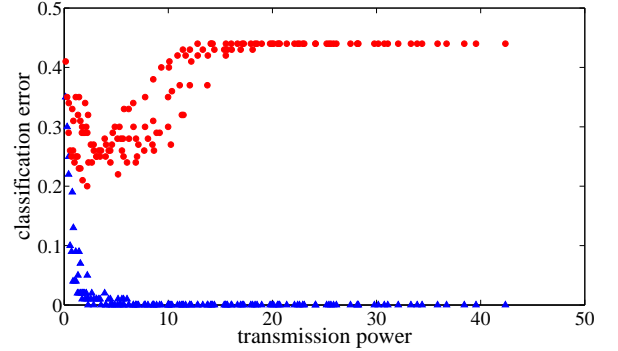
Fig. 10. Tenfold cross-validation training error (blue triangle markers) and test error (red circle markers) on sonar dataset for (a) parallel, (b) serial, and (c) binary tree network architectures.

there are enough sensors to fit the data, adding more sensors leads to overfitting and a degradation of test performance. That a small number of sensors, which perform dimensionality reduction, yield optimal classification performance is good from the perspective of resource usage. Among different possible choices of network architectures, we have compared three particular choices. Others are certainly possible, including the investigated topologies but with different d_i proportions. For the chosen d_i proportions, all three network topologies have essentially the same classification performance, but this is not true for other choices.

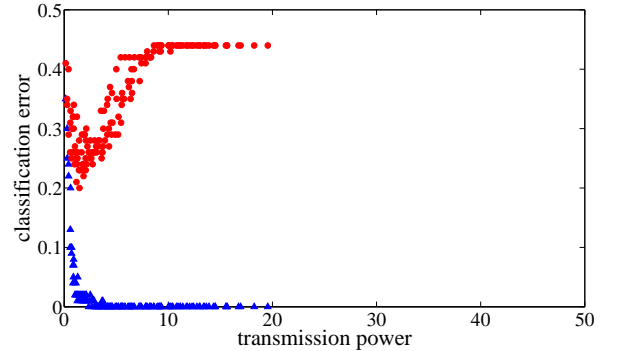
In this empirical investigation of classification performance versus resource usage, the main observation is that the two are not at odds. The decrease of resource usage is coin-



(a)



(b)



(c)

Fig. 11. Training error (blue triangle markers) and test error (red circle markers) on arcene dataset for (a) parallel, (b) serial, and (c) binary tree network architectures.

cident with the prevention of overfitting, which leads to improved classification performance. Oftentimes there is a tradeoff between resource usage and performance, but that is not the case in the overfitting regime. Additionally, among the network architectures compared, the parallel and binary tree architectures use less power in communication than the serial architecture for equivalent classification performance. The plotted transmission power values, however, are based on choosing the parent-child links to exactly minimize (18); in practice, this minimization will only be approximate for the binary tree architecture and will require a certain amount of communication overhead. Therefore, the parallel architecture, which requires no optimization, is recommended for this application. This new distributed dimensionality reduc-

tion formulation and empirical study suggests a direction for future research, namely the problem of finding the number of sensors, the network structure, and the set of d_i that optimize generalization error in classification for a given transmission power budget and given number of training samples n .

D. Spatially-Distributed Sensor Node Data

As a confirmation of the results given for emulated sensor network data in Section IV-C, here we present results on two datasets arising from spatially-distributed sensor nodes. The first dataset is based on sensor measurements collected at the Intel Berkeley Research Laboratory in 2004. The second dataset is based on sensor measurements collected at the Army Research Laboratory in 2007 [75].

The Intel Berkeley dataset as available contains temperature, relative humidity, and light measurements for 54 sensors over more than a month. A classification task is required for the methodology developed in this paper, and thus we define two classes based on the light measurements, dark and bright. The dark class corresponds to the average light being less than 125 lx and the bright class to greater than 125 lx. Our formulation requires a correspondence among measurements from different sensors in order to define a single sample j ; the sensor measurements are time-stamped with an epoch number such that measurements from different sensors with the same epoch number correspond to the same time. However, each epoch number corresponds to much fewer than 54 sensors. Thus we take length 60 blocks of epoch numbers and consider all measurements within a block to correspond to the same time. We take the first reading if a block contains more than one reading from the same sensor. Even with this blocking, if we insist that a sample needs data from all 54 sensors, we obtain very few samples. Thus we only consider 12 sensors, numbered 1, 2, 3, 4, 6, 31, 32, 33, 34, 35, 36, and 37 in the dataset. With such processing, we obtain $n = 2346$ samples.

Spatial locations of the sensors are given. For the network structure, we consider a fusion center located in the center of the sensors and links between nodes according to the Euclidean minimum spanning tree with the fusion center at the root. We train on the first quarter of the samples containing temperature and relative humidity measurements and test on the latter three quarters of the samples, varying the number of sensors and the d_i scaling. The training and test errors as a function of total transmission power in the network is given in Fig. 12(a). As in previous results, we see the effects of overfitting. An intermediate transmission power level is optimal for classification performance even with spatially-distributed sensor node data.

The Army Research Laboratory data consists of sensor nodes that take four acoustic, three seismic, one electric field, and four passive infrared measurements. Measurements are taken during the dropping of a 14 pound steel cylinder from nine inches above the ground and during no significant human activity. The cylinder dropping happens at various spatial locations in relation to the sensors. In this dataset, we have 200 samples of cylinder dropping and 200 samples of no activity. We train on the first half of the samples and test on

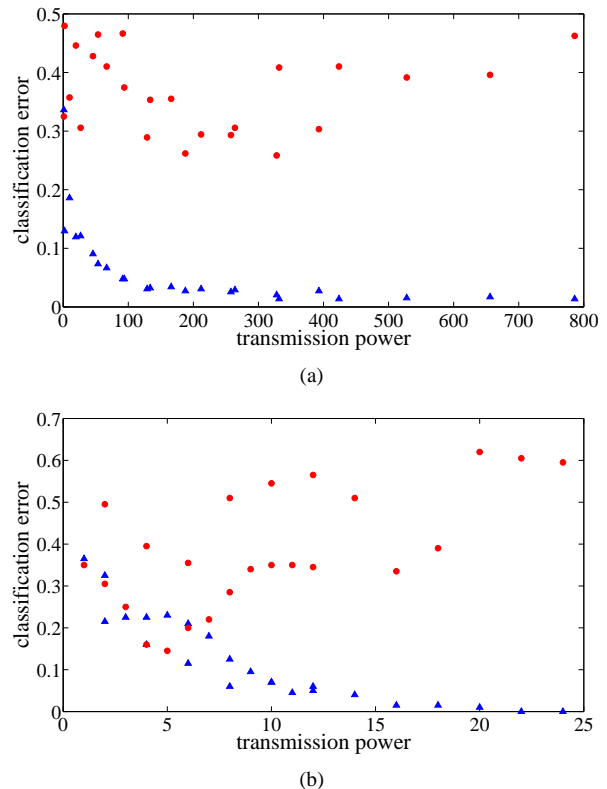


Fig. 12. Training error (blue triangle markers) and test error (red circle markers) on (a) Intel Berkeley dataset and (b) Army Research Laboratory dataset.

the remaining samples. The fusion center is again placed in the center of the sensors and a minimum spanning tree network is used. Training error and test error are plotted in Fig. 12(b) for different numbers of sensors and different d_i scalings. Again, we see that an intermediate level of transmission power is optimal for classification test error, with overfitting for large transmission powers.

V. CONCLUSION

In this paper, we have formulated linear dimensionality reduction driven by the objective of margin-based classification. We have developed an optimization approach that involves alternation between two minimizations: one to update a classifier decision function and the other to update a matrix on the Stiefel manifold. We have both analytically and empirically looked at the phenomenon of overfitting: analytically through the Rademacher complexity, and empirically through experiments on several real datasets, illustrating that dimensionality reduction is an important component in improving classification accuracy. We have also analytically characterized the consistency of the dimensionality-reduced classifier. We have described how our proposed optimization scheme can be distributed in a network containing a single sensor through a message-passing approach, with the classifier decision function updated at the fusion center and the dimensionality reduction matrix updated at the sensor. Additionally, we have extended the formulation to tree-structured fusion networks.

Papers such as [32], [34] have advocated nonparametric learning, of which margin-based classification is a subset, for inference in distributed settings such as wireless sensor networks. Reducing the amount of communication is an important consideration in these settings, which we have addressed in this paper through a joint linear dimensionality reduction and margin-based classification method applicable to networks in which sensors measure more than one variable. Reducing communication is often associated with a degradation in performance, but in this application it is not the case in the regime when dimensionality reduction prevents overfitting. Thus, dimensionality reduction is important for two distinct reasons: reducing the amount of resources consumed, and obtaining good generalization.

ACKNOWLEDGMENT

The authors thank J. H. G. Dauwels, J. W. Fisher III and S. R. Sanghavi for valuable discussions, P. Bodik, W. Hong, C. Guestrin, S. Madden, M. Paskin and R. Thibaux for collecting the Intel Berkeley data, T. Damarla, S. G. Iyengar and A. Subramanian for furnishing the Army Research Laboratory data, K. M. Carter, R. Raich and A. O. Hero III for information preserving component analysis software, and R. D. Cook, L. Forzani and D. Tomassi for sufficient dimension reduction software.

REFERENCES

- [1] K. R. Varshney and A. S. Willsky, "Learning dimensionality-reduced classifiers for information fusion," in *Proc. Int. Conf. Inf. Fusion*, Seattle, WA, Jul. 2009, pp. 1881–1888.
- [2] K. R. Varshney, "Frugal hypothesis testing and classification," Ph.D. Thesis, Mass. Inst. Technol., Cambridge, MA, 2010.
- [3] M. Çetin, L. Chen, J. W. Fisher, III, A. T. Ihler, R. L. Moses, M. J. Wainwright, and A. S. Willsky, "Distributed fusion in sensor networks," *IEEE Signal Process. Mag.*, vol. 23, no. 4, pp. 42–55, Jul. 2006.
- [4] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh, "Fidelity and yield in a volcano monitoring sensor network," in *Proc. USENIX Symp. Operating Syst. Des. Implement.*, Seattle, WA, Nov. 2006, pp. 381–396.
- [5] L. Zong, J. Houser, and T. R. Damarla, "Multi-modal unattended ground sensor (MMUGS)," in *Proc. SPIE*, vol. 6231, Apr. 2006, p. 623118.
- [6] Z. Zhu and T. S. Huang, *Multimodal Surveillance: Sensors, Algorithms, and Systems*. Boston: Artech House, 2007.
- [7] H. L. Van Trees, *Detection, Estimation, and Modulation Theory*. New York: Wiley, 1968.
- [8] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer, 1995.
- [9] L. Zwald, R. Vert, G. Blanchard, and P. Massart, "Kernel projection machine: A new tool for pattern recognition," in *Adv. Neural Inf. Process. Syst. 17*. Cambridge, MA: MIT Press, 2005, pp. 1649–1656.
- [10] S. Mosci, L. Rosasco, and A. Verri, "Dimensionality reduction and generalization," in *Proc. Int. Conf. Mach. Learn.*, Corvallis, OR, Jun. 2007, pp. 657–664.
- [11] G. Blanchard and L. Zwald, "Finite-dimensional projection for classification and statistical learning," *IEEE Trans. Inf. Theory*, vol. 54, no. 9, pp. 4169–4182, Sep. 2008.
- [12] A. Srivastava and X. Liu, "Tools for application-driven linear dimension reduction," *Neurocomputing*, vol. 67, pp. 136–160, Aug. 2005.
- [13] Y. Lin, "A note on margin-based loss functions in classification," *Stat. Probabil. Lett.*, vol. 68, no. 1, pp. 73–82, Jun. 2004.
- [14] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe, "Convexity, classification, and risk bounds," *J. Am. Stat. Assoc.*, vol. 101, no. 473, pp. 138–156, Mar. 2006.
- [15] K. R. Varshney and A. S. Willsky, "Classification using geometric level sets," *J. Mach. Learn. Res.*, vol. 11, pp. 491–516, Feb. 2010.
- [16] A. M. Martínez and A. C. Kak, "PCA versus LDA," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 2, pp. 228–233, Feb. 2001.
- [17] J. Bi, K. P. Bennett, M. Embrechts, C. M. Breneman, and M. Song, "Dimensionality reduction via sparse support vector machines," *J. Mach. Learn. Res.*, vol. 3, pp. 1229–1243, Mar. 2003.
- [18] B. Krishnapuram, A. J. Hartemink, L. Carin, and M. A. T. Figueiredo, "A Bayesian approach to joint feature selection and classifier design," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1105–1111, Sep. 2004.
- [19] L. Wolf and A. Shashua, "Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach," *J. Mach. Learn. Res.*, vol. 6, pp. 1855–1887, Nov. 2005.
- [20] V. Y. F. Tan, S. Sanghavi, J. W. Fisher, III, and A. S. Willsky, "Learning graphical models for hypothesis testing and classification," *IEEE Trans. Signal Process.*, vol. 58, no. 11, pp. 5481–5495, Nov. 2010.
- [21] K. Huang and S. Aviyente, "Sparse representation for signal classification," in *Adv. Neural Inf. Process. Syst. 19*. Cambridge, MA: MIT Press, 2007, pp. 609–616.
- [22] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Discriminative learned dictionaries for local image analysis," in *Proc. IEEE CS Conf. Comput. Vis. Pattern Recogn.*, Anchorage, AK, 2008.
- [23] D. M. Blei and J. D. McAuliffe, "Supervised topic models," in *Adv. Neural Inf. Process. Syst. 20*. Cambridge, MA: MIT Press, 2008, pp. 121–128.
- [24] S. Lacoste-Julien, F. Sha, and M. I. Jordan, "DiscLDA: Discriminative learning for dimensionality reduction and classification," in *Adv. Neural Inf. Process. Syst. 21*. Cambridge, MA: MIT Press, 2009, pp. 897–904.
- [25] S. Lazebnik and M. Raginsky, "Supervised learning of quantizer codebooks by information loss minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 7, pp. 1294–1309, Jul. 2009.
- [26] V. Koltchinskii, "Rademacher penalties and structural risk minimization," *IEEE Trans. Inf. Theory*, vol. 47, no. 5, pp. 1902–1914, Jul. 2001.
- [27] P. L. Bartlett and S. Mendelson, "Rademacher and Gaussian complexities: Risk bounds and structural results," *J. Mach. Learn. Res.*, vol. 3, pp. 463–482, Nov. 2002.
- [28] R. R. Tenney and N. R. Sandell, Jr., "Detection with distributed sensors," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-17, no. 4, pp. 501–510, Jul. 1981.
- [29] J. N. Tsitsiklis, "Decentralized detection," Lab. Inf. Decision Syst., Mass. Inst. Technol., Tech. Rep. P-1913, Sep. 1989.
- [30] P. K. Varshney, *Distributed Detection and Data Fusion*. New York: Springer-Verlag, 1996.
- [31] J.-F. Chamberland and V. V. Veeravalli, "Decentralized detection in sensor networks," *IEEE Trans. Signal Process.*, vol. 51, no. 2, pp. 407–416, Feb. 2003.
- [32] X. Nguyen, M. J. Wainwright, and M. I. Jordan, "Nonparametric decentralized detection using kernel methods," *IEEE Trans. Signal Process.*, vol. 53, no. 11, pp. 4053–4066, Nov. 2005.
- [33] 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. 52–63, Jan. 2006.
- [34] —, "Distributed learning in wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 23, no. 4, pp. 56–69, Jul. 2006.
- [35] M. Gastpar, P. L. Dragotti, and M. Vetterli, "The distributed Karhunen-Loève transform," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5177–5196, Dec. 2006.
- [36] I. D. Schizas, G. B. Giannakis, and Z.-Q. Luo, "Distributed estimation using reduced-dimensionality sensor observations," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4284–4299, Aug. 2007.
- [37] O. Roy and M. Vetterli, "Dimensionality reduction for distributed estimation in the infinite dimensional regime," *IEEE Trans. Inf. Theory*, vol. 54, no. 4, pp. 1655–1669, Apr. 2008.
- [38] E. A. Patrick and F. P. Fischer, II, "Nonparametric feature selection," *IEEE Trans. Inf. Theory*, vol. IT-15, no. 5, pp. 577–584, Sep. 1969.
- [39] R. Lotlikar and R. Kothari, "Adaptive linear dimensionality reduction for classification," *Pattern Recognition*, vol. 33, no. 2, pp. 185–194, Feb. 2000.
- [40] J. C. Principe, D. Xu, and J. W. Fisher, III, "Information-theoretic learning," in *Unsupervised Adaptive Filtering*, S. Haykin, Ed. New York: Wiley, 2000, vol. 1, pp. 265–320.
- [41] K. Torkkola, "Feature extraction by non-parametric mutual information maximization," *J. Mach. Learn. Res.*, vol. 3, pp. 1415–1438, Mar. 2003.
- [42] Z. Nenadic, "Information discriminant analysis: Feature extraction with an information-theoretic objective," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 8, pp. 1394–1407, Aug. 2007.
- [43] M. Thangavelu and R. Raich, "Multiclass linear dimension reduction via a generalized Chernoff bound," in *Proc. IEEE Workshop Mach. Learn. Signal Process.*, Cancún, Mexico, Oct. 2008, pp. 350–355.

- [44] K. M. Carter, R. Raich, and A. O. Hero, III, "An information geometric approach to supervised dimensionality reduction," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Taipei, Taiwan, Apr. 2009.
- [45] K.-C. Li, "Sliced inverse regression for dimension reduction," *J. Am. Stat. Assoc.*, vol. 86, no. 414, pp. 316–327, Jun. 1991.
- [46] —, "On principal Hessian directions for data visualization and dimension reduction: Another application of Stein's lemma," *J. Am. Stat. Assoc.*, vol. 87, no. 420, pp. 1025–1039, Dec. 1992.
- [47] F. Chiaromonte and R. D. Cook, "Sufficient dimension reduction and graphics in regression," *Ann. Inst. Stat. Math.*, vol. 54, no. 4, pp. 768–795, Dec. 2002.
- [48] K. Fukumizu, F. R. Bach, and M. I. Jordan, "Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces," *J. Mach. Learn. Res.*, vol. 5, pp. 73–99, Jan. 2004.
- [49] Sajama and A. Orlitsky, "Supervised dimensionality reduction using mixture models," in *Proc. Int. Conf. Mach. Learn.*, Bonn, Germany, Aug. 2005, pp. 768–775.
- [50] K. Fukumizu, F. R. Bach, and M. I. Jordan, "Kernel dimension reduction in regression," *Ann. Stat.*, vol. 37, no. 4, pp. 1871–1905, Aug. 2009.
- [51] X. Liu, A. Srivastava, and K. Gallivan, "Optimal linear representations of images for object recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 5, pp. 662–666, May 2004.
- [52] F. Pereira and G. Gordon, "The support vector decomposition machine," in *Proc. Int. Conf. Mach. Learn.*, Pittsburgh, PA, Jun. 2006, pp. 689–696.
- [53] D.-S. Pham and S. Venkatesh, "Robust learning of discriminative projection for multicategory classification on the Stiefel manifold," in *Proc. IEEE CS Conf. Comput. Vis. Pattern Recogn.*, Anchorage, AK, Jun. 2008.
- [54] I. W.-H. Tsang, A. Kocsor, and J. T.-Y. Kwok, "Large-scale maximum margin discriminant analysis using core vector machines," *IEEE Trans. Neural Netw.*, vol. 19, no. 4, pp. 610–624, Apr. 2008.
- [55] S. Ji and J. Ye, "Linear dimensionality reduction for multi-label classification," in *Proc. Int. Joint Conf. Artificial Intell.*, Pasadena, CA, Jul. 2009, pp. 1077–1082.
- [56] S. Ji, L. Tang, S. Yu, and J. Ye, "Extracting shared subspace for multi-label classification," in *Proc. ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining*, Las Vegas, NV, Aug. 2008, pp. 381–389.
- [57] M. Rousson and N. Paragios, "Prior knowledge, level set representations & visual grouping," *Int. J. Comput. Vis.*, vol. 76, no. 3, pp. 231–243, 2008.
- [58] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM J. Matrix Anal. A.*, vol. 20, no. 2, pp. 303–353, Jan. 1998.
- [59] J. H. Manton, "Optimization algorithms exploiting unitary constraints," *IEEE Trans. Signal Process.*, vol. 50, no. 3, pp. 635–650, Mar. 2002.
- [60] Y. Nishimori and S. Akaho, "Learning algorithms utilizing quasi-geodesic flows on the Stiefel manifold," *Neurocomputing*, vol. 67, pp. 106–135, Aug. 2005.
- [61] U. von Luxburg and O. Bousquet, "Distance-based classification with Lipschitz functions," *J. Mach. Learn. Res.*, vol. 5, pp. 669–695, Jun. 2004.
- [62] A. N. Kolmogorov and V. M. Tihomirov, " ϵ -entropy and ϵ -capacity of sets in functional spaces," *Am. Math. Soc. Translations Series 2*, vol. 17, pp. 277–364, 1961.
- [63] G. D. Chakerian and P. Filliman, "The measures of the projections of a cube," *Studia Scientiarum Mathematicarum Hungarica*, vol. 21, no. 1–2, pp. 103–110, 1986.
- [64] P. Filliman, "Extremum problems for zonotopes," *Geometriae Dedicata*, vol. 27, no. 3, pp. 251–262, Sep. 1988.
- [65] O. Bousquet, "New approaches to statistical learning theory," *Ann. Inst. Statist. Math.*, vol. 55, no. 2, pp. 371–389, Jun. 2003.
- [66] I. Steinwart, "Consistency of support vector machines and other regularized kernel classifiers," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 128–142, Jan. 2005.
- [67] X. Shen and W. H. Wong, "Convergence rate of sieve estimates," *Ann. Stat.*, vol. 22, no. 2, pp. 580–615, Jun. 1994.
- [68] A. W. van der Vaart, *Asymptotic Statistics*. Cambridge, UK: Cambridge University Press, 1998.
- [69] X. Wang and M. Leeser, "A truly two-dimensional systolic array FPGA implementation of QR decomposition," *ACM Trans. Embed. Comput. Syst.*, vol. 9, no. 1, Oct. 2009.
- [70] J. C. Maxwell, *A Treatise on Electricity and Magnetism*. Oxford, UK: Clarendon Press, 1873.
- [71] P. Bhattacharyya and B. K. Chakrabarti, "The mean distance to the n th neighbour in a uniform distribution of random points: An application of probability theory," *Eur. J. Phys.*, vol. 29, no. 3, pp. 639–645, May 2008.
- [72] Y.-A. Le Borgne, S. Raybaud, and G. Bontempi, "Distributed principal component analysis for wireless sensor networks," *Sensors*, vol. 8, no. 8, pp. 4821–4850, Aug. 2008.
- [73] A. Asuncion and D. J. Newman, "UCI machine learning repository," Available: <http://archive.ics.uci.edu/ml>, 2007.
- [74] R. D. Cook and L. Forzani, "Principal fitted components for dimension reduction in regression," *Statist. Sci.*, vol. 23, no. 4, pp. 485–501, Nov. 2008.
- [75] R. Damarla, M. Beigi, and A. Subramanian, "Human activity experiments performed at ARL," Tech. Rep., Apr. 2007.



Kush R. Varshney (S'00–M'10) was born in Syracuse, NY in 1982. He received the B.S. degree (magna cum laude) in electrical and computer engineering with honors from Cornell University, Ithaca, NY in 2004. He received the S.M. degree in 2006 and the Ph.D. degree in 2010, both in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge.

He is a research staff member in the Business Analytics and Mathematical Sciences Department at the IBM Thomas J. Watson Research Center, Yorktown Heights, NY. While at MIT, he was a research assistant with the Stochastic Systems Group in the Laboratory for Information and Decision Systems and a National Science Foundation Graduate Research Fellow. He has been a visiting student at École Centrale, Paris, and an intern at Lawrence Livermore National Laboratory, Sun Microsystems, and Sensis Corporation. His research interests include statistical signal processing, statistical learning, and image processing.

Dr. Varshney is a member of Eta Kappa Nu, Tau Beta Pi, IEEE, and ISIF. He received a best student paper travel award at the 2009 International Conference on Information Fusion.



Alan S. Willisky (S'70–M'73–SM'82–F'86) joined the Massachusetts Institute of Technology, Cambridge, in 1973 and is the Edwin Sibley Webster Professor of Electrical Engineering and Director of the Laboratory for Information and Decision Systems.

He was a founder of Alphatech, Inc. and Chief Scientific Consultant, a role in which he continues at BAE Systems Advanced Information Technologies. From 1998 to 2002 he served on the U.S. Air Force Scientific Advisory Board. He has received several awards including the 1975 American Automatic Control Council Donald P. Eckman Award, the 1979 ASCE Alfred Noble Prize, the 1980 IEEE Browder J. Thompson Memorial Award, the IEEE Control Systems Society Distinguished Member Award in 1988, the 2004 IEEE Donald G. Fink Prize Paper Award, Doctorat Honoris Causa from Université de Rennes in 2005, and the 2009 Technical Achievement Award from the IEEE Signal Processing Society. In 2010, he was elected to the National Academy of Engineering.

Dr. Willisky has delivered numerous keynote addresses and is coauthor of the text *Signals and Systems*. His research interests are in the development and application of advanced methods of estimation, machine learning, and statistical signal and image processing.