12th International Conference on Information Fusion
Seattle, WA, USA, July 6-9, 2009

Learning Dimensionality-Reduced Classifiers for Information Fusion

Kush R. Varshney and Alan S. Willsky
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA, USA
krv@mit.edu, willsky@mit.edu

Abstract – *The fusion of multimodal sensor information often requires learning decision rules from samples of high-dimensional data. Each data dimension may only be weakly informative for the detection problem of interest. Also, it is not known a priori which components combine to form a lower-dimensional feature space that is most informative. To learn both the combination of dimensions and the decision rule specified in the reduced-dimensional space together, we jointly optimize the linear dimensionality reduction and margin-based supervised classification problems, representing dimensionality reduction by matrices on the Stiefel manifold. We describe how the learning procedure and resulting decision rule can be implemented in parallel, serial, and tree-structured fusion networks.*

Keywords: supervised classification, linear dimensionality reduction, information fusion, Stiefel manifold, sensor network

1 Introduction

The journalist Malcolm Gladwell has written that "in good decision making, frugality matters." "Even the most complicated of relationships ... have an identifiable underlying pattern." "In picking up these sorts of patterns, less is more. Overloading the decision makers with information ... makes picking up that signature harder, not easier" [1]. The calculation of sufficient statistics, such as the likelihood ratio for binary detection and classification, is a way to losslessly reduce the dimensionality of high-dimensional sensor measurements before applying a decision rule defined in the reduced-dimensional space. Doing so, however, requires full knowledge of the probability distribution generating the measurements, which is often not available.

Situations in which the probability distribution is unknown, but samples from it are available, call for *supervised learning* [2]. For the most part, supervised learning methods produce decision rules defined in the full high-dimensional measurement space rather than in a reduced-dimensional space. In this paper, we develop a method for jointly learning the reduced-dimensional space and the decision rule defined in that reduced-dimensional space from samples. We focus on *linear* dimensionality reduction because of its simplicity and efficiency [3].

Motivated by applications such as wireless sensor networks, distributed decision making and information fusion have received much attention in recent times [4, 5]. Stemming from resource constraints, the distinguishing attribute of these applications is the limited transmission and computation capacity of sensors. The majority of work in distributed decision making has focused on the situation when probability distributions are known, but there has been some work on learning for distributed settings as well [6, 7, 8]. Constrained resources provide one motivation to be frugal in decision making, but a distinct fundamental reason to control complexity when learning from finite data is the structural risk minimization principle [9]. The model class from which the decision rule is selected should be restricted in order to improve generalization and avoid overfitting.

Most popular methods of linear dimensionality reduction, including principal component analysis (PCA) and independent component analysis, find linear transformations and can thus be posed as optimization problems on the Stiefel or Grassmann manifold with different objectives [3]. The Stiefel manifold is the set of all linear subspaces with basis specified, and the Grassman manifold is the set of all linear subspaces with the basis left unspecified. In many applications of interest, the objectives of the popular methods do not align with the ultimate task that the data is to be used for, and consequently are suboptimal with respect to the objective of interest. The problem of interest to us is statistical classification or detection. We propose an optimization problem on the Stiefel manifold whose objective is that of *margin-based classification* [10, 11], and develop an iterative coordinate descent algorithm for its solu-

tion. Several classification methods are margin-based, e.g. logistic regression, support vector machine (SVM), and the level set classifier of [12].

We show how the linear dimensionality reduction of heterogeneous data may be distributed in parallel, serial, or general tree-structured fusion networks with a fusion center via individual Stiefel manifold matrices at each sensor. We implement the coordinate descent learning algorithm in the fusion network through a message-passing approach. Dimensionality reduction aids the generalization of classifiers, especially in the presence of uninformative data dimensions, and reduces the amount of communication in the network.

Prior work has also considered the problem of linear dimensionality reduction for classification. In [13, 3], the classifier is a nearest neighbor classifier and the objective function includes validation data; optimization is through Markov chain Monte Carlo-type simulated annealing. In [14], a linear regression, rather than classification, objective and a regression parameter/Stiefel manifold coordinate descent algorithm analogous to ours is developed. Kernel dimensionality reduction is derived in [15], which builds on the idea that the low-dimensional features should be such that the conditional probability of the class labels given the high-dimensional data equals the conditional probability of the class labels given the low-dimensional features. Maximum margin discriminant analysis is a method for dimensionality reduction based on the SVM that finds the low-dimensional features one by one instead of all at once, and also does not simultaneously give a classifier [16]. The method that we propose has an explicit margin-based classification objective, finds all low-dimensional features in a joint manner, and gives both the dimensionality reduction mapping and the classifier as output. The prior work has not considered information fusion networks; however, the framework we develop may be applicable to these other methods as well.

The paper is organized as follows. In Section 2, we review the definition of and optimization methods on the Stiefel manifold from the perspective of linear dimensionality reduction. The section also describes margin-based classification. Section 3 combines the ideas of optimization on the Stiefel manifold and margin-based classification to give a joint linear dimensionality reduction and classification objective as well as an iterative algorithm. Section 4 shows how the basic method of Section 3 extends to fusion networks, and Section 5 concludes.

2 Preliminaries

2.1 The Stiefel Manifold

Linear dimensionality reduction from D dimensions to $d \leq D$ dimensions can be represented by a matrix $\mathbf{A} \in \mathbb{R}^{D\times d}$. With a data vector $\mathbf{x} \in \mathbb{R}^D$, $\mathbf{A}^T\mathbf{x}$ is in d dimensions. $\mathbf{A}$ is constrained to have orthonormal columns, i.e. $\mathbf{A}^T\mathbf{A} = \mathbf{I}$, so that there is no redundancy in representation and so that the scaling of the dimensions has no effect. This constrained set,

$$\mathcal{V}_{D,d} = \{\mathbf{A} \in \mathbb{R}^{D\times d}, d \leq D | \mathbf{A}^T\mathbf{A} = \mathbf{I}\},$$

is known as the Stiefel manifold.

Generally speaking, an optimization problem on the Stiefel manifold is solved to find the dimensionality reduction matrix:

$$\min E(\mathbf{A}) \quad \text{s.t. } \mathbf{A} \in \mathcal{V}_{D,d}, \tag{1}$$

where E is a scalar-valued function. Some special functions, including that for PCA, can be minimized through eigendecomposition. For differentiable functions, several iterative minimization algorithms exist [17, 18, 19].

We give the expression for gradient descent along geodesics of the Stiefel manifold [17]. Specifically, let $\mathbf{E_A}$ denote the $D \times d$ matrix with elements $\frac{\partial E}{\partial a_{ij}}$. The gradient is:

$$\mathbf{G} = \mathbf{E_A} - \mathbf{A}\mathbf{E_A}^T\mathbf{A}. \tag{2}$$

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

$$\mathbf{A}(\tau) = \mathbf{A}(0)\mathbf{M}(\tau) + \mathbf{Q}\mathbf{N}(\tau), \tag{3}$$

where $\mathbf{QR}$ 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.

2.2 Margin-Based Classification

In margin-based classification, we are given training data $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, with data vectors $\mathbf{x}_n \in \mathbb{R}^D$ and class labels $y_n \in \{-1, +1\}$, and would like to learn a classifier $\hat{y}(\cdot) = \text{sign}(\varphi(\cdot)) : \mathbb{R}^D \to \{-1, +1\}$. The decision function φ is chosen to minimize the energy functional:

$$E(\varphi) = \sum_{n=1}^{N} \mathrm{L}(y_n\varphi(\mathbf{x}_n)) + \lambda J(\varphi), \tag{4}$$

where L is a margin-based loss function [10, 11], for example the logistic loss function:

$$\mathrm{L}_{\text{logistic}}(z) = \log\left(1 + e^{-z}\right)$$

or the hinge loss function:

$$\mathrm{L}_{\text{hinge}}(z) = \max\{0, 1 - z\},$$

and the additional term J is used for regularization.

In the kernel SVM, L is the hinge loss, the decision functions φ are in a reproducing kernel Hilbert space, the regularization term is the squared norm in that space, and optimization may be carried out through quadratic programming techniques [2]. In the level set classifier of [12], any margin-based loss function may be used, the decision functions are in the space of signed distance functions, the regularization term is the surface area of the zero level set of φ, and optimization is carried out through curve evolution.

3 Linear Dimensionality Reduction for Classification

We formulate joint linear dimensionality reduction and classification by extending the energy functional (4) to also include a $D \times d$ matrix $\mathbf{A}$:

$$E(\varphi, \mathbf{A}) = \sum_{n=1}^{N} \mathrm{L}(y_n \varphi(\mathbf{A}^T \mathbf{x}_n)) + \lambda J(\varphi), \qquad (5)$$

with the constraint $\mathbf{A} \in \mathcal{V}_{D,d}$. The decision function φ is defined in the reduced d-dimensional space. An option for performing the minimization amenable to distributed implementation is coordinate descent, alternating minimizations with fixed $\mathbf{A}$ and with fixed φ. With $\mathbf{A}$ fixed, the minimization may be performed exactly in the same manner as to minimize (4), as discussed in Section 2.2.

With φ fixed, the general problem of minimizing a function $E(\mathbf{A})$ subject to $\mathbf{A}$ lying on the Stiefel manifold is encountered, as discussed in Section 2.1. The function $E(\mathbf{A}) = \sum_{n=1}^{N} \mathrm{L}(y_n \varphi(\mathbf{A}^T \mathbf{x}_n))$ is differentiable with respect to $\mathbf{A}$ for differentiable loss functions. The first derivative is:

$$\mathbf{E}_{\mathbf{A}} = \sum_{n=1}^{N} y_n \, \mathrm{L}'(y_n \varphi(\mathbf{A}^T \mathbf{x}_n)) \times \mathbf{x}_n \begin{bmatrix} \varphi_1(\mathbf{A}^T \mathbf{x}_n) & \cdots & \varphi_d(\mathbf{A}^T \mathbf{x}_n) \end{bmatrix}. \qquad (6)$$

Note that $\mathbf{x}_n$ is a $D \times 1$ vector and that $\begin{bmatrix} \varphi_1(\mathbf{A}^T \mathbf{x}_n) & \cdots & \varphi_d(\mathbf{A}^T \mathbf{x}_n) \end{bmatrix}$ is a $1 \times d$ vector, where $\varphi_i(\cdot)$ is the partial derivative of the decision function with respect to dimension i. For the logistic loss function:

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

and for the hinge loss function:

$$\mathrm{L}'_{\text{hinge}}(z) = -\operatorname{step}(1 - z).$$

The gradient descent along Stiefel manifold geodesics then involves applying equations (2) and (3) with the matrix derivative (6).

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 $N = 1000$ input data samples, of which 502 have label $y_n = -1$ and 498 have label $y_n = +1$. The input dimensionality 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. 1(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.

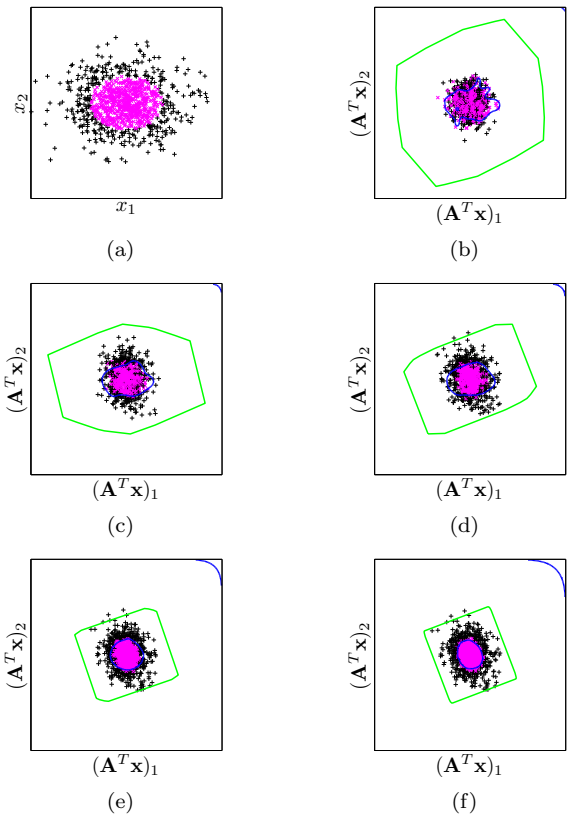

Figure 1: Illustrative example. Magenta × markers indicate label −1. Black + markers indicate label +1. The blue line is the decision boundary. The green line is the convex hull of the projection of an eight-dimensional hypercube to two dimensions by $\mathbf{A}$. (a) The first two input data dimensions. (b) Initial dimensionality reduction matrix and first decision boundary. (c)–(e) Intermediate iterations. (f) Final dimensionality reduction and decision boundary.

The matrix $\mathbf{A}$ is randomly initialized and the level set classifier of [12] with the logistic loss function is used. At convergence, the optimization procedure ought to give an $\mathbf{A}$ matrix with all zeroes in the bottom six rows and an elliptical decision boundary. In order to visualize the $\mathbf{A}$ matrix, we show the convex hull of the projection of a D-dimensional hypercube by $\mathbf{A}$ onto

Table 1: Initial and Final $\mathbf{A}$ in Illustrative Example

Initial		Final	
0.0274	−0.4639	0.3546	−0.9314
0.4275	0.2572	0.9343	0.3556
0.4848	0.1231	0.0158	−0.0002
−0.0644	0.4170	−0.0003	−0.0219
0.0138	0.3373	0.0157	−0.0609
0.5523	0.2793	−0.0192	0.0186
0.1333	0.0283	0.0194	−0.0365
0.5043	−0.5805	−0.0104	−0.0066

$d = 2$ dimensions. In general, this convex hull is a point-symmetric polygon with $2D$ sides; if $\mathbf{A}$ is aligned to one or more of the D dimensions, then the polygon has fewer sides. (The silhouette of a cube is a hexagon in general, but is a square when viewed straight on.) Fig. 1(b) shows the decision boundary resulting from the first optimization for φ with the random initialization for $\mathbf{A}$, before the first gradient descent step on the Stiefel manifold. Fig. 1(c)–(e) show intermediate iterations and Fig. 1(f) shows the final learned classifier and linear dimensionality reduction matrix. As the coordinate descent progresses, the convex hull of the hypercube projection 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.

The initial $\mathbf{A}$ matrix and the final $\mathbf{A}$ matrix are given in Table 1. Conforming to the expected behavior, the final decision boundary is almost an ellipse and the final $\mathbf{A}$ has very little energy in the bottom six rows. (The curved piece of the decision boundary in the top right corner of the domain is an artifact of level set classification and does not affect classification performance.) The classification is invariant to rotations and reflections in the x_1–x_2 plane, which is why we do not expect the identity matrix in the top two rows of the final $\mathbf{A}$. As this example indicates, the procedure is capable of making large changes to $\mathbf{A}$.

4 Information Fusion Networks

Several important decision making applications contain distributed multimodal sensors with limited data transmission capacity. A classification paradigm that intelligently reduces the dimensionality of measurements locally at sensors before transmitting them to a decision maker or fusion center is critical in these settings. Making use of our formulation of joint linear dimensionality reduction and classification for this task, first with a single remote sensor, the dimensionality reduction matrix $\mathbf{A}$ resides at the sensor and the decision function φ resides at the fusion center, as illustrated in Fig. 2(a). The sensor transmits $\mathbf{A}^T\mathbf{x} \in \mathbb{R}^d$ rather than the full measurements $\mathbf{x} \in \mathbb{R}^D$, thus saving on transmission costs.

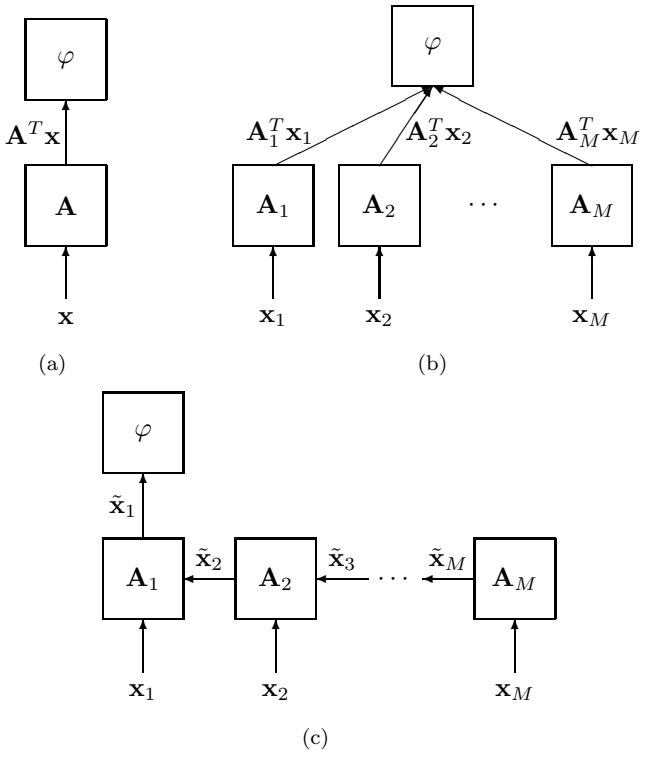

Figure 2: Fusion networks with (a) single sensor, (b) parallel sensors, and (c) serial sensors.

Our joint linear dimensionality reduction and classification formulation supports distributed implementation not only in the operational phase, but also in training. The coordinate descent procedure described in the previous section can be naturally implemented in distributed settings with communication costs related to d rather than D. Additionally, the optimization for φ (the application of a margin-based classification algorithm), which is much more computationally intensive than the optimization for $\mathbf{A}$, takes place at the fusion center. We make the assumption, as in [6, 7, 8], that the class labels y_n of the training set are available at the fusion center. The transmission-constrained sensor only needs to send $\mathbf{A}^T\mathbf{x}_n$ to the fusion center for it to be able to optimize for φ. For the sensor to be able to take a gradient step along a Stiefel manifold geodesic to update $\mathbf{A}$, the fusion center needs to send $y_n \, \mathrm{L}'(y_n \varphi(\mathbf{A}^T\mathbf{x}_n))$ and $\begin{bmatrix}\varphi_1(\mathbf{A}^T\mathbf{x}_n) & \cdots & \varphi_d(\mathbf{A}^T\mathbf{x}_n)\end{bmatrix}$.

4.1 Multisensor Fusion

The dimensionality reduction/classification framework extends to the more interesting *multisensor* information fusion network case. With M sensors, the training data consists of the class labels y_n and the data vectors $\mathbf{x}_{m,n} \in \mathbb{R}^{D_m}$ measured at sensor m for $m = 1, \ldots, M$. Each sensor has its own dimensionality reduction matrix on the Stiefel manifold $\mathbf{A}_m$. We consider parallel,

serial, and general tree-structured fusion networks.

In the parallel network, illustrated in Fig. 2(b), sensor m has matrix $\mathbf{A}_m \in \mathcal{V}_{D_m,d_m}$ and transmits $\mathbf{A}_m^T\mathbf{x}_{m,n}$ to the fusion center. The decision function $\varphi : \mathbb{R}^{\sum_{m=1}^M d_m} \to \mathbb{R}$ is applied to a stacked vector consisting of the reduced-dimensional data from each sensor. The margin-based classification objective for the parallel fusion network is:

$$E(\varphi, \mathbf{A}_1, \ldots, \mathbf{A}_M) = \sum_{n=1}^N \mathrm{L}\left(y_n\varphi\left(\begin{bmatrix}\mathbf{A}_1^T\mathbf{x}_{1,n}\\ \vdots \\ \mathbf{A}_M^T\mathbf{x}_{M,n}\end{bmatrix}\right)\right) + \lambda J(\varphi). \quad (7)$$

The optimization of $E(\varphi)$ at the fusion center for fixed $\mathbf{A}_m$ is a straightforward application of a margin-based classification algorithm. Sensor m only needs to send message $\mathbf{A}_m^T\mathbf{x}_{m,n}$ to the fusion center.

For the Stiefel manifold portion of the coordinate descent, we find the partial derivative of the objective function with respect to $\mathbf{A}_m$:

$$\mathbf{E}_{\mathbf{A}_m} = \sum_{n=1}^N y_n \mathrm{L}'\left(y_n\varphi\left(\begin{bmatrix}\mathbf{A}_1^T\mathbf{x}_{1,n}\\ \vdots \\ \mathbf{A}_M^T\mathbf{x}_{M,n}\end{bmatrix}\right)\right)\mathbf{x}_{m,n} \times \begin{bmatrix}\varphi_i\left(\begin{bmatrix}\mathbf{A}_1^T\mathbf{x}_{1,n}\\ \vdots \\ \mathbf{A}_M^T\mathbf{x}_{M,n}\end{bmatrix}\right) & \cdots & \varphi_j\left(\begin{bmatrix}\mathbf{A}_1^T\mathbf{x}_{1,n}\\ \vdots \\ \mathbf{A}_M^T\mathbf{x}_{M,n}\end{bmatrix}\right)\end{bmatrix}, \quad (8)$$

where $i = \sum_{\mu=1}^{m-1} d_\mu + 1$ and $j = \sum_{\mu=1}^{m} d_\mu$. Similar to the single sensor network, for sensor m to perform its gradient update of $\mathbf{A}_m$, the fusion center needs to send it a message containing the L' term and the subvector of the decision function gradient given in (8).

In a serial network, with sensors labeled such that sensor 1 is the child of the fusion center and sensor $m+1$ is the child of sensor m for $m = 1, \ldots, M-1$, we look at the case where the parent sensor fuses its own measured data with the dimensionality-reduced information received from the child sensor. Thus we have dimensionality reduction matrices

$$\mathbf{A}_m = \begin{bmatrix}\mathbf{A}_{m,\text{self}}\\ \mathbf{A}_{m,\text{child}}\end{bmatrix} \in \mathcal{V}_{D_m+d_{m+1},d_m}$$

for $m = 1, \ldots, M-1$ with $\mathbf{A}_{m,\text{self}} \in \mathbb{R}^{D_m\times d_m}$ and $\mathbf{A}_{m,\text{child}} \in \mathbb{R}^{d_{m+1}\times d_m}$, and $\mathbf{A}_M \in \mathcal{V}_{D_M,d_M}$. (The last sensor has no child.) Let $\tilde{\mathbf{x}}_{M,n} = \mathbf{A}_M^T\mathbf{x}_{M,n}$ and

$$\tilde{\mathbf{x}}_{m,n} = \mathbf{A}_m^T\begin{bmatrix}\mathbf{x}_{m,n}\\ \tilde{\mathbf{x}}_{m+1,n}\end{bmatrix} \quad (9)$$

for $m = 1, \ldots, M-1$. Sensor m transmits $\tilde{\mathbf{x}}_{m,n}$ to its parent, culminating in transmission of $\tilde{\mathbf{x}}_{1,n}$ to the fusion center. This network is illustrated in Fig. 2(c).

The margin-based classification objective for the serial network is:

$$E(\varphi, \mathbf{A}_1, \ldots, \mathbf{A}_M) = \sum_{n=1}^N \mathrm{L}\left(y_n\varphi\left(\tilde{\mathbf{x}}_{1,n}\right)\right) + \lambda J(\varphi). \quad (10)$$

We can optimize for φ at the fusion center for fixed $\mathbf{A}_m$ after a message-passing sweep of $\tilde{\mathbf{x}}_{m,n}$ starting from sensor M. Getting the required gradient information to the sensors requires a message-passing sweep in the opposite direction. Introducing notation $\tilde{\boldsymbol{\varphi}}'_{1,n} = \begin{bmatrix}\varphi_1\left(\tilde{\mathbf{x}}_{1,n}\right) & \cdots & \varphi_{d_1}\left(\tilde{\mathbf{x}}_{1,n}\right)\end{bmatrix}$ and

$$\tilde{\boldsymbol{\varphi}}'_{m+1,n} = \tilde{\boldsymbol{\varphi}}'_{m,n}\mathbf{A}_{m,\text{child}}^T \quad (11)$$

for $m = 1, \ldots, M-1$, we find the matrix partial derivatives to be:

$$\mathbf{E}_{\mathbf{A}_m} = \sum_{n=1}^N y_n \mathrm{L}'\left(y_n\varphi\left(\tilde{\mathbf{x}}_{1,n}\right)\right)\begin{bmatrix}\mathbf{x}_{m,n}\\ \tilde{\mathbf{x}}_{m+1,n}\end{bmatrix}\tilde{\boldsymbol{\varphi}}'_{m,n} \quad (12)$$

for $m = 1, \ldots, M-1$, and

$$\mathbf{E}_{\mathbf{A}_M} = \sum_{n=1}^N y_n \mathrm{L}'\left(y_n\varphi\left(\tilde{\mathbf{x}}_{1,n}\right)\right)\mathbf{x}_{M,n}\tilde{\boldsymbol{\varphi}}'_{M,n}. \quad (13)$$

The message sensor m receives from its parent contains $y_n \mathrm{L}'\left(y_n\varphi\left(\tilde{\mathbf{x}}_{1,n}\right)\right)$ and $\tilde{\boldsymbol{\varphi}}'_{m,n}$. Calculating the outgoing forward and backward messages from the incoming ones, given in (9) and (11), requires only simple matrix-vector products.

Based on the parallel and serial fusion networks presented, it is straightforward to generalize the joint linear dimensionality reduction and classification to any tree-structured network.

4.2 Radar System Example

We now present classification results with linear dimensionality reduction in fusion networks on the ionosphere dataset from the UCI machine learning repository [20]. The dataset contains $N = 351$ samples of $D = 34$-dimensional radar return data collected by a system in Goose Bay, Labrador. The two classes of returns are those that show structure in the ionosphere and those that do not. We report training and test error through tenfold cross-validation, i.e. we split the dataset into ten roughly equal pieces, train on nine tenths and test on one tenth for each of the ten splits, and report the average error. For the classifier, we use the SVM with radial basis function kernel and default parameters from the Matlab bioinformatics toolbox.

First we look at the training and test error in parallel networks as a function of $\sum_{m=1}^M d_m$ for different numbers of sensors M. The thirty-four dimensions are allocated to the sensors as equally as possible in the order given in the repository, e.g. the first twelve dimensions to sensor 1, the next eleven dimensions to sensor 2, and

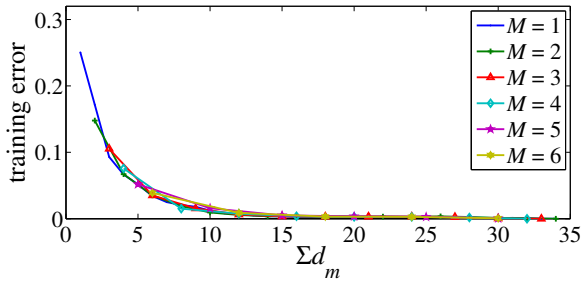

Figure 3: Tenfold cross-validation training error on ionosphere dataset for parallel fusion network optimized using joint linear dimensionality reduction and classification coordinate descent.

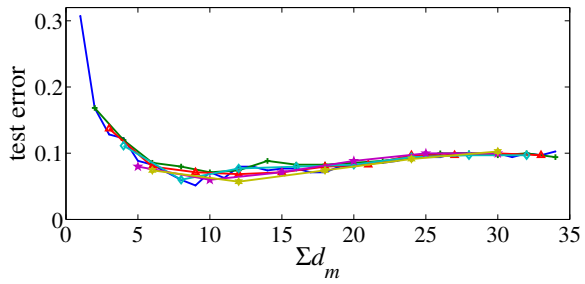

Figure 4: Tenfold cross-validation test error on ionosphere dataset for parallel fusion network optimized using joint linear dimensionality reduction and classification coordinate descent.

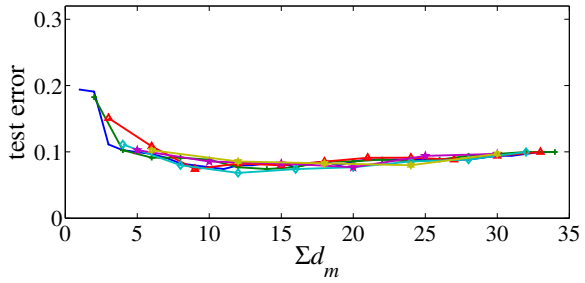

Figure 5: Tenfold cross-validation test error on ionosphere dataset for parallel fusion network with linear dimensionality reduction by PCA.

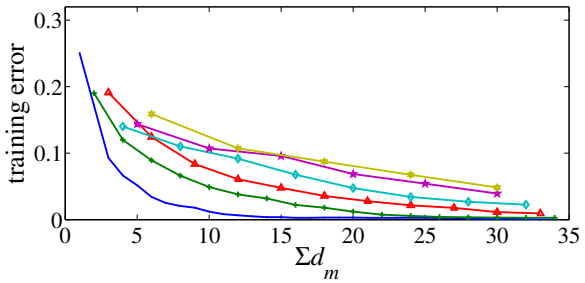

Figure 6: Tenfold cross-validation training error on ionosphere dataset for serial fusion network optimized using joint linear dimensionality reduction and classification coordinate descent.

the last eleven dimensions to sensor 3 for $M = 3$. The reduced dimension d_m is set to be equal for all sensors and varied to take values from one to the highest value such that $Md_m \leq D$. Fig. 3 shows the training error and Fig. 4 shows the test error for the φ and $\mathbf{A}_m$ resulting from the coordinate descent optimization. The legend in Fig. 3 also applies to Fig. 4 through Fig. 8. For comparison purposes, Fig. 5 shows the test error for $\mathbf{A}_m$ not optimized for margin-based classification, but obtained from PCA.

The first thing to notice in the plots is that the training error decreases monotonically with $\sum d_m$, whereas the test error first decreases, but then increases. This effect is a manifestation of the structural risk minimization principle: increasing complexity eventually leads to overfitting and increasing test error even though training error continues to decrease. The second thing to notice is that there is a negligible difference in the classification performance for different numbers of sensors. Not much is lost in a parallel arrangement when doing the dimensionality reduction at individual sensors. The third thing to notice is that the minimum test performance with the Stiefel manifold optimization is less than that using PCA matrices, so clearly the optimization with classification objective is providing a benefit. The minimum test error also occurs for a smaller value of $\sum d_m$ with the classification-optimized $\mathbf{A}_m$, so the optimization yields better performance with less resource usage in testing.

We also look at the training and test error in serial networks as a function of $\sum d_m$ for different numbers of sensors. The measurement dimensions are allocated to the sensors in the same way as for the parallel sensor results. Here also, the reduced dimension d_m is set to be equal for all sensors and varied to take values from one to the highest value such that $Md_m \leq D$. Fig. 6 and Fig. 7 show the training and test error, respectively, for margin-based classification-optimized $\mathbf{A}_m$. Fig. 8 shows the test classification error for PCA matrices.

As for the parallel network, the serial network exhibits overfitting, especially with one, two, and three sensors. The overfitting regime has not been reached for the larger numbers of sensors. Unlike the parallel network, there are significant differences in both training and test error with the number of sensors. The error plots are lower and to the left for decreasing values of M. In the serial network, the decision function is defined in d_1 dimensions rather than in $\sum d_m$ dimensions in the parallel network, and also the information

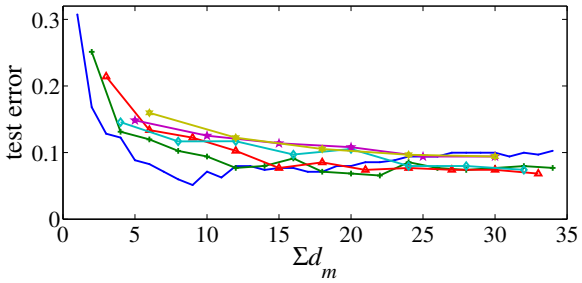

Figure 7: Tenfold cross-validation test error on ionosphere dataset for serial fusion network optimized using joint linear dimensionality reduction and classification coordinate descent.

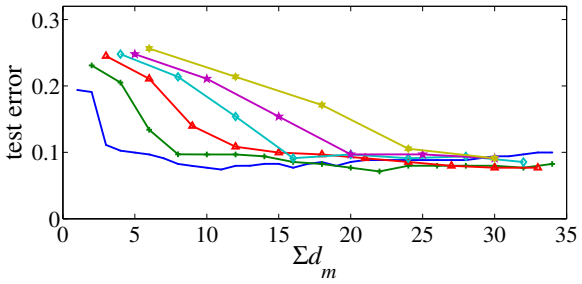

Figure 8: Tenfold cross-validation test error on ionosphere dataset for serial fusion network with linear dimensionality reduction by PCA.

is fused as it passes to the fusion center. For the same amount of data transmitted by sensors, less effective information is communicated in the serial network than in the parallel network and more so as the number of sensors increases. As with the parallel network, the comparison of the coordinate descent-optimized network and the PCA matrix network test performances indicate that the matrix optimization yields better classification performance and moreover does so with less communication.

4.3 Model Selection

A question one may ask is how to choose the reduced dimension from the training data alone, without access to the test data. Any popular model selection method, including those based on cross-validation, bootstrapping, and information criteria, can be used. As an example, let us look at an Akaike-like information criterion [21] for the fusion network with single sensor. The number of free parameters in a matrix in $\mathcal{V}_{D,d}$ is $k = Dd - d(d+1)/2$ and the information criterion we consider is:

$$2k + N \ln(\text{training error}).$$

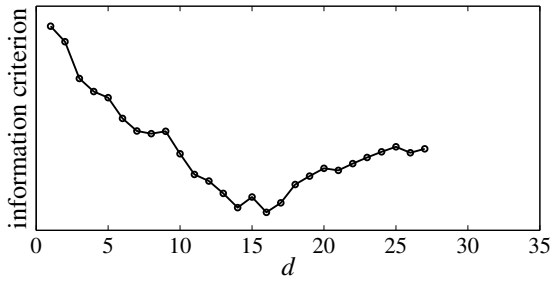

Figure 9: Akaike-like information criterion on ionosphere dataset as a function of the dimensionality of the reduced space for the fusion network with single sensor.

The value of this information criterion as a function of d is plotted in Fig. 9 for the ionosphere dataset. We see that the lowest information criterion value occurs for an intermediate value of d close to the value yielding minimum test error.

This type of model selection considers only overfitting effects; in sensor networks, resource constraints also need to be taken into account. The parallel network is preferred from the perspective of classification performance per amount of data transmitted. However, in a physically realized wireless sensor network, for example, the parallel network requires much long distance communication from sensors to the fusion center and incurs more resource cost per amount of data transmitted due to path attenuation. Thus selecting the topology of the fusion network along with the d_m is complicated for general tree-structured fusion networks.

5 Conclusion

In this paper, we have described a formulation for linear dimensionality reduction driven by the objective of margin-based classification. This involves alternation between two minimizations: one to update a classifier decision function and the other to update a matrix on the Stiefel manifold. Dimensionality reduction is important for two distinct reasons: reducing the amount of resources consumed, and avoiding overfitting.

We have described how our proposed optimization scheme can be distributed in a network containing a remote sensor, 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 parallel, serial, and tree-structured fusion networks. The joint dimensionality reduction and classification has superior classification performance to that of a dimensionality reduction method not matched to the classification task, PCA, as would be expected. However, also, the best classification performance occurs for smaller reduced dimension with the joint optimization.

The formulation we have presented opens many interesting questions regarding model selection, network topology selection, and sensor management. It would also be interesting to investigate semi-supervised learning in this context, and explore possible connections to feedforward neural networks and training by the backpropagation algorithm.

6 Acknowledgment

This work was supported in part by a MURI funded through ARO Grant W911NF-06-1-0076 and by a MURI funded through AFOSR Grant FA9550-06-1-0324. The authors thank Justin H. G. Dauwels and John W. Fisher III for input.

References

[1] M. Gladwell, *Blink: The Power of Thinking Without Thinking*. New York: Little, Brown and Company, 2005.

[2] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA: MIT Press, 2002.

[3] A. Srivastava and X. Liu, "Tools for application-driven linear dimension reduction," *Neurocomputing*, vol. 67, pp. 136–160, Aug. 2005.

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

[5] 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.

[6] 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.

[7] 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.

[8] ——, "Distributed learning in wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 23, no. 4, pp. 56–69, Jul. 2006.

[9] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer, 1995.

[10] Y. Lin, "A note on margin-based loss functions in classification," *Stat. Probabil. Lett.*, vol. 68, no. 1, pp. 73–82, Jun. 2004.

[11] 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.

[12] K. R. Varshney and A. S. Willsky, "Supervised learning of classifiers via level set segmentation," in *Proc. IEEE Workshop Mach. Learn. Signal Process.*, Cancún, Mexico, Oct. 2008, pp. 115–120.

[13] 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.

[14] 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, Alaska, Jun. 2008.

[15] 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.

[16] 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.

[17] 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.

[18] J. H. Manton, "Optimization algorithms exploiting unitary constraints," *IEEE Trans. Signal Process.*, vol. 50, no. 3, pp. 635–650, Mar. 2002.

[19] Y. Nishimori and S. Akaho, "Learning algorithms utilizing quasi-geodesic flows on the Stiefel manifold," *Neurocomputing*, vol. 67, pp. 106–135, Aug. 2005.

[20] A. Asuncion and D. J. Newman, "UCI machine learning repository," Available at http://archive.ics.uci.edu/ml/, 2007.

[21] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Autom. Control*, vol. AC-19, no. 6, pp. 716–723, Dec. 1974.