Hierarchical Linear Discriminant Analysis for Beamforming

Jaegul Choo*, Barry L. Drake†, and Haesun Park*

Abstract

This paper demonstrates the applicability of the recently proposed supervised dimension reduction, hierarchical linear discriminant analysis (h-LDA) to a well-known spatial localization technique in signal processing, beamforming. The main motivation of h-LDA is to overcome the drawback of LDA that each cluster is modeled as a unimodal Gaussian distribution. For this purpose, h-LDA extends the variance decomposition in LDA to the subcluster level, and modifies the definition of the within-cluster scatter matrix. In this paper, we present an efficient h-LDA algorithm for oversampled data, where data dimension is larger than the dimension of the data vectors. The new algorithm utilizes the Cholesky decomposition based on a generalized singular value decomposition framework. Furthermore, we analyze the data model of h-LDA by relating it to the two-way multivariate analysis of variance (MANOVA), which fits well within the context of beamforming applications. Although beamforming has been generally dealt with as a regression problem, we propose a novel way of viewing beamforming as a classification problem, and apply a supervised dimension reduction, which allows the classifier to achieve better accuracy. Our experimental results demonstrate that h-LDA outperforms several dimension reduction methods such as LDA and kernel discriminant analysis, and regression approaches such as the regularized least squares and kernelized support vector regression.

*College of Computing, Georgia Institute of Technology, 266 Ferst Drive, Atlanta, GA 30332, USA, {joyfull, hpark}@cc.gatech.edu

†SEAL/AMDD, Georgia Tech Research Institute, 7220 Richardson Road, Smyrna, GA 30080, USA, barry.drake@gtri.gatech.edu

The work of these authors was supported in part by the National Science Foundation grants CCF-0621889 and CCF-0732318. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

I. Overview of Our Work

The key innovation of this paper is to cast one of the famous problems in signal processing, beamforming, as a classification problem, and then to apply a supervised dimension reduction technique with the goal of classification performance improvement. The method we focus on is the newly proposed dimension reduction method, hierarchical linear discriminant analysis (h-LDA) [**?**]. h-LDA is intended to ameliorate the essential problem of LDA [**?**] that each cluster in the data has to have a single Gaussian distribution. In [**?**], h-LDA has shown its strength in face recognition in which persons' images vary significantly depending on angles, illuminations, and facial expressions.

Beamforming is used in various areas such as radar, sonar, and wireless communications, and is primarily divided into two areas: transmission and reception. In transmission, beamforming deploys the antenna array with carefully chosen amplitudes and phases so that it can have maximal directionality of the transmitted signal in space. In reception, by deploying the antenna arrays and processing their received signals, beamforming plays the role of maximizing the antenna array gain in the direction of a desired signal while minimizing the gain in other directions, where interfering signals (such as jammers) may be present in the signal space. These directions are called angles of arrival (AOA's) of the respective signals. This paper addresses the latter case of the passive receiver.

Since the original dimension is usually not very high in beamforming, e.g., the number of antennas or array elements, compared to other applications such as facial image data or text document corpus, the role of a dimension reduction is not merely a peripheral preprocessing step, e.g. for noise removal, but a part of the classifier building step itself. Furthermore, the fact that the beamforming problem usually has a small number of classes, e.g. two in the case where the source sends binary signals, applying a supervised dimension reduction virtually covers up the role of the classifier. To be specific, the optimal reduced dimension of LDA is $c-1$, where c is the number of classes, and LDA would reduce the dimension to simply one for binary signals in beamforming. If we think of a classification task also as a dimension reduction to one followed by the comparison to a certain threshold, the transformation matrix obtained from a supervised dimension reduction method such as LDA in this case is, in practice, equivalent to coefficients learning of a claasifier model, which is the main job when building a classifier

such as the support vector machine. Starting from such a motivation, we show how powerful the idea of applying a supervised dimension reduction turns out to be for the beamforming problem compared to the traditional methods such as regularized least squares fitting and the recently applied techniques, kernelized support vector regression (SVR). From extensive simulation, we also draw the key observation that the hierarchical LDA (h-LDA) is superior to various supervised dimension reduction methods such as LDA and the kernel discriminant analysis (KDA). From these experiments, and by utilizing the Cholesky decomposition, we develop an efficient h-LDA algorithm for the oversampled case, where the number of data samples is larger than the data vector dimension, such as in beamforming. Also, we present an analysis for the data model inherent in h-LDA by relating it to two-way multivariate analysis of variance (MANOVA), which shows that the additive nature in the h-LDA data model fits exactly to that of beamforming.

The paper is organized as follows: In Section 2, we briefly describe the beamforming problem and its approaches. Our supervised dimension reduction, h-LDA, and its algorithmic details are discussed in Section 3. In Section 4, we show the theoretical insight about the common characteristic in the data models between h-LDA and beamforming based on two-way multivariate analysis of variance (MANOVA). In Section 5, we present simulation examples of the application of h-LDA to beamforming, and finally conclusions are discussed in Section 6.

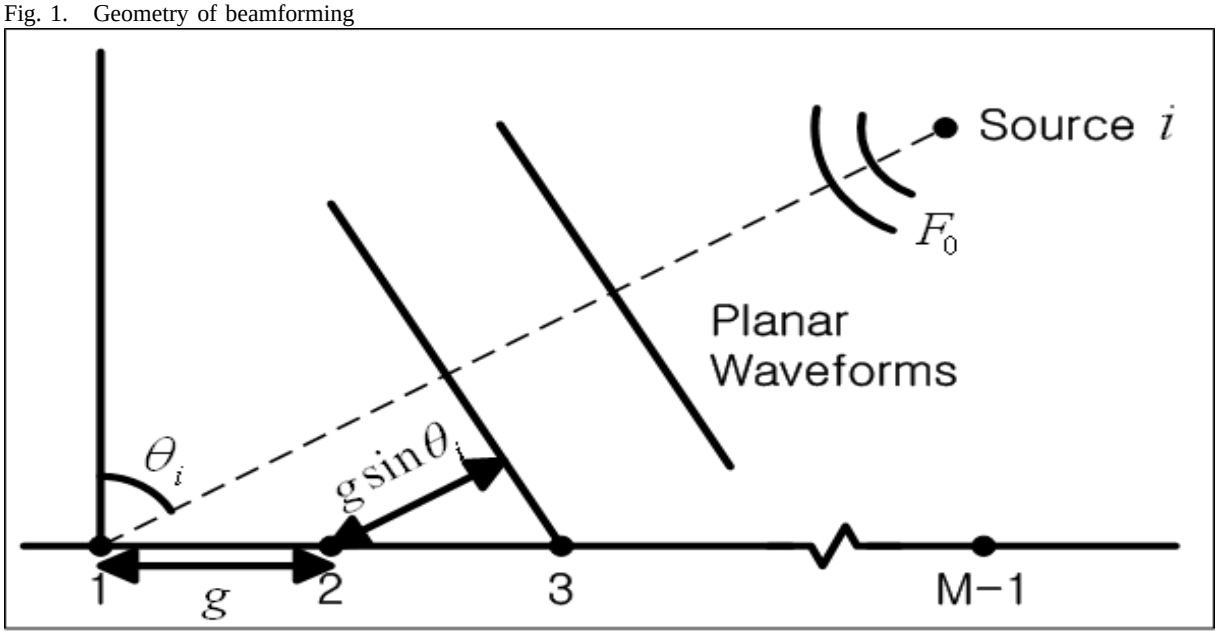
II. Beamforming and Recent Development

Let us assume we have an M element antenna array of receivers, and L sources s_i's with different angles of arrival (AOA's) θ_i, $1 \leq i \leq L$ that can be well resolved in angle. Here the source signal s_d with θ_d, where d is the index of the desired signal, is assumed to be the signal we want to detect, i.e. the desired signal. Then the measurement vector or signal model $x[n] \in \mathbb{C}^{M\times 1}$ of the M element array at time index n can be written as

$$x[n] = As[n] + e[n], \tag{1}$$

where $s[n] = \begin{bmatrix} s_1[n] & s_2[n] & \cdots & s_L[n] \end{bmatrix}^T \in \mathbb{R}^{L\times 1}$ are the signals from the L sources, and $e[n] \in \mathbb{C}^{M\times 1}$ is an additive white Gaussian noise vector impinging on each element in an antenna array. In this paper, we assume for simplicity the signal $s[n]$ to be the binary phase-shift keying signal having either -1 or 1. However, it should be noted that the ideas in this paper can be easily generalized to other types of signals, such as radar receivers with linear frequency

Fig. 1. Geometry of beamforming



modulated (LFM) or pulsed Doppler waveforms, which will be the topic of future work. Each column of $A = \left[\begin{array}{cccc} a(\theta_1) & a(\theta_2) & \cdots & a(\theta_M) \end{array} \right] \in \mathbb{C}^{M \times L}$ is the so-called steering vector for each source defined as

$$a(\theta_i) \quad = \quad \left[\begin{array}{c} e^{j2\pi f_i \cdot 0} \\ e^{j2\pi f_i \cdot 1} \\ \vdots \\ e^{j2\pi f_i \cdot (M-1)} \end{array} \right] \in \mathbb{C}^{M \times 1}$$

with the spatial frequency f_i as

$$f_i = F_0 \frac{g}{c} \sin\theta_i,$$

where g is the distance between consecutive sensors in the array and c is the velocity of light (See Figure **??**). Notice from Eq. (**??**), when we want to communicate only with the desired signal, not only does the Gaussian noise $e[n]$ corrupt the desired signal, but signals from the other sources may act as interference further reducing the array gain in the direction of the desired signal.

The output of a linear beamformer (linear combiner) is defined as

$$y[n] = w^H x[n],$$

where $w \in \mathbb{C}^{M\times 1}$ is the vector of beamformer weights. The task of beamforming is typically, assuming that certain transmitted data is known for training purposes, to decide the optimal vector w so that the detection performance of the desired signal is maximized and the energy contained in the sidelobes is as small as possible. The estimation $\hat{s}_d[n]$ for the desired signal $s_d[n]$ is made based on

$$\hat{s}_d[n] = \begin{cases} +1 & \text{if } \Re(y[n]) \geq 0 \\ -1 & \text{if } \Re(y[n]) < 0, \end{cases} \tag{2}$$

where $\Re(\cdot)$ indicates the real part of a complex number in the parenthesis.

There are generally two different approaches for beamforming depending on whether the value of θ_d, the AOA of the desired signal, is available or not. When θ_d is available, the well-known minimum variance distortionless response (MVDR) method minimizes the variance $w^H R_{xx} w$ at the beamformer output $y[n]$ subject to the constraint $w^H a(\theta_d) = 1$, where R_{xx} is the covariance of $x[n]$. In this case the optimal solution w_{MVDR} is obtained as

$$w_{MVDR} = \frac{R_{xx}^{-1} a(\theta_d)}{a(\theta_d)^H R_{xx}^{-1} a(\theta_d)}.$$

On the other hand, if θ_d is unknown, which is the case of interest in this paper, one can use a minimum mean square error (MMSE) estimator for w, which is shown as

$$w = R_{xx}^{-1} r_{xy}, \tag{3}$$

where r_{xy} is the cross-covariance between $x[n]$ and $y[n]$ provided that $x[n]$ and $y[n]$ are zero mean. With no assumptions on the probability distribution on $x[n]$, this MMSE solution is easily converted to least squares (LS) fitting within the context of regression analysis as follows. Suppose the time index n of the training data goes from 1 to N (usually N is larger than M), and denote the training data matrix X and its known signal vector of the desired source s_d, respectively, as

$$X = \begin{bmatrix} x[1] & x[2] & \cdots & x[N] \end{bmatrix} \in \mathbb{C}^{M\times N} \text{ and}$$

$$s_d = \begin{bmatrix} s_d[1] \\ s_d[2] \\ \vdots \\ s_d[N] \end{bmatrix} \in \{-1,\, 1\}^N .$$

Then we can set up the overdetermined system for w as

$$w^H X \simeq s_d^T \Leftrightarrow X^H w \simeq s_d$$

and its least squares solution is

$$w_{LS} = (XX^H)^{-1} X s_d,$$

where the covariance R_{xx} and the cross-covariance r_{xy} in Eq. (**??**) are replaced by the sample covariances XX^H and Xs_d respectively. Also, in order to enhance the robustness against noise in X, one can introduce a regularization term γI into the least squares solution as

$$w_{RLS} = (XX^H + \gamma I)^{-1} X s_d, \tag{4}$$

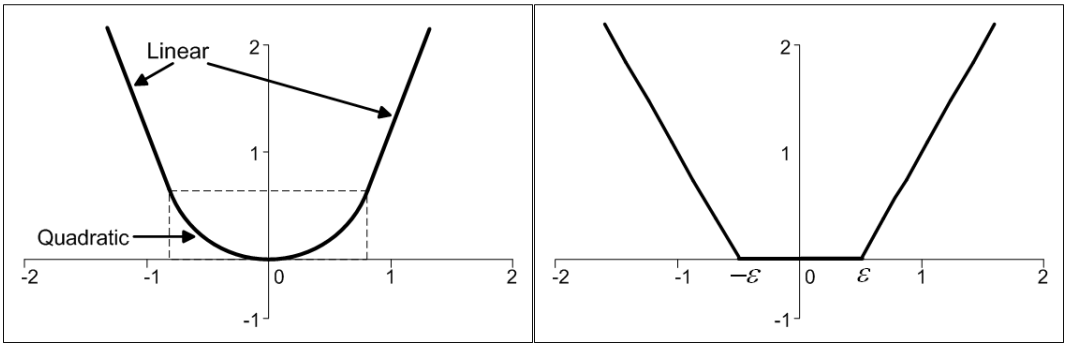
which is equivalent to imposing regularization on the Euclidean norm of w with a weighting parameter γ. As a way of improving the quality of an estimate of R_{xx}, a relevance vector machine (RVM) method was applied recently from the Bayesian perspective [**?**].

Whereas the least squares or the regularized least squares approach takes the squared loss function, which is sensitive to outliers that do not have a Gaussian error distribution, other types of regression approaches utilize a loss function less sensitive to outliers such as the Huber loss function, or ϵ-sensitive loss function as shown in Figure **??**. When an ϵ-sensitive loss function combined with the Euclidean regularization for w is used, the regression framework becomes the support vector regression (SVR) where w is determined only by a subset of the data. Recently, SVR has been applied to the beamforming problem and resulted in good bit error rate (BER) performance over existing methods such as the regularized LS or MVDR [**?**], [**?**].

Since the original dimension of beamforming problems, which corresponds to the number of sensors in an antenna array, is usually not high, e.g. on the order of tens or less, the data is often not easily fitted as a linear model. Consequently, nonlinear kernelization techniques that take the data to much higher dimensions were also applied [**?**], [**?**]. In particular, the kernelized SVR using Gaussian kernel demonstrated outstanding results for beamforming applications both by handling nonlinearity of the data and by adopting a robust ϵ-sensitive loss function [**?**]. The ability to handle nonlinear input data is of fundamental importance to the beamforming community and is a significant outstanding problem. The use of Volterra filters has had very limited success since they are very sensitive numerically. In order to handle a larger class of signals and account for nonlinearities of antenna response patterns, it is vital that the ability to handle nonlinear input

Fig. 2. Example of Robust loss functions

(a) Huber loss function (b) ϵ-sensitive loss function



data be incorporated into the processing algorithms for such data. This issue often renders the solutions from current technology meaningless in the face of the increasing sophistication of countermeasures and demands for more accurate characterization of the signal space [**?**].

III. HIERARCHICAL LDA FOR SUB-CLUSTERED DATA AND BEAMFORMING

In this section, we present a novel approach for solving the beamforming problem by viewing it as a classification problem rather than a regression that fits the data to the desired signal s_d. In other words, from a regression point of view, w is solved so that the value of $y[n] = w^H x[n]$ can be as close as possible to the desired signal value itself, which is either -1 or 1 in beamforming. However, our perspective is that all the values of $y[n]$ obtained by applying w^H does not necessarily need to gather around such target values. Even though they are represented in a wide range of values, instead, as long as they are easily separable in the reduced dimension of $y[n]$, incorporating just a simple classification technique would give us a good signal detection performance in the reduced dimension. We can then solve for w such that the resulting values of $y[n]$ can just be classified to either class as correctly as possible by a classifier in the reduced dimension. Such an idea naturally leads us to the possibility of applying supervised dimension reduction methods that can facilitate the classification. Based on this motivation, we focus on applying one of the recently proposed methods, h-LDA, which handles multi-modal Gaussian data. In what follows, we will describe h-LDA in detail and show how the h-LDA algorithm can be efficiently developed using the Cholesky decomposition for oversampled problems such as beamforming. We start with a brief introduction of LDA, which is the basis for our analytic and algorithmic contents of h-LDA.

A. Linear Discriminant Analysis (LDA)

LDA obtains an optimal dimension-reduced representation of the data by a linear transformation that maximizes the *conceptual* ratio of the between-cluster scatter (variance) versus the within-cluster scatter of the data [**?**], [**?**].

Given a data matrix $A = [a_1\, a_2 \cdots a_n] \in \mathbb{C}^{m\times n}$, where m columns a_i, $i = 1, \ldots, n$, of A represent n data items in an m dimensional space, assume that the columns of A are partitioned into p clusters as

$$A = [A_1 \quad A_2 \quad \cdots \quad A_p],$$

where

$$A_i \in \mathbb{C}^{m\times n_i} \text{ and } \sum_{i=1}^{p} n_i = n.$$

Let $\mathcal{N}_i$ denote the set of column indices that belong to cluster i, n_i the size of $\mathcal{N}_i$, a_k the data point represented in the k-th column vector of A, $c^{(i)}$ the centroid of the i-th cluster, and c the global centroid.

The scatter matrix within the i-th cluster $S_w^{(i)}$, the within-cluster scatter matrix S_w, the between-cluster scatter matrix S_b, and the total (or mixture) scatter matrix S_t are defined [**?**], [**?**], respectively, as

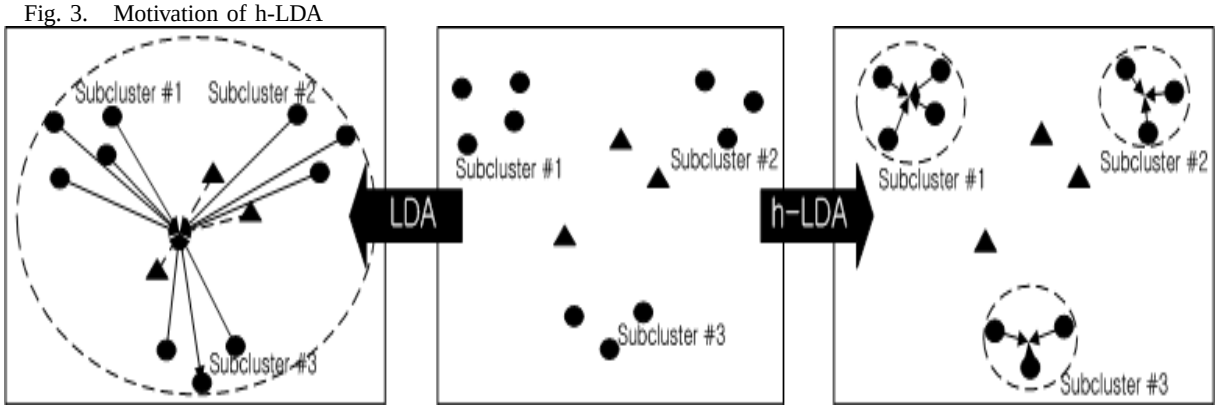
$$S_w^{(i)} = \sum_{k\in\mathcal{N}_i} (a_k - c^{(i)})(a_k - c^{(i)})^T,$$

$$\begin{aligned} S_w &= \sum_{i=1}^{p} S_w^{(i)} \\ &= \sum_{i=1}^{p} \sum_{k\in\mathcal{N}_i} (a_k - c^{(i)})(a_k - c^{(i)})^T, \end{aligned} \tag{5}$$

$$\begin{aligned} S_b &= \sum_{i=1}^{p} \sum_{k\in\mathcal{N}_i} (c^{(i)} - c)(c^{(i)} - c)^T \\ &= \sum_{i=1}^{p} n_i (c^{(i)} - c)(c^{(i)} - c)^T, \text{ and} \end{aligned} \tag{6}$$

$$\begin{aligned} S_t &= \sum_{k=1}^{n} (a_k - c)(a_k - c)^T \\ &= S_w + S_b. \end{aligned} \tag{7}$$

Fig. 3. Motivation of h-LDA



In the lower dimensional space obtained by a linear transformation

$$G^T : x \in \mathbb{C}^{m\times 1} \rightarrow y \in \mathbb{C}^{l\times 1},$$

the within-cluster, the between-cluster, and the total scatter matrices become

$$S_w^Y = G^T S_w G,\ S_b^Y = G^T S_b G,\ \text{and}\ S_t^Y = G^T S_t G,$$

where the superscript Y denotes the scatter matrices in the l dimensional space obtained by applying G^T. In LDA, an optimal linear transformation matrix G^T is found so that it minimizes the within-cluster scatter measure, trace(S_w^Y), and at the same time, maximizes the between-cluster scatter measure, trace(S_b^Y). This optimization problem of two distinct measures is usually replaced with one that maximizes

$$J(G) = \text{trace}((G^T S_w G)^{-1}(G^T S_b G)), \tag{8}$$

which is the ratio of the within-cluster radius and the between-cluster distance in the reduced dimensional space.

B. Hierarchical LDA (h-LDA)

h-LDA was originally proposed to mitigate a shortcoming of LDA. Specifically, the data distribution in each cluster is assumed to be a unimodal Gaussian distribution in LDA. However, data from many real-world problems cannot be explained by such a restrictive assumption, and this assumption may manifest itself in practical ways when applying LDA in real applications. Severe distortion in the original data may result causing the within-cluster radius to be as tight as possible without considering their multi-modality. Consider the case of Fig. **??**, where the

triangular points belong to one cluster, and the circular points to the other cluster which can be further clustered into three subclusters. LDA may represent the data from these two clusters close to each other as a result of minimizing the within-cluster radius among circular points. On the other hand, h-LDA can avoid ths problem by textitemphasizing within-subcluster structure based on further variance decomposition of S_w in Eq. (**??**) and the modification of its definition. In this respect, one significant advantage of h-LDA is that it reduces such distortions due to multi-modality. Following is the formulation of h-LDA, which solves this fundamental problem of LDA.

h-LDA assumes that the data in cluster i, A_i, can be further clustered into q_i subclusters as

$$A_i = [A_{i1} \quad A_{i2} \quad \cdots \quad A_{iq_i}],$$

where

$$A_{ij} \in \mathbb{C}^{m \times n_{ij}}, \quad \sum_{j=1}^{q_i} n_{ij} = n_i.$$

Let $\mathcal{N}_{ij}$ denote the set of column indices that belong to the subcluster j in cluster i, n_{ij} the size of $\mathcal{N}_{ij}$ and $c^{(ij)}$ the centroid of each subcluster. Then, we can define the scatter matrix within subcluster j of cluster i, $S_{w_s}^{(ij)}$, their sum in cluster i, $S_{w_s}^{(i)}$, and the scatter matrix between subclusters in cluster i, $S_{b_s}^{(i)}$, respectively, as

$$S_{w_s}^{(ij)} = \sum_{k \in \mathcal{N}_{ij}} (a_k - c^{(ij)})(a_k - c^{(ij)})^T,$$

$$\begin{aligned} S_{w_s}^{(i)} &= \sum_{j=1}^{q_i} S_{w_s}^{(ij)} \\ &= \sum_{j=1}^{q_i} \sum_{k \in \mathcal{N}_{ij}} (a_k - c^{(ij)})(a_k - c^{(ij)})^T, \text{ and} \end{aligned}$$

$$\begin{aligned} S_{b_s}^{(i)} &= \sum_{j=1}^{q_i} \sum_{k \in \mathcal{N}_{ij}} (c^{(ij)} - c^{(i)})(c^{(ij)} - c^{(i)})^T \\ &= \sum_{j=1}^{q_i} n_{ij} (c^{(ij)} - c^{(i)})(c^{(ij)} - c^{(i)})^T. \end{aligned}$$

Then, the within-subcluster scatter matrix S_{w_s} and the between-subcluster scatter matrix S_{b_s} are defined, respectively, as

$$\begin{aligned} S_{w_s} &= \sum_{i=1}^{p} S_{w_s}^{(i)} = \sum_{i=1}^{p}\sum_{j=1}^{q_i} S_{w_s}^{(ij)} \\ &= \sum_{i=1}^{p}\sum_{j=1}^{q_i}\sum_{k\in\mathcal{N}_{ij}} (a_k - c^{(ij)})(a_k - c^{(ij)})^T, \end{aligned} \tag{9}$$

$$\begin{aligned} S_{b_s} &= \sum_{i=1}^{p} S_{b_s}^{(i)} \\ &= \sum_{i=1}^{p}\sum_{j=1}^{q_i}\sum_{k\in\mathcal{N}_{ij}} (c^{(ij)} - c^{(i)})(c^{(ij)} - c^{(i)})^T \\ &= \sum_{i=1}^{p}\sum_{j=1}^{q_i} n_i (c^{(ij)} - c^{(i)})(c^{(ij)} - c^{(i)})^T. \end{aligned}$$

From the identity

$$a_k - c = (a_k - c^{(ij)}) + (c^{(ij)} - c^{(i)}) + (c^{(i)} - c),$$

it can be proved that

$$S_t = S_{w_s} + S_{b_s} + S_b \tag{10}$$

where the between-cluster scatter matrix S_b is defined as in Eq. (**??**). Comparing Eq. (**??**) with Eq. (**??**), the within-cluster scatter matrix S_w in LDA is equivalent to the sum of the within-subcluster scatter matrix S_{w_s} and the between-subcluster scatter matrix S_{b_s} as

$$S_w = S_{w_s} + S_{b_s}. \tag{11}$$

Now we propose a new within-cluster scatter matrix S_w^h, which is a convex combination of S_{w_s} and S_{b_s} as

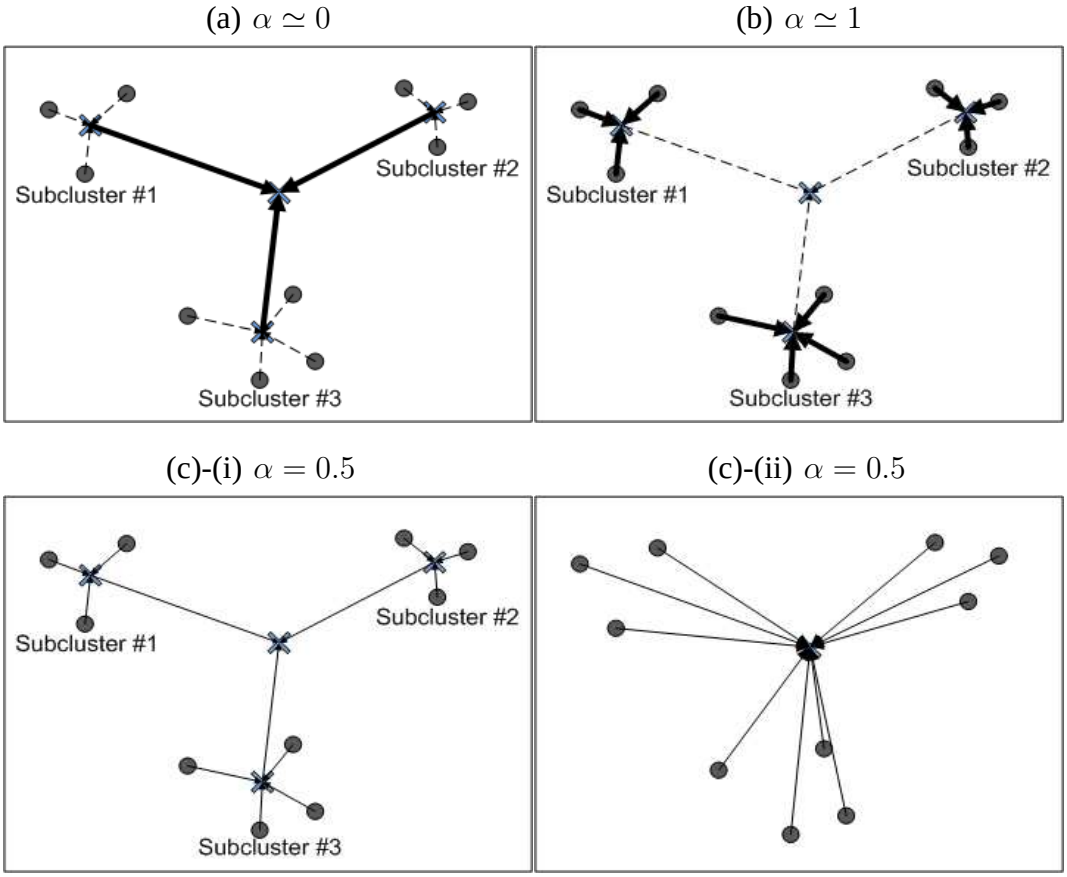
$$S_w^h = \alpha S_{w_s} + (1-\alpha) S_{b_s}, \quad 0 \le \alpha \le 1, \tag{12}$$

where α determines relative weights between S_{w_s} and S_{b_s}. By replacing S_w with the newly-defined S_w^h, h-LDA finds the solution that maximizes the new criterion

$$J^h(G) = \text{trace}((G^T S_w^h G)^{-1}(G^T S_b G)). \tag{13}$$

Consider the following three cases: $\alpha \simeq 0$, $\alpha \simeq 1$, and $\alpha = 0.5$. When $\alpha \simeq 0$ (see Figure **??**(a)), the within-subcluster scatter matrix S_{w_s} is disregarded and the between-subcluster scatter

Fig. 4. Behavior of h-LDA depending on the parameter α. All data points in each figure belong to one cluster.



matrix S_{b_s} is emphasized, which can be considered as the original LDA applied after every data point is relocated to its corresponding subcluster centroid. When $\alpha \simeq 1$ (see Figure **??**(b)), h-LDA minimizes only the within-subcluster radii, disregarding the distances between subclusters within each cluster. When $\alpha = 0.5$, the within-subcluster scatter matrix S_{w_s} and the between-subcluster scatter matrix S_{b_s} are equally weighted so that h-LDA becomes equivalent to LDA by Eq. (**??**), which shows the equivalence of the within-cluster scatter matrices between Figure **??**(c)-(i) and **??**(c)-(ii). Hence, h-LDA can be viewed as a generalization of LDA, and the parameter α can be chosen by parameter optimization schemes such as cross-validation in order to attain maximum classification performance. Considering the motivation of h-LDA, attention should be paid to the case of $0.5 < \alpha \simeq 1$ since this can mitigate the unimodal Gaussian assumption weakness of the classical LDA, which can produce a transformation that projects the points in one cluster onto essentially one point in the reduced dimensional space.

C. Efficient Algorithm for h-LDA

In this section, we present the algorithm for h-LDA and its efficient version using the Cholesky decomposition in oversampled cases, i.e. $m < n$. The basic algorithm for h-LDA is primarily based on the generalized singular value decomposition (GSVD) framework, which has its foundations in the original LDA solution via the GSVD [**?**], [**?**], LDA/GSVD. We also point out that the GSVD algorithm is a familiar method to the signal processing community, particularly for direction-of-arrival (DOA) estimation [**?**]. In order to describe the h-LDA algorithm, let us define the "square-root" factors, H_{w_s}, H_{b_s}, H_w^h, and H_b of S_{w_s}, S_{b_s}, S_w^h, and S_b, respectively, as

$$\begin{aligned} H_{w_s} = & \\ & [A_{11} - c^{(11)}e^{(11)^T}, \ldots, A_{1q_1} - c^{(1q_1)}e^{(1q_1)^T}, \\ & A_{21} - c^{(21)}e^{(21)^T}, \ldots, A_{2q_2} - c^{(2q_2)}e^{(2q_2)^T}, \\ & \ldots, \\ & A_{p1} - c^{(p1)}e^{(p1)^T}, \ldots, A_{pq_p} - c^{(pq_p)}e^{(pq_p)^T}] \\ & \in \mathbb{C}^{m\times n}, \end{aligned} \tag{14}$$

$$\begin{aligned} H_{b_s} = & \\ & [\sqrt{n_{11}}(c^{(11)} - c^{(1)}), \ldots, \sqrt{n_{1q_1}}(c^{(1q_1)} - c^{(1)}), \\ & \sqrt{n_{21}}(c^{(21)} - c^{(2)}), \ldots, \sqrt{n_{2q_2}}(c^{(2q_2)} - c^{(2)}), \\ & \ldots, \\ & \sqrt{n_{pq_p}}(c^{(pq_p)} - c^{(p)}), \ldots, \sqrt{n_{pq_p}}(c^{(pq_p)} - c^{(p)})] \\ & \in \mathbb{C}^{m\times s}, \end{aligned} \tag{15}$$

$$H_w^h = [\sqrt{\alpha}H_{w_s}\ \sqrt{1-\alpha}H_{b_s}] \in \mathbb{C}^{m\times(n+s)}, \text{ and} \tag{16}$$

$$\begin{aligned} H_b \ = \ & [\sqrt{n_1}(c^{(1)} - c), \sqrt{n_2}(c^{(2)} - c), \\ & \ldots, \sqrt{n_p}(c^{(p)} - c)] \in \mathbb{C}^{m\times p}, \end{aligned} \tag{17}$$

Algorithm 1 h-LDA/GSVD

Given a data matrix $A \in \mathbb{C}^{m\times n}$ where the columns are partitioned into p clusters, and each of them is further clustered into q_i clusters for $i = 1, \ldots, p$, respectively, this algorithm computes the dimension reducing transformation $G \in \mathbb{C}^{m\times(p-1)}$. For any vector $x \in \mathbb{C}^{m\times 1}$, $y = G^T x \in \mathbb{C}^{(p-1)\times 1}$ gives a $(p-1)$ dimensional representation of x.

1) Compute $H_{w_s} \in \mathbb{C}^{m\times n}$, $H_{b_s} \in \mathbb{C}^{m\times s}$, and $H_b \in \mathbb{C}^{m\times p}$ from A according to Eqs. (**??**), (**??**), and (**??**), respectively, where $s = \sum_{i=1}^{p} q_i$.
2) Compute the complete orthogonal decomposition of $K^h = \begin{pmatrix} H_b^T \\ (H_w^h)^T \end{pmatrix} = \begin{pmatrix} H_b^T \\ \sqrt{\alpha}H_{w_s}^T \\ \sqrt{1-\alpha}H_{b_s}^T \end{pmatrix} \in \mathbb{C}^{(p+n+s)\times m}$, i.e. $P^T K^h V = \begin{pmatrix} R & 0 \\ 0 & 0 \end{pmatrix}$, where $P \in \mathbb{C}^{(p+n+s)\times(p+n+s)}$ and $V \in \mathbb{C}^{m\times m}$ are orthogonal, and R is a square matrix with $rank(K^h) = rank(R)$.
3) Let $t = rank(K^h)$.
4) Compute W from the SVD of $P(1:p, 1:t)$, i.e., $U^T P(1:p, 1:t)W = \Sigma$.
5) Compute the first $p-1$ columns of $V \begin{pmatrix} R^{-1}W & 0 \\ 0 & I \end{pmatrix}$, and assign them to G.

where $s = \sum_{i=1}^{p} q_i$ and $e^{(ij)} \in \mathbb{R}^{n_{ij}\times 1}$ is a vector where all components are 1's. Then the scatter matrices can be expressed as

$$\begin{aligned} &S_{w_s} = H_{w_s} H_{w_s}^T,\ S_{b_s} = H_{b_s} H_{b_s}^T, \\ &S_w^h = H_w^h (H_w^h)^T, \text{ and } S_b = H_b H_b^T. \end{aligned} \tag{18}$$

Assuming $S_w^h = H_w^h (H_w^h)^T$ is nonsingular, it can be shown that [**?**], [**?**]

$$\text{trace}((G^T S_w^h G)^{-1} G^T S_b G) \le \text{trace}((S_w^h)^{-1} S_b) = \sum_i \lambda_i,$$

where λ_i's are the eigenvalues of $(S_w^h)^{-1} S_b$. The upper bound on $J^h(G)$ is achieved as

$$\max_G \text{trace}((G^T S_w^h G)^{-1} G^T S_b G) = \text{trace}((S_w^h)^{-1} S_b)$$

Algorithm 2 h-LDA/Chol

Given a data matrix $A \in \mathbb{C}^{m\times n}$ where the columns are partitioned into p clusters, and each of them is further clustered into q_i clusters for $i = 1, \ldots, p$, respectively, this algorithm computes the dimension reducing transformation $G \in \mathbb{C}^{m\times(p-1)}$. For any vector $x \in \mathbb{C}^{m\times 1}$, $y = G^T x \in \mathbb{C}^{(p-1)\times 1}$ gives a $(p-1)$ dimensional representation of x.

1) Compute $H_{w_s} \in \mathbb{C}^{m\times n}$, $H_{b_s} \in \mathbb{C}^{m\times s}$, and $H_b \in \mathbb{C}^{m\times p}$ from A according to Eqs. (**??**), (**??**), and (**??**), respectively, where $s = \sum_{i=1}^{p} q_i$.
2) Compute $S_w^h = [\sqrt{\alpha}H_{w_s}\ \sqrt{1-\alpha}H_{b_s}][\sqrt{\alpha}H_{w_s}\ \sqrt{1-\alpha}H_{b_s}]^T$, and its Cholesky decomposition, i.e. $S_w^h = (C_w^h)^T C_w^h$.
3) Compute the reduced QR decomposition of $K^h = \begin{pmatrix} H_b^T \\ C_w^h \end{pmatrix} \in \mathbb{C}^{(p+m)\times m}$, i.e. $P^T K^h = F$, where $P \in \mathbb{C}^{(p+m)\times(p+m)}$ has orthonormal columns and $F \in \mathbb{C}^{n\times n}$ is upper triangular.
4) Compute W from the SVD of $P(1:p, 1:t)$, i.e., $U^T P(1:p, 1:t)W = \Sigma$.
5) Compute the first $p-1$ columns of $X = F^{-1}W$, and assign them to G.

when $G \in \mathbb{R}^{m\times l}$ consists of l eigenvectors of $(S_w^h)^{-1}S_b$ corresponding to the l largest eigenvalues in the eigenvalue problem

$$(S_w^h)^{-1}S_b x = \lambda x, \tag{19}$$

where l is the number of nonzero eigenvalues of $(S_w^h)^{-1}S_b$. Since the rank of S_b is at most $p-1$, if we set $l = p-1$, and solve for G from Eq. (**??**), then we can obtain the best dimension reduction that does not lose the cluster separability measured by trace$((S_w^h)^{-1}S_b)$.

One limitation of using the criteria $J^h(G)$ in Eq. (**??**) is that S_w^h must be invertible. However, in many applications, the dimensionality m is often much greater than the number of data n, making S_w^h singular. Expressing λ as α^2/β^2, and using Eq. (**??**), Eq. (**??**) can be rewritten as

$$\beta^2 H_b H_b^T x = \alpha^2 H_w^h (H_w^h)^T x. \tag{20}$$

Then, this reformulation turns out to be a generalized singular value decomposition (GSVD) problem [**?**], [**?**], [**?**], and it can give the solution of h-LDA regardless of the singularity of S_w^h. This GSVD-based h-LDA algorithm is summarized in Algorithm **??** h-LDA/GSVD. For more details, see [**?**], [**?**].

Now we present a method to improve efficiency for the oversampled case, i.e. $m < n$, which is often the case in beamforming. This is achieved by feeding the smaller sized square root factor of S_w^h in place of H_w^h in Step 2 of Algorithm **??**. From Eqs. (**??**) and (**??**), we can find such a square root factor of S_w^h by computing the Cholesky decomposition as

$$\begin{aligned} \underbrace{S_w^h}_{m\times m} &= \underbrace{[\sqrt{\alpha}H_{w_s}\ \sqrt{1-\alpha}H_{b_s}]}_{m\times(n+s)}\underbrace{[\sqrt{\alpha}H_{w_s}\ \sqrt{1-\alpha}H_{b_s}]^T}_{(n+s)\times m} \\ &= \underbrace{(C_w^h)^T}_{m\times m}\underbrace{C_w^h}_{m\times m}. \end{aligned} \tag{21}$$

By replacing H_w^h with a newly computed square root factor, $(C_w^h)^T$, the matrix size of K^h in Step 2 of Algorithm **??** is reduced from $(p+n+s)\times m$ to $(p+m)\times m$, which makes the computation much faster. As we will show later in Section 4.2, the sum of the number of subclasses over all classes, $s = \sum_{i=1}^{p} q_i$, is equivalent to 2^L in beamforming, where L is the number of sources, and thus the efficiency improvement would be even better as the number of sources increases. In summary, the efficient h-LDA algorithm for the oversampled case is shown in Algorithm **??** h-LDA/Chol.

IV. Relationship between h-LDA and Beamforming

In this section, through the relationship between h-LDA and two-way multivariate analysis of variance (MANOVA), we show the additive nature in the h-LDA data model. Based on that, we elicit the close connection between h-LDA and beamforming, which will justify the suitability of h-LDA to beamforming problems.

A. h-LDA and MANOVA

MANOVA [**?**], [**?**] is a hypothesis testing method that determines whether the data of each cluster is significantly different from each other based on the data distribution, or equivalently, whether the treatment factor that assigns different treatments (or cluster labels) actually indicates a significantly different data distribution depending on the cluster label. For instance, suppose we have two groups of infant trees and provide only one group with plenty of water. If we observe the heights of the trees a few months later, probably the average heights of the two groups would be noticeably different compared to the variation of heights within each group, and we could conclude that the treatment factor of giving more water has a significant influence

on the data. The observed data in this example is just a one dimensional value, i.e. heights, but if the dimensionality of the data becomes larger and the number of clusters increases, then this test would require a more sophisticated measure. This is the main motivation of MANOVA.

MANOVA assumes each cluster is modeled as a Gaussian with its own mean vector but with a common covariance matrix. It can be easily seen that the estimates of the within-cluster and the between-cluster covariances correspond to Eq. (**??**) and Eq. (**??**) respectively, and Eq. (**??**) holds accordingly. Among the many MANOVA tests for the significant difference between cluster-wise data distributions, the Hotelling-Lawley trace test [**?**] uses trace$(S_w^{-1}S_b)$ as a cluster separability measure. As shown in Section 3.1, LDA gives the dimension reduced representation that preserves this measure as in the original space. Therefore, it is interesting to see that although the objective of LDA is different from that of MANOVA based on the Hotelling-Lawley trace measure, they are based on the same measure of class separability. Accordingly, the dimension reduction by LDA would not affect MANOVA tests since LDA preserves trace$(S_w^{-1}S_b)$ in the lower dimensional space.

Now we apply a similar analogy to the relationship between h-LDA and two-way MANOVA. Starting from the data model of two-way MANOVA, we derive its variance decomposition, and show the equivalence between the Hotelling-Lawley trace test and the h-LDA criterion. In two-way MANOVA, each datum is assigned a pair of cluster labels, which are determined by two treatment factors. To be more specific in the above example where an experiment with trees was considered, two treatment factors such as water and light might be considered as potential treatment factors. Depending on whether sufficient water and/or light are provided, the heights of trees are observed as a dependent variable. The Two-way MANOVA test determines if each factor has a significant effect on the height of trees as well as if the two factors are independent.

In two-way MANOVA, the k-th data point with its label pair (i, j), which corresponds to the i-th treatment from the first factor and the j-th treatment from the second factor, is modeled as

$$x_k = c + c^{(i\cdot)} + c^{(\cdot j)} + \epsilon_{ij} + \epsilon_k, \tag{22}$$

where c is the global mean, $c^{(i\cdot)}$ for $i = 1, \cdots, p$ is the mean of the data with the i-th treatment from the first factor, $c^{(\cdot j)}$ for $j = 1, \cdots, q$ is the mean of the data with the j-th treatment from the second factor, and ϵ_{ij} and ϵ_k are independent and identically distributed (i.i.d.) zero mean Gaussian random variables. Without loss of generality, we can impose the assumption that

$\sum_{i=1}^{p} c^{(i\cdot)} = 0$ and $\sum_{j=1}^{q} c^{(\cdot j)} = 0$. The model in Eq. (**??**) implies that the cluster mean with label pair (i, j) is represented as an additive model of two independent values, $c^{(i\cdot)}$ and $c^{(\cdot j)}$, with the cluster-wise error term ϵ_{ij}. Then the instance-wise error term ϵ_k is added to each datum x_k.

The total scatter matrix S_t, the residual scatter matrix S_r, the interaction scatter matrix S_i, the first factor between-cluster scatter matrix S_{b1} and the second factor between-cluster scatter matrix S_{b2} are defined respectively as

$$\begin{aligned} S_t &= \sum_{i=1}^{p}\sum_{j=1}^{q}\sum_{k\in\mathcal{N}_{ij}} (a_k - c)(a_k - c)^T \\ &= \sum_{k=1}^{n} (a_k - c)(a_k - c)^T, \end{aligned}$$

$$S_r = \sum_{i=1}^{p}\sum_{j=1}^{q}\sum_{k\in\mathcal{N}_{ij}} (a_k - c^{(ij)})(a_k - c^{(ij)})^T, \tag{23}$$

$$\begin{aligned} S_a &= \sum_{i=1}^{p}\sum_{j=1}^{q}\sum_{k\in\mathcal{N}_{ij}} (c^{(ij)} - c^{(i\cdot)} - c^{(\cdot j)} + c) \\ &\qquad (c^{(ij)} - c^{(i\cdot)} - c^{(\cdot j)} + c)^T \\ &= \sum_{i=1}^{p}\sum_{j=1}^{q} n_{ij}(c^{(ij)} - c^{(i\cdot)} - c^{(\cdot j)} + c) \\ &\qquad (c^{(ij)} - c^{(i\cdot)} - c^{(\cdot j)} + c)^T, \end{aligned}$$

$$\begin{aligned} S_{b1} &= \sum_{i=1}^{p}\sum_{j=1}^{q}\sum_{k\in\mathcal{N}_{ij}} (c^{(i\cdot)} - c)(c^{(i\cdot)} - c)^T \\ &= \sum_{i=1}^{p} n_{i\cdot}(c^{(i\cdot)} - c)(c^{(i\cdot)} - c)^T, \end{aligned}$$

$$\begin{aligned} S_{b2} &= \sum_{i=1}^{p}\sum_{j=1}^{q}\sum_{k\in\mathcal{N}_{ij}} (c^{(\cdot j)} - c)(c^{(\cdot j)} - c)^T \\ &= \sum_{j=1}^{q} n_{\cdot j}(c^{(\cdot j)} - c)(c^{(\cdot j)} - c)^T. \end{aligned}$$

From the above definitions, the total scatter matrix S_t in two-way MANOVA is decomposed as

$$S_t = S_r + S_a + S_{b2} + S_{b1}. \tag{24}$$

Assuming $q_1 = q_2 = \cdots = q_p = q$ in h-LDA, the within-subcluster scatter matrix S_{w_s} in Eq. (**??**) becomes the same as the residual scatter matrix S_r in Eq. (**??**). If we view the first factor label i as the cluster label of interest in h-LDA, and equate Eq. (**??**) and Eq. (**??**), we obtain

$$\begin{aligned} S_{b_s} &= S_a + S_{b2} \text{ and} \\ S_b &= S_{b1}. \end{aligned}$$

Now in two-way MANOVA, the Hotelling-Lawley trace [**?**] gives the class separability measures due to the first and second factors respectively as

$$\begin{aligned} H_1 &= \text{trace}(S_{w_s}^{-1} S_{b1}) \text{ and} \\ H_2 &= \text{trace}(S_{w_s}^{-1} S_{b2}). \end{aligned} \tag{25}$$

By comparing these measures with the statistically-predetermined thresholds, it is determined whether an observed response to the treatment is statistically significant. Similarly, the Hotelling-Lawley measure determines whether an interaction between two factors exists, i.e. whether two factors are independent of each other, based on

$$H_a = \text{trace}(S_{w_s}^{-1} S_a).$$

Comparing Eq. (**??**) with the h-LDA criterion of Eq. (**??**), the solution of h-LDA with $\alpha = 1$ gives the optimal linear transformation that preserves the Hotelling-Lawley trace measure H_1 in the two-way MANOVA model. Thus, this particular case of $\alpha = 1$ tells us that the underlying data model of h-LDA maintains the additive nature of two independent factors as in Eq. (**??**).

B. h-LDA and Beamforming

In this section, we examine more closely the beamforming data model in Eq. (**??**) from the perspective of Eq. (**??**). Eq. (**??**) can be rewritten as

$$x[n] = s_d[n]a(\theta_d) + \sum_{i=1,\, i\neq d}^{L} s_i[n]a(\theta_i) + e[n]. \tag{26}$$

We can map the value of $s_d[n]$, which is either of $\{-1,\, 1\}$, to the binary cluster label indicating either of $\{1,\, 2\}$, and assign a different subcluster label to each combination of the other $L-1$ signals $\left[\begin{array}{cccccc} s_1[n] & \cdots & s_{d-1}[n] & s_{d+1}[n] & \cdots & s_L[n] \end{array}\right]$, which results in 2 clusters with 2^{L-1} subclusters in each cluster. Note that $s = 2 \times 2^{L-1} = 2^L$, which makes the matrix size exponentially

increasing in terms of L in Algorithm **??**, but not in Algorithm **??**. If we set the global mean, c, to zero, the first factor mean, $c^{(i\cdot)}$, as $-a(\theta_d)$ or $a(\theta_d)$, the second factor mean, $c^{(\cdot j)}$, as the value of $\sum_{i=1,\, i\neq d}^{L} s_i[n]a(\theta_i)$ for each different set of subcluster labels, the cluster-wise error term, ϵ_{ij}, as zero, and the instance-wise error term, e_k, as $e[n]$, then the two data models of Eq. (**??**) and Eq. (**??**) become equivalent. Here $\epsilon_{ij} = 0$ means that there is no interaction between the first and second factors, i.e. the desired and the non-desired signals, which makes the model purely additive in terms of the mean of each factor. Thus, using the relationship between the beamforming problem and the two-way MANOVA model shown in Section 4.1, we can match the beamforming data model and h-LDA by setting $\alpha = 1$ in Eq. (**??**).

V. Computer Simulations

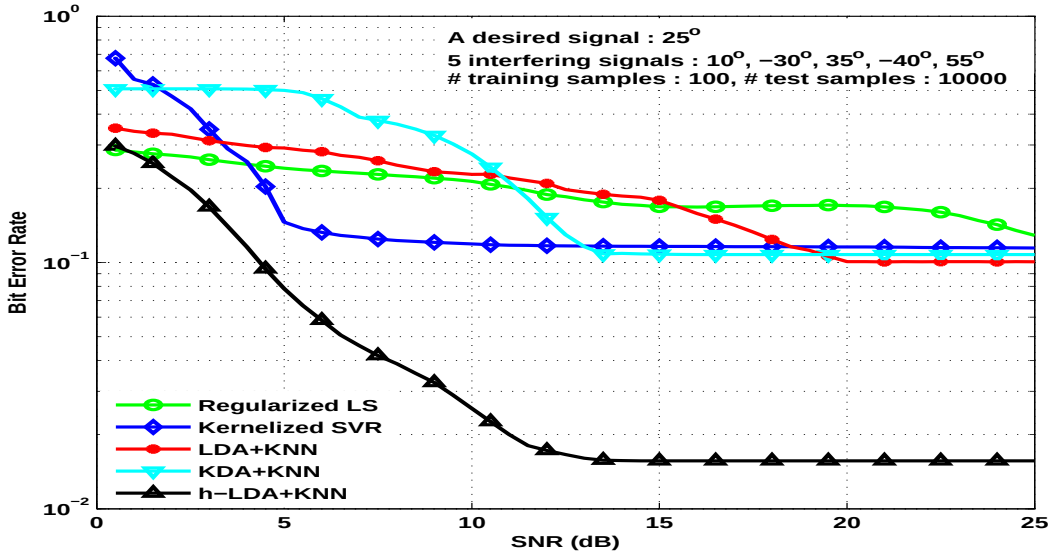
A. Experimental Setup

In order to evaluate the performance of the proposed h-LDA algorithm, various computer simulations have been carried out. All the experiments were done using Matlab on Windows XP with 1.8GHz CPU with 2.0GB memory. First we generated training and test samples according to Eq. (**??**) using binary phase-shift keying signals. Here we assigned a unit amount of signal power equally to each source but with different AOA's. We compared h-LDA with four other algorithms including two regression methods, the regularized LS and the kernelized SVR, and two supervised dimension reduction methods, LDA and kernel discriminant analysis (KDA). KDA does the same computations shown in Eq. (**??**) but in kernel space [**?**], [**?**]. To estimate the signals for unknown test data, we have used the k-nearest neighbor classifier, where $k = 1$, after the dimension reduction method was used, and the threshold comparison in Eq. (**??**) when the regression method was used. As a performance measure, bit error rates and computing times are presented.

B. Parameter Selection

Each method requires its own model parameter. For the regularized LS, we set $\gamma = 10^{-1}$ from Eq. (**??**) although it did not affect the results significantly. In the case of h-LDA, the relative weight parameter α in Eq. (**??**) was set to 1 to match the data model as discussed in Section 4.2. The kernelized SVR requires two parameters besides the kernel bandwidth, one of which is the cost parameter C that gives a weight to the sum of slack variables, and the other is the

Fig. 5. Bit error rates depending on SNR



error tolerance parameter ϵ in the ϵ-sensitive loss function. Those two parameters were chosen so that we can recover the previous results on the kernelized SVR in beamforming shown in [**?**] as

$$C = 3\sigma \text{ and } \epsilon = 3\sigma\sqrt{\frac{\ln N_{tr}}{N_{tr}}},$$

where σ^2 is the noise power, and N_{tr} is the training sample size. Such choices are somewhat heuristic, but we found these work well, and for more details, see [**?**]. The kernel used in the kernelized SVR and KDA was the widely-used Gaussian kernel, and its bandwidth parameter value was carefully chosen as 2^{-4} among values ranging from 2^{-10} to 2^{10} after many simulation trials in order to obtain the best possible results. Actually, our expeirments were shown to be very sensitive to the value of this kernel parameter, and other general schemes such as the cross-validation method did not give a reliable value since the training sample size was small (~ 100).

C. Subcluster Discovery for h-LDA

h-LDA requires the information about subcluster label values for training data as well as its label information, which are the bits sent by the desired source signal. As described in Section 4.2, each subcluster label corresponds to the various combinations of interfering signals, which may or may not be available. In order to obtain subcluster label values for interfering signals,

Fig. 6. Bit error rates depending on the training sample size

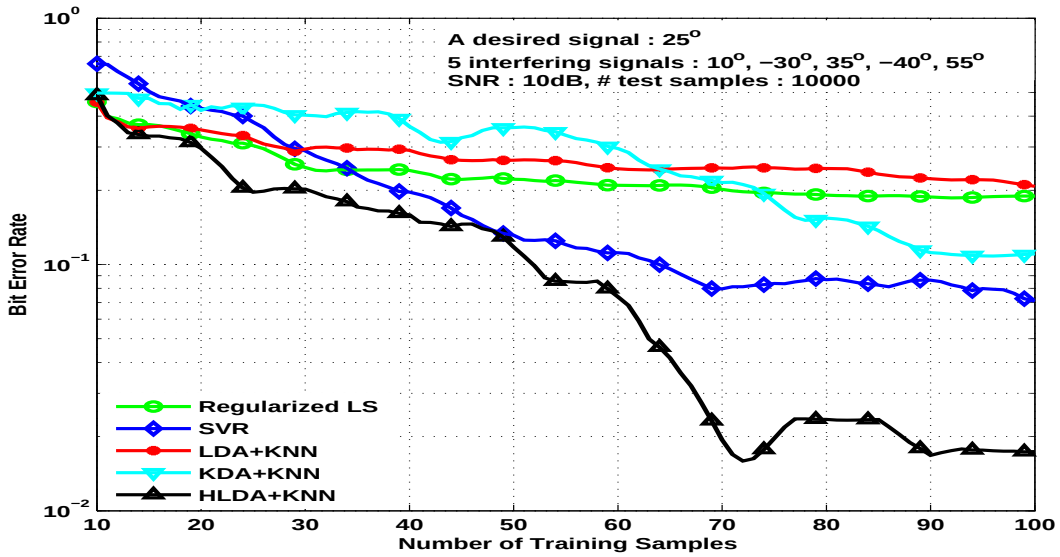


TABLE I

COMPARISON OF COMPUTING TIMES. FOR EACH CASE, THE AVERAGED COMPUTING TIMES OF 100 TRIALS WERE PRESENTED EXCEPT FOR THE KERNELIZED SVR.

# training/test # sensor/interfering signals	Phase	Regularized LS	Kernelized SVR	LDA/ GSVD	KDA/ GSVD	h-LDA/ GSVD	h-LDA/ Chol
200/1000	Training	.0005	239.5	.0086	.2619	.0222	.0154
7/4	Test	.0002	0.4756	.6987	.9484	.7013	.7005
400/3000	Training	.0011	3018	.0381	1.420	.0747	.0391
9/6	Test	.0001	1.180	1.146	1.505	1.147	1.146
600/5000	Training	.0020	67394	.1020	4.046	.1680	.0740
11/8	Test	.0001	2.028	1.622	2.432	1.618	1.626

we adopted a simple clustering algorithm, k-means clustering, and applied it to each cluster data in the training set. Comparing the results with the true subcluster labels, we could observe the value well over 0.9 for the purity measure [**?**] and close to 0 in terms of entropy measure [**?**] for most cases, which reveals the underlying subcluster structure properly. In this paper, the labels obtained by clustering were used rather than the true subcluster labels.

D. Results

Bit Error Rates: Figure **??** shows the bit error rates versus signal to noise ratio (SNR). In this simulation, we set the training and the test sample sizes to 100 and 10000 respectively, and

the AOA's of the desired and five interfering signals were 25°, and (10°, -30°, 35°, -40°, 55°), respectively. Figure **??** depicts the bit error rate performance as a function of the number of training samples. Here we fixed SNR as 10dB, but the other parameters were not changed from Figure **??** except for the training sample size that varied from 10 to 100.

From these two experiments, we can see that h-LDA consistently works best throughout almost all the different SNR values and training sample sizes, which proves the superiority and the robustness of h-LDA over both regression-based methods. These results also confirm the advantage of h-LDA over LDA that h-LDA stems from as long as a clear subcluster structure can be found. Furthermore, it is worthwhile to note that, while still being a linear model, h-LDA performs even better than nonlinear methods including kernelized SVR and KDA.

Computing Times: Table **??** compares the computing times in the training phase and the test phase for each method. The regularized LS involves only one QR decomposition followed by one upper triangular system solution in the training phase and a simple threshold comparison in the test phase, and thus its computing time is much faster than any other methods. In contrast, the kernelized SVR was shown to require significantly more time than the other methods, especially in the training phase. This is due to the quadratic optimization of SVR that requires a lot of computation. In the test phase, the kernelized SVR took more time than the regularized LS although it has a similar form of threshold comparison, which is because of intensive kernel evaluations.

Among dimension reduction methods, we can see that KDA requires the most time in both the training and the test phase because it involves kernel evaluations. Actually, KDA requires the most computing time in the test phase due to kernel evaluations between unknown test data and all training samples. In the training phase, h-LDA/GSVD was shown to be comparably slower than LDA/GSVD since h-LDA/GSVD has a larger size K^h matrix, i.e., $(p+n+s)\times m$, in Algorithm **??**, whereas LDA/GSVD has this matrix size of $(p+n)\times m$. However, h-LDA/Chol improves the computing time of h-LDA/GSVD by reducing the matrix size of K^h to $(p+m)\times m$. Although the matrix size in h-LDA/Chol is smaller than that in LDA/GSVD for oversampled cases, h-LDA involves an additional Cholesky decomposition for S_w^h in Eq. (**??**). However the advantage of adopting the Cholesky decomposition is proven to be obvious as the original data size increases as shown in Table **??**. Except for KDA, the computing times in the test phase using the k-nearest neighbor classifier are almost the same.

Overall, the above experimental results indicate that while maintaining the computational efficiency, h-LDA shows excellent classification ability in beamforming.

VI. Conclusions

In this study, we have presented an efficient algorithm and the data model of the recently proposed supervised dimension reduction method, h-LDA. Based on the GSVD technique, we have shown the improved efficiency of our algorithm by reducing the matrix size owing to the Cholesky decomposition, and identified the data model for h-LDA through the analysis between h-LDA and two-way MANOVA. We then demonstrated the successful application of h-LDA to a signal processing area, beamforming. Such an application is meaningful in that beamforming, which has primarily been viewed as a regression problem, was dealt with as a classification problem, which gives us the possibility of applying supervised dimension reduction methods, thus improving the classification ability to the relevant classes. From extensive simulations, we have proven the performance of h-LDA in terms of classification error rates and computing times. This was compared with regression methods such as the regularized LS and the kernelized SVR, and the supervised dimension reduction methods such as LDA and KDA. These results reveal many promising aspects of h-LDA. First, by demonstrating its better performance than the existing regression-based methods, we enlightened the applicability of supervised dimension reduction techniques to signal processing problems including beamforming. Second, h-LDA, in particular, confirmed its superiority over LDA for data with multi-modal structure, which represents a wider class of problems in many data analysis contexts such as signal processing and others.