$\mathcal{AWDA}$: An Adaptive Wishart Discriminant Analysis

Haoyi Xiong[1,*], Wei Cheng[2], Wenqing Hu[1], Jiang Bian[1], and Zhishan Guo[1]
[1]Missouri University Science and Technology, USA, [2]NEC Laboratories America, USA

***Abstract*—Linear Discriminant Analysis (LDA) is widely-used for supervised dimension reduction and linear classification. Classical LDA, however, suffers from the *ill-posed* estimation problem on data with high dimension and low sample size (HDLSS). To cope with this problem, in this paper, we propose an Adaptive Wishart Discriminant Analysis ($\mathcal{AWDA}$) for classification, that makes predictions in an ensemble way. Comparing to existing approaches, $\mathcal{AWDA}$ has two advantages: 1) leveraging the Wishart distribution, $\mathcal{AWDA}$ ensembles multiple LDA classifiers parameterized by the sampled covariance matrices via a *Bayesian Voting Scheme*, which theoretically improves the robustness of classification, compared to LDA classifiers using a single (probably ill-posed) covariance matrix estimator; 2) $\mathcal{AWDA}$ updates the weights for voting optimally to adapt the local information of each new input data, so as to enable the nonlinear classification. Theoretical analysis indicates that $\mathcal{AWDA}$ guarantees a close approximation to the optimal Bayesian inference and thus achieves robust performance on high dimensional data. Extensive experiments on real-world datasets show that our approach outperforms state-of-the-art algorithms by a large margin.**

***Index Terms*—Data Mining, Classification, Linear Discriminant Analysis, Bayesian Inference and Wishart Distribution**

I. Introduction

Linear Discriminant Analysis (LDA) is frequently used as one of the common performance benchmarks for linear classification problem, with respect to LDA's provable Bayesian optimality [1]. However, recent studies demonstrate the limitation of classical LDA under high dimension and low sample size (HDLSS) settings [2]. This is because it is difficult to recover the "true" parameters, e.g., (inverse) covariance matrix, from a *relatively small* number of training samples [3]. When the number of dimensions of data is larger than the number of samples, the sample covariance estimation used in classical LDA, is singular and not invertible. In this case, LDA cannot produce any valid prediction. Even when the sample size is larger than the number of dimensions, the sample (inverse) covariance estimation could be quite different from the "true" (inverse) covariance matrix, with an inconsistent estimate of the largest eigenvalues and almost-orthogonal eigenvectors to the truth [3]. Such *ill-posed estimation problem* significantly degrades the performance of LDA. Moreover, the data for practical classification problem is usually not linearly separable [4, 5]. Therefore, the linear classification sometimes cannot provide good results.

To address the ill-posed estimation problem, several regularization-based methods have been proposed to accurately estimate the (inverse) covariance matrix [6–8] or linear coefficients [9, 10] under high dimension and low sample size settings [11]. Further, to handle the non-linearity, some kernel-based or nonparameteric LDA classifiers [12–15] have been proposed. In summary, these methods intend to improve LDA classification through optimizing the parameters of LDA, such as (inverse) covariance matrices, linear projection metrics, or kernel settings, in a so-called *optimal model selection* manner [16].

Instead of "bidding" the optimal parameter in the full and usually unknown parameter space, in this work, we intend to improve LDA in an ensemble way [17], while *adapting to the new input data*. Specifically, we first sample a set of (inverse) covariance matrices from *both training data and the new input data*, then "weighted-averages" the classification results of multiple LDA classifiers parameterized by the sampled inverse covariance matrices via a *Bayesian Voting Scheme* [18]. Theoretical studies show that such Bayesian voting scheme can secure a *wider margin* and guarantee a good classification performance with a *lower generalization error bound* [18]. This theoretically guarantees that the proposed framework can "on average" outperform those regularization-based LDA classifiers using only *single* (inverse) covariance matrix estimator [19]. More importantly, the sampled (inverse) covariance matrices used by different LDA classifiers are updated with each new input data instance. In this way, the proposed classifier enables nonlinear classification by leveraging local information of the input data [20].

However, the aforementioned *Input-Adaptive Bayesian Voting Scheme* is not computationally efficient. As the sampled (inverse) covariance matrices are assumed to be updated to adapt each new input data for classification, the sampling complexity is very high. Especially, when the number of dimensions of data is high, it is quite time-consuming to sample the (inverse) covariance matrices, while ensuring each sampled matrix is positive-semidefinite. Thus, we propose a novel method, Adaptive Wishart Discriminant Analysis ($\mathcal{AWDA}$), which can approximate the optimal prediction results with minimal sampling efforts.

Specifically, $\mathcal{AWDA}$ first *surrogates* the distribution of inverse covariance matrices using a Wishart distribution estimated from the training data, then a set of inverse covariance matrices are sampled based on the distribution. The "weighted-averaged" result over the classification results from LDA classifiers parameterized by these sampled inverse covariance matrices are used for prediction. The "weights" in ensemble learners are updated by each new input data for classification optimally in a Bayesian manner. In this way, $\mathcal{AWDA}$ can approximate the aforementioned *input-adaptive Bayesian voting* schema, with proven convergence rate. Our theoretical analysis further proves that (1) the error of approximation could quickly converge with the increasing number of sampled

TABLE I: Summary of notations

Symbol	Definition
n	the number of training samples
m	the number of sampled inverse covariance matrices
p	the number of dimensions of data sample
λ	the tuning parameter of De-Sparsified Graphical Lasso
$\widehat{T}$	the scale matrix of De-Sparsified Graphical Lasso
x_i	the vector of i-th data sample (1≤i≤ n)
l_i	the label of i-th data sample ($1\leq i \leq n$, $l_i \in${-1, +1})
$\mathbf{x}$	the vector of new input data sample for classification
Θ	the inverse covariance matrix estimated ($\Theta \in \mathcal{S}^n_+$)
$\mathbb{X}_{+1}$	set of positive samples
$\mathbb{X}_{-1}$	set of negative samples
$\bar{x}$	mean value of all training samples
$\bar{x}_{+1}$	the mean value of positive training samples
$\bar{x}_{-1}$	the mean value of negative training samples
$\bar{\Sigma}_{+1}$	the covariance of positive training samples
$\bar{\Sigma}_{-1}$	the covariance of negative training samples
$\bar{\Sigma}$	the variance of all training samples
$f(\cdot)$	the classification function of Fisher's LDA
$\mathcal{W}(\cdot)$	Wishart distribution
$\mathcal{W}^{-1}(\cdot)$	Inverse Wishart distribution

inverse covariance matrices m at the speed $\mathcal{O}(m^{-\frac{1}{2}})$; and (2) the error is not sensitive to the dimensions of the data, that means the performance of high dimensional data classification could be well-guaranteed.

In the rest of the paper, we introduce the background and formulate the problem of research in Section II. In Section III, we present the proposed algorithm $\mathcal{AWDA}$. We brief the theoretical analysis on $\mathcal{AWDA}$ in Section IV. In Section V, we evaluate $\mathcal{AWDA}$ with other baseline algorithms using binary classification benchmark datasets and large-scale real-world Electronic Health Record (EHR) datasets. We review the related work, discuss the future work and finally conclude the paper in Section VI and VII.

II. Preliminaries and Problem Formulation

In this section, we first introduce the preliminaries of our research, then formulate the research problem of this paper. Important notations are listed in Table I.

A. Linear Discriminant Analysis

To solve the binary classification problem aforementioned, we consider a simple LDA classifier $f(\mathbf{x}) \in \{\pm 1\}$ based on the given p-dimensional data vector **x** and the labeled samples $x_1, x_2, ...x_n$

$$f(\mathbf{x}, \bar{x}, \bar{x}_{+1}, \bar{x}_{-1}, \Theta) = sign\left((\mathbf{x}-\bar{x})^T \,\Theta\, (\bar{x}_{+1}-\bar{x}_{-1})\right), \tag{1}$$

where the signal function $sign(\cdot)$ maps the non-negative input to $+1$ and the negative input to -1; $\bar{x}$ refers to the mean vector of all samples $x_1, x_2, ...x_n$; and $\bar{x}_{+1}$, $\bar{x}_{-1}$ refer to the mean vectors of the positive samples and negative samples receptively. Θ is the inverse covariance matrix (namely precision matrix) estimated from data sample $x_1, x_2, ...x_n$. The most common estimation of the inverse covariance matrix is the inverse of pooled sample estimation considering the two classes, i.e., $\Theta = \bar{\Sigma}^{-1}$. Thus, we write $f(x, \bar{\Sigma}^{-1})$ as the classical Fisher's Linear Discriminant Analysis.

B. Bayesian Voting Scheme

Given a binary classifier $h_\omega(\mathbf{x}) \in \{\pm 1\}$, which is parameterized by ω, the Bayesian Voting Classification [18] of the classifier is:

$$sign\left(\int_\omega h_\omega(\mathbf{x})p(\omega)d_\omega\right), \tag{2}$$

where $p(\omega)$ is the prior probability of the parameter ω. As a binary classifier, the above classifier in Eq. 2 outputs the label with the highest weighted vote. The theoretical advantages of Bayesian voting scheme are addressed in [18].

C. Problem Formulation

To handle the uncertainty of (inverse-) covariance matrix estimates for LDA, through combining Bayesian Voting and LDA, we can consider a new classifier as:

$$sign\left(\int\limits_{\Theta\geq 0} f(\mathbf{x}, \bar{x}, \bar{x}_{+1}, \bar{x}_{-1}, \Theta)P(\Theta|x_1, x_2, ...x_n, \mathbf{x})d\Theta\right), \tag{3}$$

where $P(\Theta|x_1, x_2, ...x_n, \mathbf{x})$ is the probability of the inverse covariance matrix $\hat{\Theta}$, given the n training samples $x_1, x_2, ...x_n$ as well as the new sample for prediction **x**. In our research, we named this pattern as *Input Adaptive Bayesian Voting*. Note that we take the new input vector x into account for generating the "hypothesis" Θ of Bayesian inference.

With all above backgrounds and settings in mind, the problem of this research is to compute Eq. 3. However, there exist at least two major technical challenges:

Challenge I: Fast Computation and Lazy Sampling - To compute the integral in Eq. 3, a common solution is to leverage a Monte-Carlo Integration algorithm [21] that first randomly samples a group of positive-semidefinite matrices e.g., $\Theta_1, \Theta_2 \ldots \Theta_m$ from the distribution with probability density function $P(\Theta|x_1, x_2, ...x_n, \mathbf{x})$, then averages $f(\mathbf{x}, \bar{x}, \bar{x}_{+1}, \bar{x}_{-1}, \Theta)$ over the sampled positive-semidefinite matrices as $\frac{1}{m} \cdot \sum_{i=1}^m \cdot f(\mathbf{x}, \bar{x}, \bar{x}_{+1}, \bar{x}_{-1}, \Theta_i)$. This method can give an approximate result of Eq. 3. However, the density function of the sampled positive-semidefinite matrices $P(\Theta|x_1, x_2, ...x_n, \mathbf{x})$ depends on the input **x**. That means, for each new testing sample **x**, we have to build a new probability distribution based on $P(\Theta|x_1, x_2, ...x_n, \mathbf{x})$, then sample a new group of positive-semidefinite matrices and run the Monte-Carlo Integration accordingly. Obviously, the computational cost to re-sample a new group of positive-semidefinite matrices for each new input **x** is high. Thus, we need a *"Lazy Sampling"* mechanism, which only samples a group of positive-semidefinite matrices once, then uses the same group of matrices for arbitrary input **x**.

Challenge II: Approximation and Sampling Complexity - The accuracy of classification highly depends on whether the proposed algorithm can approximate the Eq. 3 as well as the sampling complexity (i.e., how many sampled (inverse) covariance matrices are used in the inference). For the high-dimensional numeric integration [22], the approximation is usually bottle-necked by the number of dimensions (e.g., the number of data dimensions is p, the number of dimensions of positive-semidefinite matrices $p \times p$) and the sampling complexity (e.g., the number of sampled positive-semidefinite matrices m). Intuitively, the convergence of algorithms can be improved, with increasing sampling complexity and lower dimensionality. However, we aim at proposing an algorithm to approximate Eq. 3 with a low computational/sampling complexity while ensuring a close approximation. Especially, we require an approximation rate that is not sensitive to the dimensionality of the data sample (p), so as to enable the high dimensional data classification.

In the rest of the paper, we present a novel classifier, *Adaptive Wishart Discriminant Analysis – $\mathcal{AWDA}$*, which tackles the two research challenges, with low computational/sampling complexity and proven dimensionality-insensitive approximation rate.

III. $\mathcal{AWDA}$: THE ADAPTIVE WISHART DISCRIMINANT ANALYSIS

In this section, we introduce our solution to compute Eq. 3 as follows. We first re-formulate Eq. 3. Then, we introduce the algorithm of $\mathcal{AWDA}$ to compute the reformulation of Eq. 3.

A. Problem Reformulation

We first define $P(\mathbf{x}|\Theta)$ as the probability of input vector $\mathbf{x}$ given the inverse covariance matrix Θ, and $P(\Theta|x_1, x_2...x_n)$ as the probability of the inverse covariance matrix Θ, given the training samples $x_1, x_2...x_n$. Then, we define a function:

$$g(\mathbf{x}) = \int_{\Theta \geq 0} f(\mathbf{x}, \bar{x}, \bar{x}_{+1}, \bar{x}_{-1}, \Theta) P(x|\Theta) \cdot P(\Theta|x_1, x_2...x_n)\, d\Theta. \tag{4}$$

Theorem 1. *Eq. 3 is equivalent to the classification result of $sign(g(\boldsymbol{x}))$.*

Proof. Assuming $x_1, x_2, ...x_n,$ and $\mathbf{x}$ are drawn *i.i.d.* from an unknown distribution, according to the Bayesian theorem, we decompose $P(\Theta|x_1, ...x_n, \mathbf{x})$ as

$$\begin{aligned} P(\Theta|x_1, ...x_n, \mathbf{x}) &= \frac{P(\mathbf{x}|\Theta)P(x_1...x_n|\Theta)P(\Theta)}{P(\mathbf{x})P(x_1...x_n)} \\ &= P(\mathbf{x}|\Theta)P(\Theta|x_1, x_2...x_n) \cdot P(\mathbf{x})^{-1}. \end{aligned} \tag{5}$$

Thus, Eq. 3 can be re-written as $sign(p(\mathbf{x})^{-1}\, g(\mathbf{x}))$. As $p(\mathbf{x})^{-1}$ is positive for $\forall \mathbf{x}$. As a result, we can conclude $sign(g(\mathbf{x})) = sign(p(\mathbf{x})^{-1} \cdot g(\mathbf{x}))$ is consistently equivalent to Eq. 3. □

In this way, the key of proposed research is to compute Eq. 4. We propose a straightforward method, $\mathcal{AWDA}$: the algorithm consists of a probabilistic model that can generate m sampled inverse covariance matrices according to the density function $P(\Theta|x_1, x_2...x_n)$, then calculate Eq. 3 through Monte-Carlo Integration using the sampled inverse covariance matrices. The design of $\mathcal{AWDA}$ is described in the following.

B. $\mathcal{AWDA}$ Framework Design

Given the reformulation of the problem, we design the algorithm of $\mathcal{AWDA}$ that consists of the following two phases:

- *Training Phase* - Given the labeled data pairs $\langle x_1, l_1\rangle$, $\langle x_2, l_2\rangle \ldots \langle x_n, l_n\rangle$ for training, this phase outputs the model of $\mathcal{AWDA}$ classifier as a group of sampled inverse covariance matrices and the mean vectors ($\Theta_1 \ldots \Theta_m$, $\bar{x}$, $\bar{x}_{+1}$, $\bar{x}_{-1}$) drawn from a Wishart distribution. The pseudo code is given in Algorithm 1. Note that a tuning parameter λ is used here for the Wishart distribution modeling.
- *Testing Phase* - Given a new data vector $\mathbf{x}$ for classification, the sampled matrix $\Theta_1 \ldots \Theta_m$ and the estimated means $\bar{x}$, $\bar{x}_{+1}$, $\bar{x}_{-1}$, the algorithm outputs the classification result e.g., ± 1 through Bayesian inference. The pseudo code is given in Algorithm 2.

Note that, through the problem reformulation addressed in **Theorem 1**, $\mathcal{AWDA}$ is expected to classify any data vector $\mathbf{x}$ (i.e., approximate Eq. 3 with arbitrary $\mathbf{x}$) using only one group of sampled inverse covariance matrices. The design and implementation of the key parts of $\mathcal{AWDA}$ is addressed in following sections.

Algorithm 1 Training Phase of $\mathcal{AWDA}$

```
 1: procedure $\mathcal{AWDA}$_TRAIN($\langle x_1, l_1\rangle \ldots \langle x_n, l_n\rangle$, $\lambda$, $p$, $m$)
 2: /* Step I: Pooled Sample Estimation */
 3:     $\mathbb{X}_{+1} \leftarrow$ set of positive samples$\langle x_1, l_1\rangle \ldots \langle x_n, l_n\rangle$
 4:     $\mathbb{X}_{-1} \leftarrow$ set of negative samples$\langle x_1, l_1\rangle \ldots \langle x_n, l_n\rangle$
 5:     $\bar{x} \leftarrow \frac{1}{n}\sum_{i=1}^{n} x_i$;
 6:     $\bar{x}_{+1} \leftarrow \frac{1}{|\mathbb{X}_{+1}|}\sum_{\forall x_i \in \mathbb{X}_{+1}} x_i$;
 7:     $\bar{x}_{-1} \leftarrow \frac{1}{|\mathbb{X}_{-1}|}\sum_{\forall x_i \in \mathbb{X}_{-1}} x_i$;
 8:     $\bar{\Sigma}_{+1} \leftarrow \frac{1}{|\mathbb{X}_{+1}|}\sum_{\forall x_i \in \mathbb{X}_{+1}} (x_i - \bar{x}_{+1})(x_i - \bar{x}_{+1})^T$;
 9:     $\bar{\Sigma}_{-1} \leftarrow \frac{1}{|\mathbb{X}_{-1}|}\sum_{\forall x_i \in \mathbb{X}_{-1}} (x_i - \bar{x}_{+1})(x_i - \bar{x}_{-1})^T$;
10:     $\bar{\Sigma} = \frac{|\mathbb{X}_{+1}|}{n}\bar{\Sigma}_{+1} + \frac{|\mathbb{X}_{-1}|}{n}\bar{\Sigma}_{-1}$;
11: /* Step II: Wishart Distribution Modeling and Sampling */
    /*Using De-Sparsified Graphical Lasso [23]*/
12:     $\widehat{T} \leftarrow$ DSGLasso($\bar{\Sigma}, \lambda$);
13:     $v \leftarrow max\{n, p\}$;
14:     Build Wishart distribution $\mathcal{W}(\widehat{T}, v)$;
15: For i = 1 to m
16:     $\Sigma_i \leftarrow$ draw a random sample on $\mathcal{W}^{-1}(\widehat{T}^{-1}, v)$;
17:     $\Theta_i \leftarrow \Sigma_i^{-1}$;
18: End
19:     return ($\Theta_1 \ldots \Theta_m$, $\bar{x}$, $\bar{x}_{+1}$, $\bar{x}_{-1}$).
```

C. $\mathcal{AWDA}$_TRAIN: Training Phase of $\mathcal{AWDA}$

As shown in Algorithm 1, given the labeled data pairs $\langle x_1, l_1\rangle, \langle x_2, l_2\rangle \ldots \langle x_n, l_n\rangle$ for training, the *Training Phase* builds the $\mathcal{AWDA}$ model with a set of sampled inverse covariance matrices $\Theta_1, \Theta_2, \ldots \Theta_m$ and the estimation of mean vectors $\bar{x}$, $\bar{x}_{+1}$, $\bar{x}_{-1}$. While one can straightforwardly estimate the mean vectors $\bar{x}, \bar{x}_{+1}, \bar{x}_{-1}$ with the sample estimation using lines 3-7 in Algorithm 1, the way to sample inverse covariance matrices from the training data is still challenging.

Algorithm 2 Testing Phase of $\mathcal{AWDA}$

```
1: procedure $\mathcal{AWDA}$_TEST($\mathbf{x}$, $(\Theta_1 \ldots \Theta_m, \bar{x}, \bar{x}_{+1}, \bar{x}_{-1})$)
2: /*Binary Classification as Bayesian Inference*/
3:     $\bar{g} \leftarrow 0$;
4: For i = 1 to m
5:     $\bar{g} \leftarrow \bar{g} + \frac{1}{m} \cdot f(\mathbf{x}, \bar{x}, \bar{x}_{+1}, \bar{x}_{-1}, \Theta_i) \cdot P(\mathbf{x}|\Theta_i)$;
   /*the probability $P(\mathbf{x}|\Theta_i)$ is addressed in Eq. 10*/
6: End
7:     return sign($\bar{g}$).
```

In our research, to sample inverse covariance matrices Θ_1, $\Theta_2, \ldots \Theta_m$ according to $P(\Sigma|x_1, x_2...x_n)$, $\mathcal{AWDA}$ leverages a Wishart Distribution [24] namely $\mathcal{W}(\widehat{T}, v)$, where $\widehat{T}$ refers to the positive-definite *scale* matrix for the Wishart distribution and v is the degree of freedom. In following paragraphs, we introduce the detailed design of the *Training Phase*.

1) Sample Estimation for Means and Covariance Matrices: Given the the labeled data pairs $\langle x_1, l_1\rangle, \langle x_2, l_2\rangle \ldots \langle x_n, l_n\rangle$, $\mathcal{AWDA}$ first estimates the pooled sample covariance matrix $\bar{\Sigma}$ using the code listed in lines 3-10 in Algorithm 1. Specifically, the algorithm first sorts all samples for training into two groups – positive samples and negative samples, then estimate the sample mean and sample covariance matrices separately using the data in these two groups. Further, the algorithm pools the two sample covariance matrices as a whole, with respect to the frequencies of the two groups.

2) Scale Matrix Estimation: Given the pooled sample covariance matrix estimation $\bar{\Sigma}$, to advance the estimation accuracy, the algorithm estimates the scale matrix $\widehat{T}$ using De-Biased Graphical Lasso [23]. The calculation of De-Sparsified Graphical Lasso (denoted by `DSGlasso`$(\bar{\Sigma}, \lambda)$ in line 12 of Algorithm 1), based on the pooled sample covariance matrix $\bar{\Sigma}$ and the given tuning parameter λ, is in the following.

$$\widehat{T} = 2\widehat{\Theta}_{(\lambda)} - \widehat{\Theta}_{(\lambda)}\bar{\Sigma}\widehat{\Theta}_{(\lambda)}, \tag{6}$$

where $\widehat{\Theta}_{(\lambda)}$ refers to the Graphical Lasso estimator:

$$\widehat{\Theta}_{(\lambda)} = \operatorname*{argmin}_{\Theta \geq 0} \left(\operatorname{tr}(\bar{\Sigma}\Theta) - \log|\Theta| + \lambda \sum_{j \neq k} |\Theta_{jk}| \right), \tag{7}$$

where $\sum_{j \neq k} |\Theta_{jk}|$ refers to the sum of absolute value of the non-diagonal elements in matrix Θ.

3) Wishart Distribution Probabilistic Modeling: Based on the scale matrix $\widehat{T}$ estimated in the last step, $\mathcal{AWDA}$ builds the Wishart Distribution $\mathcal{W}(\widehat{T}, v)$. The density function of $\mathcal{W}(\widehat{T}, v)$ is defined as follows: given any $p \times p$ positive definite matrix Θ (as the inverse of potential covariance matrix), the probability density of Θ on Wishart Distribution $\mathcal{W}(\widehat{T}, v)$ is:

$$P_w(\Theta|\widehat{T}, v) = \frac{1}{2^{vp/2}\left|\widehat{\mathbf{T}}\right|^{v/2}\Gamma_p\left(\frac{v}{2}\right)} |\mathbf{\Theta}|^{(v-p-1)/2} e^{-(1/2)\operatorname{tr}(\widehat{\mathbf{T}}^{-1}\mathbf{\Theta})}, \tag{8}$$

where $|\cdot|$ refers to the determinant and the multivariate gamma function is defined as:

$$\Gamma_p\left(\frac{v}{2}\right) = \pi^{p(p-1)/4} \prod_{j=1}^{p} \Gamma\left(\frac{v}{2} + \frac{1-j}{2}\right).$$

In our research, we set the degree of freedom v using the maximum of the training dataset size n and the dimensionality of data p (i.e., $v = max\{n, p\}$).

4) Wishart Distribution Sampling: Based on the typical Inverse-Wishart Sampling Algorithm [25], $\mathcal{AWDA}$ first builds an Inverse-Wishart Distribution $\mathcal{W}^{-1}(\widehat{T}^{-1}, v)$, which is the conjugate prior for the covariance matrix of a multivariate Gaussian distribution. Then the algorithm randomly generates m covariance matrices $\Sigma_1, \Sigma_2...\Sigma_m$ drawn from the Inverse-Wishart Distribution $\mathcal{W}^{-1}(\widehat{T}^{-1}, v)$. Finally, the algorithm estimates the inverse of $\Sigma_1, \Sigma_2...\Sigma_m$ as the sampled inverse covariance matrices $\Theta_1, \Theta_2...\Theta_m$. The design of this algorithm is listed in lines 14-17 in Algorithm 1.

D. $\mathcal{AWDA}$_TEST: Testing Phase of $\mathcal{AWDA}$

Given a new data vector $\mathbf{x}$ for classification, and the trained $\mathcal{AWDA}$ model with sampled inverse covariance matrices $\Theta_1, \Theta_2...\Theta_m$ and the mean vectors $\bar{x}$, $\bar{x}_{+1}$, $\bar{x}_{-1}$, the *Testing* Phase outputs the classification result, via the approximation to Eq. 3, as:

$$\bar{g}(\mathbf{x}) = \frac{1}{m} \sum_{1 \leq i \leq m} \left(f(\mathbf{x}, \bar{x}, \bar{x}_{+1}, \bar{x}_{-1}, \Theta_i) P(x|\Theta_i) \right), \tag{9}$$

where $P(\mathbf{x}|\Theta_i)$ refers to the probability of the input vector $\mathbf{x}$ given the inverse covariance matrix Θ_i. In this paper, we characterize the probability as:

$$P(\mathbf{x}|\Theta_i) = \frac{1}{\sqrt{2\pi|\mathbf{\Theta_i^{-1}}|}} e^{-\frac{1}{2}(\mathbf{x}-\bar{\boldsymbol{x}})^T \mathbf{\Theta_i} (\mathbf{x}-\bar{\boldsymbol{x}})}, \tag{10}$$

Finally, as addressed in Algorithm 2, the algorithm $\mathcal{AWDA}$ uses $sign(\bar{g}(\mathbf{x}))$ as the classification result.

The performance analysis of the proposed algorithm based on Eq. 9 to approximating the formulated problem expressed in Eq. 3 will be addressed in the following section.

IV. Approximation Analysis

In this section, we present how close the approximation is that $\bar{g}(\mathbf{x})$ in $\mathcal{AWDA}$ can be used to approximate the re-formulated problem $g(\mathbf{x})$.

First of all, considering the fast convergence rate of De-Sparsified Graphical Lasso [23] i.e., $||\widehat{T} - \Theta^*||_\infty = \mathcal{O}_p(\sqrt{log\ p\ /n})$, with a fixed number of dimensions p and an increasing number of samples n, we are more confident to follow an assumption frequently made in many of previous Bayesian inference studies [26–28]:

Assumption 1. *For any positive-semidefinite matrix Θ i.e., $\forall\Theta \geq 0$, there exists $P(\Theta|x_1, x_2...x_n) = P_w(\Theta|\widehat{T}, v)$, where $P_w(\Theta|\widehat{T}, v)$ refers to the Wishart probability of Θ based on the mean positive-semidefinite matrix $\widehat{T}$ and $v = n-1$. $\widehat{T}$ is an estimate of inverse covariance matrix on samples $x_1, x_2...x_n$.*

With Assumption 1, we can substitute $P(\Theta|x_1, x_2...x_n)$ with $P_w(\Theta|\widehat{T}, v)$ i.e., the conjugate prior of inverse covariance matrix based on Wishart Distribution, to enable the Bayesian inference.

Theorem 2. *Under Assumption 1, for any $\eta > 0$ sufficiently small, with the increasing number of sampled inverse covariance matrices m, our algorithm $\bar{g}(\boldsymbol{x})$ converges to $g(\boldsymbol{x})$ with convergence rate $|g(\boldsymbol{x}) - \bar{g}(\boldsymbol{x})| \leq \mathcal{O}_p(\sqrt{-\log(\eta/2)/2m})$ with probability at least $1-\eta$.*

Proof. Sampled inverse covariance matrices $\Theta_1, \Theta_2, ..., \Theta_m$ are i.i.d. and all drawn from the Wishart distribution $\mathcal{W}(\widehat{T}, v)$ with probability density function $P_w(\Theta|\widehat{T}, v)$. By the classical Law of Large Numbers we know that as $m \to \infty$ we have

$$\begin{aligned}\lim_{m\to\infty} \bar{g}(\mathbf{x}) &= \int\limits_{\Theta\geq 0} f(\mathbf{x}, \bar{x}, \bar{x}_{+1}, \bar{x}_{-1}, \Theta) P(x|\Theta) P_w(\Theta|\widehat{T}, v) d_\Theta \\ &= g(\mathbf{x}),\end{aligned}$$

under Assumption 1.

Let δ^2 be the variance of $f(\mathbf{x}|\Theta)P(\mathbf{x}|\Theta)$ under the distribution given by $P_w(\Theta|\widehat{T}, v)$, so that

$$\begin{aligned}\delta^2 &= \mathbf{Var}_w f(\mathbf{x}|\Theta)P(\mathbf{x}|\Theta) \\ &= \int_{\Theta\geq 0} \left(f(\mathbf{x},\Theta)P(\mathbf{x}|\Theta) - g(\mathbf{x})\right)^2 P_w(\Theta|\widehat{T}, v) d_\Theta \ .\end{aligned}$$

By the Central Limit Theorem we know that for any $\gamma > 0$ we have

$$\lim_{m\to\infty} P_w\left(|\bar{g}(\mathbf{x}) - g(\mathbf{x})| \leq \gamma\frac{\delta}{\sqrt{m}}\right) = \frac{1}{\sqrt{2\pi}}\int_{-\gamma}^{\gamma} e^{-t^2/2} dt \ .$$

Moreover, based on Hoeffding's inequality [Hoeffding, 1963], we can conclude that for any $\eta > 0$ sufficiently small, as m is large, with probability at least $1-\eta$, we have

$$|g(\mathbf{x}) - \bar{g}(\mathbf{x})| \leq \sqrt{-\frac{1}{2m}\cdot\log\left(\frac{\eta}{2}\right)} \ .$$

□

Based on **Theorem. 2**, we can conclude that the classification result of $sign(\bar{g}(\mathbf{x}))$ should be equivalent to Eq. 3, when the number of sampled inverse covariance matrices m is large. Our later experiments show that, with more than 100 sampled inverse covariance matrices $m \geq 100$, $\mathcal{AWDA}$ can deliver decent performance and consistently outperform baseline algorithms, including SVM, Kernel SVM, Random Forest and AdaBoost.

V. Evaluation

In this section, we report the evaluation results on $\mathcal{AWDA}$ using two sets of experiments. We first introduce the experimental results based on four benchmark datasets for binary classification, where we compare the performance of $\mathcal{AWDA}$ with existing LDA algorithms and other models. Then we present the experimental results based on a real-world medical application, where we further demonstrate the advantages of the proposed algorithm using several case studies.

A. Experiment I: Evaluation on Benchmark Datasets

In this experiment, we use six binary classification benchmark datasets [29] to test the performance of $\mathcal{AWDA}$. To validate the superiority of $\mathcal{AWDA}$ over classical LDA, we use six baseline approaches for comparison: LDA (with pseudo inverse [30]), Two-stage LDA [31, 32], Linear Support Vector Machine (SVM-Linear), Gaussian Kernel SVM (SVM-G), Decision Tree (D-Three) and AdaBoost. To demonstrate the effectiveness of our method, we compare our method with baseline algorithms using two metrics: *Accuracy* and *F1-Score*.

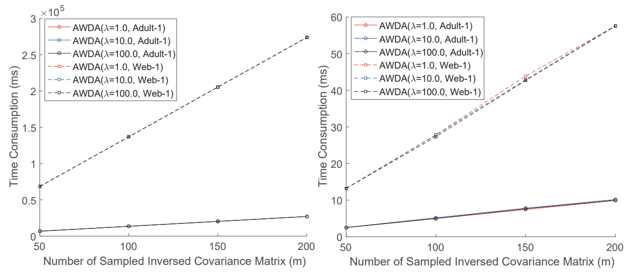
Specifically, in order to simulate the HDLSS settings ($p > m$), we set the size of training samples fixed at 50×2 with 200×2 testing samples, while the numbers of dimensions p are $p = 123$ (for Adult-1, Adult-2 and Adult-3) and $p = 300$ (Web-1,Web-2 and Web-3). Specifically, we compare $\mathcal{AWDA}$ with all six baseline algorithms, and repeat 20 times for each setting. The experimental settings show that $\mathcal{AWDA}$ is always one of the best classifiers for all datasets in all settings, and consistently outperforms other algorithms.

1) Overall Comparison: Table II and Table III present the experiment results of $\mathcal{AWDA}$ with six baselines on Adult and Web benchmark datasets. For both accuracy and F1 metrics, on a wide range of parameters (m: number of the sampled inverse covariance matrices, and λ: the regularization parameter), $\mathcal{AWDA}$ performs as one of the best classifiers among all these methods. Specifically, $\mathcal{AWDA}$ outperforms the rest six baselines with the highest accuracy and highest F1-score in the experiments with Adult-1, Adult-2, and Web-1 datasets. In the experiments with Adult-3, Two-staged LDA [31, 32] outperforms $\mathcal{AWDA}$in terms of both accuracy and F1-score, while $\mathcal{AWDA}$ ranks at the 2^{nd} and 3^{rd} places with different parameters. For Web-2 and Web-3 datasets, $\mathcal{AWDA}$ outperforms all other methods with the highest accuracy, while the linear SVM delivers slightly higher F1-score. All in all, we can conclude $\mathcal{AWDA}$ a decent binary classifier under HDLSS settings (i.e., $p = 123 > n = 50 \times 2$ for Adult benchmarks, and $p = 300 \gg n = 50 \times 2$ for Web benchmarks).

2) Performance of Input-Adaptive Mechanism: To breakdown the performance improvement caused by “Input Adaption” mechanism of $\mathcal{AWDA}$, we propose a baseline algorithm — “Averaged LDA Classifier” derived from $\mathcal{AWDA}$ that gives unique weight to the output (± 1) of each LDA classifier, based on the sampled inverse covariance matrices, and returns the sign of averaged output. Table IV presents the performance of this baseline algorithm based on Adult and Web datasets. In Table IV, the algorithm is evaluated using the same settings as Table V and Table III. Due to the page limitation, we only present the results of the baseline with fine-tuned parameters. Obviously, $\mathcal{AWDA}$ algorithm significantly outperforms the Average LDA, while the performance of Averaged LDA is more consistent. Actually the standard deviation of both accuracy and F1-score for Average LDA is around 0.0002, which is much smaller than the algorithms listed in Table V and Table III. Comparing Average LDA to $\mathcal{AWDA}$, the Averaged LDA performs more stably, as it averages the

TABLE II: Performance Comparison on Adult 1, Adult 2, and Adult 3 Benchmarks (50×2 Training Samples, 200×2 Testing Samples), where "ACC." refers to accuracy and "F1." refers to F1-Score. Note: numbers in **Red** refer to the largest values, **Blue** refers to the second-largest values, and **Orange** refers to the third-largest values.

	Data Source					
	Adult-1		Adult-2		Adult-3	
Parameter	ACC.	F1.	ACC.	F1.	ACC.	F1.
Depth	Decision Tree (D-Tree)					
10.0	0.713 ± 0.071	0.705 ± 0.074	0.717 ± 0.104	0.712 ± 0.116	0.721 ± 0.050	0.707 ± 0.066
20.0	0.686 ± 0.058	0.680 ± 0.080	0.696 ± 0.039	0.690 ± 0.054	0.698 ± 0.035	0.689 ± 0.044
	LDA					
	0.717 ± 0.056	0.710 ± 0.066	0.730 ± 0.062	0.719 ± 0.070	0.727 ± 0.052	0.722 ± 0.059
	Two-Stage LDA					
	0.757 ± 0.038	0.745 ± 0.034	0.761 ± 0.029	0.748 ± 0.037	**0.772** ± 0.055	**0.765** ± 0.058
(m, λ)	$\mathcal{AWDA}$ (*The Proposed Algorithm*)					
50, 1.0	**0.771** ± 0.023	**0.761** ± 0.028	**0.777** ± 0.021	**0.768** ± 0.020	**0.772** ± 0.022	**0.762** ± 0.027
50, 10.0	**0.769** ± 0.022	0.758 ± 0.027	**0.771** ± 0.021	**0.759** ± 0.021	**0.771** ± 0.024	0.759 ± 0.028
50, 100.0	0.768 ± 0.024	0.756 ± 0.029	0.763 ± 0.044	0.748 ± 0.041	0.768 ± 0.021	0.756 ± 0.021
100, 1.0	**0.771** ± 0.023	**0.767** ± 0.021	**0.776** ± 0.021	0.752 ± 0.034	**0.773** ± 0.022	**0.763** ± 0.027
100, 10.0	0.768 ± 0.021	0.757 ± 0.026	0.770 ± 0.020	0.758 ± 0.020	**0.771** ± 0.025	0.760 ± 0.029
100, 100.0	0.768 ± 0.024	0.755 ± 0.029	0.769 ± 0.021	0.757 ± 0.022	0.769 ± 0.027	0.758 ± 0.030
150, 1.0	**0.770** ± 0.022	**0.760** ± 0.028	**0.776** ± 0.021	**0.767** ± 0.021	**0.772** ± 0.022	0.761 ± 0.027
150, 10.0	0.768 ± 0.022	0.756 ± 0.027	**0.771** ± 0.020	**0.759** ± 0.020	0.770 ± 0.025	0.759 ± 0.029
150, 100.0	0.768 ± 0.024	0.756 ± 0.029	0.769 ± 0.021	0.757 ± 0.022	0.769 ± 0.027	0.758 ± 0.030
200, 1.0	**0.770** ± 0.023	**0.760** ± 0.027	**0.776** ± 0.020	**0.767** ± 0.020	**0.773** ± 0.023	**0.762** ± 0.027
200, 10.0	**0.769** ± 0.022	0.757 ± 0.028	**0.771** ± 0.020	**0.759** ± 0.020	**0.771** ± 0.025	0.760 ± 0.038
200, 100.0	0.768 ± 0.024	0.756 ± 0.028	0.769 ± 0.021	0.757 ± 0.021	0.769 ± 0.026	0.758 ± 0.030
	SVM-Linear (*Fine-tuned Parameter*)					
	0.748 ± 0.051	0.739 ± 0.066	0.759 ± 0.044	0.748 ± 0.047	0.751 ± 0.024	0.744 ± 0.037
Bandwidth	SVM-Kernal (Gaussian)					
0.1	0.569 ± 0.211	0.484 ± 0.363	0.578 ± 0.227	0.490 ± 0.319	0.541 ± 0.004	0.471 ± 0.233
1.0	0.657 ± 0.084	0.716 ± 0.078	0.670 ±0.121	0.643 ± 0.154	0.662 ± 0.110	0.652 ± 0.100
# of Classifiers	AdaBoost					
100.0	0.716 ± 0.069	0.674 ± 0.093	0.731 ±0.064	0.681 ± 0.095	0.743 ± 0.050	0.696 ± 0.075
200.0	0.728 ± 0.057	0.678 ± 0.089	0.733 ± 0.069	0.681 ± 0.100	0.737 ± 0.056	0.686 ± 0.085



(a) Training (b) Testing

Fig. 1: Time Consumption of $\mathcal{AWDA}$ with Varying m (the number of the sampled inverse covariance matrices) on Adult-1 and Web-1 Datasets

outputs of multiple LDA classifiers [17]. On the other hand, the proposed algorithm $\mathcal{AWDA}$ performs better with higher accuracy and F1-score, due to the input adaption mechanism used in $\mathcal{AWDA}$.

3) Time Consumption Comparison: Figure. 1 demonstrates the training and testing time consumption of $\mathcal{AWDA}$ on Adult-1 and Web 1 datasets. In the figure, we report the time consumption on varying numbers of sampled inverse covariance matrices (i.e., m). Specifically, we report the time consumption for training (average time consumption to train a classifier based on $n = 50 \times 2$ training samples, in the 20 experiments), and testing (average time consumption to classify one sample, in the 20 experiments and each experiment with 200×2 testing samples). The y-axis is set to milliseconds. Three solid lines in each figure refers to $\mathcal{AWDA}$ on Adult-1 datasets with the three λ settings, while the dash lines refer to the results on Web-1 datasets. It is obvious that, while the difference between various λ is insignificant, the time consumption for both training and testing growth linearly with the increasing m (the number of sampled inverse covariance matrices).

Compared to the time consumption of baseline algorithms evaluated in the same settings (reported in Table V), on all m settings (from 50 to 200), $\mathcal{AWDA}$ consumes significantly longer time in both training and testing. However, we believe such time consumption (especially for testing) is still tolerable

TABLE III: Performance Comparison on Web 1, Web 2 and Web 3 Benchmarks (50×2 Training Samples, 200×2 Testing Samples), where "ACC." refers to accuracy and "F1." refers to F1-Score. Note: numbers in **Red** refer to the largest values, **Blue** refers to the second-largest values, and **Orange** refers to the third-largest values.

	Data Source					
	Web-1		Web-2		Web-3	
Parameter	ACC.	F1.	ACC.	F1.	ACC.	F1.
Depth	Decision Tree (D-Tree)					
10.0	0.669 ± 0.111	0.611 ± 0.239	0.645 ± 0.068	0.550 ± 0.203	0.639 ± 0.051	0.536 ± 0.201
20.0	0.707 ± 0.046	0.692 ± 0.076	0.682 ± 0.075	0.649 ± 0.134	0.678 ± 0.072	0.630 ± 0.103
	LDA					
	0.774 ± 0.066	0.781 ± 0.064	0.704 ± 0.091	0.695 ± 0.090	0.702 ± 0.085	0.697 ± 0.103
	Two-Stage LDA					
	0.778 ± 0.080	0.757 ± 0.104	0.759 ± 0.084	0.732 ± 0.104	0.762 ± 0.051	0.733 ± 0.081
(m, λ)	$\mathcal{AWDA}$ (*The Proposed Algorithm*)					
50, 1.0	**0.836** ± 0.023	**0.836** ± 0.029	**0.837** ± 0.033	**0.834** ± 0.038	**0.839** ± 0.024	**0.837** ± 0.033
50, 10.0	0.794 ± 0.042	0.781 ± 0.068	0.787 ± 0.050	0.768 ± 0.064	0.781 ± 0.036	0.761 ± 0.058
50, 100.0	0.784 ± 0.043	0.768 ± 0.069	0.773 ± 0.052	0.749 ± 0.068	0.766 ± 0.041	0.741 ± 0.065
100, 1.0	**0.837** ± 0.022	**0.837** ± 0.028	**0.838** ± 0.033	**0.834** ± 0.037	**0.839** ± 0.024	**0.837** ± 0.033
100, 10.0	0.796 ± 0.041	0.782 ± 0.067	0.787 ± 0.051	0.768 ± 0.065	0.781 ± 0.038	0.761 ± 0.059
100, 100.0	0.782 ± 0.043	0.766 ± 0.070	0.774 ± 0.052	0.752 ± 0.068	0.766 ± 0.042	0.741 ± 0.066
150, 1.0	**0.837** ± 0.023	**0.838** ± 0.028	**0.838** ± 0.033	**0.835** ± 0.037	**0.839** ± 0.021	**0.837** ± 0.029
150, 10.0	0.795 ± 0.041	0.782 ± 0.067	0.787 ± 0.050	0.768 ± 0.064	0.780 ± 0.037	0.761 ± 0.059
150, 100.0	0.782 ± 0.043	0.765 ± 0.070	0.774 ± 0.053	0.751 ± 0.069	0.766 ± 0.041	0.742 ± 0.065
200, 1.0	**0.838** ± 0.024	**0.838** ± 0.029	**0.837** ± 0.032	**0.834** ± 0.037	**0.838** ± 0.022	**0.836** ± 0.030
200, 10.0	0.795 ± 0.042	0.782 ± 0.067	0.786 ± 0.050	0.768 ± 0.064	0.780 ± 0.037	0.760 ± 0.059
200, 100.0	0.782 ± 0.044	0.766 ± 0.070	0.774 ± 0.052	0.751 ± 0.068	0.767 ± 0.041	0.742 ± 0.065
	SVM-Linear (*Fine-tuned Parameter*)					
	0.728 ± 0.036	0.689 ± 0.042	**0.800** ± 0.055	**0.800** ± 0.054	**0.802** ± 0.046	**0.802** ± 0.055
Bandwidth	SVM-Kernal (Gaussian)					
0.1	0.561 ± 0.139	0.517 ± 0.159	0.530 ± 0.246	0.452 ± 0.038	0.545 ± 0.150	0.506 ± 0.196
1.0	0.610 ± 0.071	0.551 ± 0.182	0.615 ± 0.140	0.604 ± 0.148	0.620 ± 0.120	0.622 ± 0.161
# of Classifiers	AdaBoost					
100.0	0.554 ± 0.061	0.393 ± 0.304	0.554 ± 0.044	0.433 ± 0.264	0.553 ± 0.059	0.344 ± 0.175
200.0	0.565 ± 0.050	0.449 ± 0.257	0.568 ± 0.047	0.485 ± 0.229	0.563 ± 0.050	0.385 ± 0.126

TABLE IV: Performance of the Averaged LDA Classifier

Dataset	**Accuracy**	**F1-Score**
Adult-1	0.761±0.000	0.747±0.000
Adult-2	0.758±0.000	0.746±0.000
Adult-3	0.752±0.000	0.743±0.000
Web-1	0.654±0.000	0.575±0.000
Web-2	0.682±0.000	0.619±0.000
Web-3	0.687±0.000	0.635±0.000

for many applications (with no real-time requirements). For example, $\mathcal{AWDA}$ consumes only 60 milliseconds to classify a 300-dimension Web-1 data instance — $\mathcal{AWDA}$ can scan 60,000 web pages in one hour using a single PC. Moreover, the Monte-Carlo sampling method for integral computation can be easily parallelized with cloud computing systems, and the time consumption can be further reduced for training and testing.

In summary, the experimental results on time consumption verify our algorithm designs. The experiments were all carried out using an iMac desktop with 3.1 GHz Intel Core i5 CPU, 16G memory and macOS v10.12.

B. Experiment II: Early Detection of Diseases using Electronic Health Records (EHR) Data

While the first experiment validates the effectiveness of $\mathcal{AWDA}$ on common benchmark datasets with fixed training dataset size, in the second experiment, we evaluate the algorithm using a real-world application with varying training dataset sizes to demonstrate the potential of the proposed algorithm for handling the real-world problems.

1) Experiment Setups: We use the de-identified EHR data from the College Health Surveillance Network (CHSN), which contains over 1 million patients and 6 million visits from 31 student health centers across the United States [33]. Among all diseases recorded in CHSN, we choose mental health disorders, including *(a) anxiety disorders, (b) mood disorders, (c) depression disorders, and (d) other related mental health disorders*, as the targeted disease for early detection. We represent each patient using his/her diagnosis-frequency vector [34] based on the clustered code set (the number of dimensions $p = 295$), where four selected mental health disorder codes are considered to represent the diagnoses of

TABLE V: Time Consumption (milliseconds) of Baseline Algorithms on Adult-1 and Web-1 Datasets

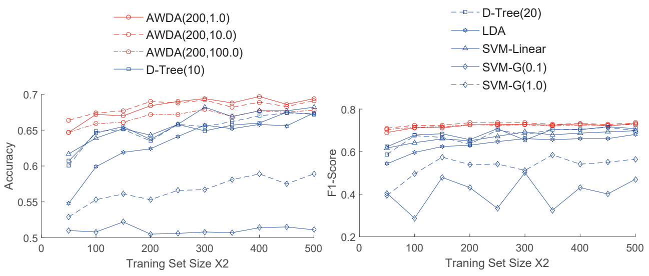
	Benchmark Datasets			
	Adult-1		Web-1	
Parameter	*Training*	*Testing*	*Training*	*Testing*
Depth	Decision Tree			
10.0	4.100	0.005	2.750	0.004
20.0	5.250	0.007	4.800	0.004
	LDA			
	23.300	0.040	274.000	0.166
	Two-Stage LDA (with dimension reduced [31, 32])			
	21.150	0.022	142.600	0.005
	SVM-Linear			
	6.950	0.005	3.350	0.005
Bandwidth	SVM-Kernal (Gaussian)			
0.1	3.250	0.031	2.400	0.046
1.0	1.700	0.021	1.800	0.036
Instance Number	AdaBoost			
100.0	316.850	0.119	104.050	0.194
200.0	608.150	0.230	209.200	0.432

mental health disorders and we do not predict these four types of mental disorders separately. Specifically, if a patient has any of these four codes in his/her EHR, we say that he/she has been diagnosed with mental health disorders as ground truth.

In order to test the early detection of diseases, for each patient with mental health disorders, we use his/her historical EHR data that was generated 90 days before he/she received the first mental health disorders diagnosis. Further, patients with less than two visits were excluded from the analysis. To demonstrate the effectiveness of our method, we continue using the *Accuracy* and *F1-Score* for evaluating the performance early detection, and compare our method with the same set of baseline algorithms. Specifically, the Accuracy metric characterizes the proportion of patients who are accurately classified in the early detection of mental disorders. The F1-Score measures both correctness and completeness of the early detection.

We perform experiments with the following settings: to build the *training sets*, we randomly select 50 to 500 patients with mental health disorders as the positive training samples, and randomly select the same number of patients not been diagnosed with any mental health disorders as negative training samples. Thus the training set for the two classes is balanced (i.e., the number of dimensions $p = 295$ and training set size is $50 \sim 500 \times 2$). To build the testing sets, we randomly select 200 patients (not included in the training set) from both positive/negative groups. Also the testing set is balanced. For each setting, we execute the seven algorithms and repeat 30 times.

2) Overall Comparison : Figure. 2 presents the performance of our approach, along with classical LDA, linear SVM, Kernel SVM and Decision Trees on 200 testing samples and varying training sample sizes. $\mathcal{AWDA}$(200, 1.0) refers to the



(a) Accuracy (b) F1-Score

Fig. 2: Overall Performance Comparison with Downstream Classifiers on EHR Data

$\mathcal{AWDA}$ classifier based on 200 sampled inverse covariance matrices using De-Sparsified Graphical Lasso with $\lambda = 1.0$ for Wishart scale matrix estimation. As can be seen from the results, $\mathcal{AWDA}$ clearly outperforms the baseline algorithms in terms of the overall accuracy, and F1-score. Due to the space limit, we don't present the comparison results based on Two-stage LDA in Figure. 2. $\mathcal{AWDA}$(200, 1.0) outperforms Two-stage LDA by achieving on average 4.3% higher accuracy and 3.5% higher F1-score. It is clear that $\mathcal{AWDA}$ outperforms all these algorithms significantly.

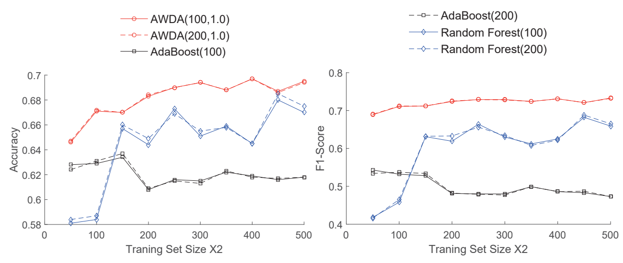
3) Comparison to Ensemble learners: As $\mathcal{AWDA}$ ensembles the classification results from multiple classifiers, we also compared $\mathcal{AWDA}$ to the existing ensemble learning algorithms, such as Random Forest and AdaBoost. To compare with ensemble learners with 100 and 200 basis classifiers, we use $\mathcal{AWDA}$ with 100 and 200 sampled inverse covariance matrices (i.e., ensemble with 100 and 200 LDA classifiers), with $\lambda = 1.0$ for Wishart mean matrix regularization.

The performance comparison is illustrated in Figure. 3. It is obvious that $\mathcal{AWDA}$ outperforms these two algorithms in both 100-instance and 200-instance settings, while the performance of Random Forest is not quite stable. Moreover, Figure. 3 also shows the performance of $\mathcal{AWDA}$ classifiers with 100 and 200 sampled inverse covariance matrices are very similar. Indeed, we tested $\mathcal{AWDA}$ with 50 to 2000 inverse covariance matrices, the prediction accuracy or F1-scores of $\mathcal{AWDA}$ is almost consistent on the varying number of sampled matrices. This indicates that $\mathcal{AWDA}$ can provide robust prediction performance, even when only a small number of inverse covariance matrices are sampled.

4) Comparison on Discretization and Regularization: $\mathcal{AWDA}$ leverages importance sampling-alike method to improve the performance of approximation to the real integral in a discretization manner. We compare $\mathcal{AWDA}$ to a classifier namely "Discrete-$\mathcal{AWDA}$" based on the simple discretization strategy:

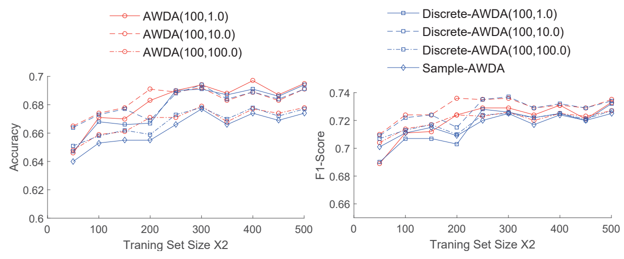
$$sign\left(\sum_{1\leq i\leq m} f(\mathbf{x}, \bar{x}, \bar{x}_{+1}, \bar{x}_{-1}, \Theta_i) P(x|\Theta_i) P_w(\Theta_i|\widehat{T}, v)\right).$$

Both two algorithms leverage a De-sparsified Graphical Lasso with $\lambda = 1.0, 10.0, 100.0$ for the Wishart mean matrix



(a) Accuracy (b) F1-Score

Fig. 3: Performance Comparison with Ensemble Learning Classifiers on EHR Data



(a) Accuracy (100) (b) F1-Score (100)

Fig. 4: Performance Comparison with Different Discretization and Regularization

estimation. Figure. 4 presents the performance comparison. It shows $\mathcal{AWDA}$ outperforms the simple discretization strategy using the same λ significantly.

In Figure 4, we also demonstrate the performance improvement contributed by regularization (De-sparsified graphical lasso) for Wishart mean matrix estimation. The lines entitled "Sample-$\mathcal{AWDA}$" refer to a derived method using the sample inverse covariance matrix (pseudo-inverse when the covariance matrix is singular) as the Wishart mean matrix. It shows that $\mathcal{AWDA}$ can outperform Sample-$\mathcal{AWDA}$ significantly.

VI. Related Work & Discussion

In this section, we first summarize the most relevant work of this paper. Then we compare our work to these related researches. Further, we discuss several open issues of our study.

We introduce several LDA extensions under High Dimension Low Sample Size (HDLSS) settings. As discussed above, when LDA works in HDLSS, there exist two major technical flaws: 1) the covariance matrix estimated using the sample estimator is usually singular under HDLSS settings, while LDA requires inverting covariance matrices for classification, and 2) under HDLSS, the sample covariance matrix estimator used in Fisher's LDA is usually with large variance, which might lead to inaccurate estimation of inverse covariance matrix [3]. To handle the singularity issue, Ye et al. [30] first extended the Pseudo-inverse and proposed a new optimization criteria to handle the singularity issues, then the same group of researchers proposed Two-staged LDA [31, 32] to improve the performance of LDA with reduced dimensionality. Direct LDA [35, 36] was proposed to use the *simultaneous diagonalization* of covariance matrices, which are non-singular, to replace the original covariance matrices. On the other hand, several researchers [9, 10, 37, 38] have proposed regularization algorithm that lowers the variance of linear coefficients for classification using ℓ_1/ℓ_2-norm penalties.

Compared to the above work, $\mathcal{AWDA}$ is distinct and novel in two ways. *First*, compared to other extensions of LDA, which focus on selecting an optimal parameter/model for binary classification, $\mathcal{AWDA}$ improves the performance of LDA model using a model averaging scheme [17]. Furthermore, compared to some statistical approaches [39] that rely on an extremely large dataset and sample (inverse) covariance matrices by re-grouping the dataset, our algorithm approximates the distribution of inverse covariance matrices using a Wishart distribution, and leverages Monte-Carlo sampling to approximate the results of Bayesian inference.

Second, our algorithm provides "local adaption" to the classical Bayesian inference for LDA classification. Per each input data for classification, $\mathcal{AWDA}$ updates the weight of each classifier in an optimal manner via a Gaussian prior. The significant performance improvement from the classical LDA is partially due to the local adaption. Note that the existing adaptive Bayesian inference [40] is mainly based on MCMC sampling and "proposal and rejection" mechanism. These methods need to sample a new group of parameters per each new input. However, the time complexity of the sampling process, which depends on how many sampled parameters are rejected by new input data, is usually not controllable. In the worst case, the time consumption of the sampling process might grow exponentially with the sampling complexity (e.g., m in our research). Our method decomposes the training and testing into phases: $\mathcal{AWDA}$ only needs to sample a group of inverse covariance matrices once for building multiple classifiers, then classify each new input data adaptively by using updated weights for multi-classifier ensemble. Through using a weight-updating mechanism, $\mathcal{AWDA}$ can work on a fixed number of sampled inverse covariance matrices, and thus significantly reduce the time complexity of sampling procedure (as was verified in our experiments, the training/testing, i.e., sampling/inference, time complexity growth linearly with the increasing m). Furthermore, compared to [40, 41], $\mathcal{AWDA}$ is the first attempt of adaptive Bayesian inference to Fisher's LDA, by addressing the uncertainty of inverse covariance matrix estimation and leveraging Wishart distribution as prior.

VII. Conclusion

In this paper, we proposed $\mathcal{AWDA}$ – a novel Fisher's LDA extension for binary classification under HDLSS. $\mathcal{AWDA}$ lowers the uncertainty of LDA parameter estimation, in a model averaging fashion, and further enables the nonlinear classification, through the Input-Adaptive Bayesian voting scheme. Theoretical analysis shows that $\mathcal{AWDA}$ guarantees a close approximation to the optimal Bayesian inference. The experimental results on common binary classification benchmarks and real-world EHR datasets show that $\mathcal{AWDA}$ outperforms baseline algorithms significantly.

References

[1] Onur C Hamsici and Aleix M Martinez. Bayes optimality in linear discriminant analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4):647–657, 2008.

[2] Amin Zollanvari and Edward R Dougherty. Random matrix theory in pattern classification: An application to error estimation. In *2013 Asilomar Conference on Signals, Systems and Computers*, 2013.

[3] T Tony Cai, Zhao Ren, Harrison H Zhou, et al. Estimating structured high-dimensional covariance and precision matrices: Optimal rates and adaptive estimation. *Electronic Journal of Statistics*, 10(1):1–59, 2016.

[4] Olvi L Mangasarian. Linear and nonlinear separation of patterns by linear programming. *Operations research*, 13(3):444–452, 1965.

[5] Kristin P Bennett and Olvi L Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization methods and software*, 1(1):23–34, 1992.

[6] Roger Peck and John Van Ness. The use of shrinkage estimators in linear discriminant analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (5):530–537, 1982.

[7] Juwei Lu, Konstantinos N Plataniotis, and Anastasios N Venetsanopoulos. Regularization studies of linear discriminant analysis in small sample size scenarios with application to face recognition. *Pattern Recognition Letters*, 26(2):181–191, 2005.

[8] Daniela M Witten and Robert Tibshirani. Covariance-regularized regression and classification for high dimensional problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(3):615–636, 2009.

[9] Line Clemmensen, Trevor Hastie, Daniela Witten, and Bjarne Ersbøll. Sparse discriminant analysis. *Technometrics*, 53(4), 2011.

[10] Jun Shao, Yazhen Wang, Xinwei Deng, Sijian Wang, et al. Sparse linear discriminant analysis by thresholding for high dimensional data. *The Annals of statistics*, 39(2):1241–1265, 2011.

[11] Peter Buhlmann and Sara Van De Geer. *Statistics for high-dimensional data: methods, theory and applications*. Springer, 2011.

[12] Neil D Lawrence and Bernhard Schölkopf. Estimating a kernel fisher discriminant in the presence of label noise. In *ICML*, volume 1, pages 306–313. Citeseer, 2001.

[13] Zhihua Zhang et al. Learning metrics via discriminant kernels and multidimensional scaling: Toward expected euclidean representation. In *ICML*, volume 2, pages 872–879, 2003.

[14] Seung-Jean Kim, Alessandro Magnani, and Stephen Boyd. Optimal kernel selection in kernel fisher discriminant analysis. In *ICML*, pages 465–472. ACM, 2006.

[15] Zhihua Zhang, Guang Dai, Congfu Xu, and Michael I Jordan. Regularized discriminant analysis, ridge regression and beyond. *JMLR*, 11(Aug):2199–2228, 2010.

[16] Kenneth P Burnham and David R Anderson. *Model selection and multimodel inference: a practical information-theoretic approach*. Springer Science & Business Media, 2003.

[17] Jennifer A Hoeting, David Madigan, Adrian E Raftery, and Chris T Volinsky. Bayesian model averaging: a tutorial. *Statistical science*, pages 382–401, 1999.

[18] Nello Cristianini and John Shawe-Taylor. Bayesian voting schemes and large margin classifiers. *Advances in Kernel MethodsSupport Vector Learning*, pages 55–68, 1999.

[19] Pascal Germain, Alexandre Lacasse, Francois Laviolette, Mario Marchand, and Jean-Francis Roy. Risk bounds for the majority vote: From a pac-bayesian analysis to a learning algorithm. *The Journal of Machine Learning Research*, 16(1):787–860, 2015.

[20] David GT Denison. *Bayesian methods for nonlinear classification and regression*, volume 386. John Wiley & Sons, 2002.

[21] John Hammersley. *Monte carlo methods*. Springer Science & Business Media, 2013.

[22] Philip J Davis and Philip Rabinowitz. *Methods of numerical integration*. Courier Corporation, 2007.

[23] Jana Jankova, Sara van de Geer, et al. Confidence intervals for high-dimensional inverse covariance estimation. *Electronic Journal of Statistics*, 9(1):1205–1229, 2015.

[24] LR Haff. Estimation of the inverse covariance matrix: Random mixtures of the inverse wishart matrix and the identity. *The Annals of Statistics*, pages 1264–1276, 1979.

[25] S Sawyer. Wishart distributions and inverse-wishart sampling. *URL: www. math. wustl. edu/sawyer/hmhandouts/Whishart. pdf*, 2007.

[26] Tom Leonard and John SJ Hsu. Bayesian inference for a covariance matrix. *The Annals of Statistics*, pages 1669–1696, 1992.

[27] Santosh Srivastava, Maya R Gupta, and Béla A Frigyik. Bayesian quadratic discriminant analysis. *Journal of Machine Learning Research*, 8(Jun):1277–1305, 2007.

[28] Ignacio Alvarez, Jarad Niemi, and Matt Simpson. Bayesian inference for a covariance matrix. *arXiv preprint arXiv:1408.4050*, 2014.

[29] John C Platt. 12 fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods*, pages 185–208, 1999.

[30] Jieping Ye, Ravi Janardan, Cheong Hee Park, and Haesun Park. An optimization criterion for generalized discriminant analysis on undersampled problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):982–994, 2004.

[31] Jieping Ye, Ravi Janardan, Qi Li, et al. Two-dimensional linear discriminant analysis. In *NIPS*, volume 4, page 4, 2004.

[32] Jieping Ye and Qi Li. A two-stage linear discriminant analysis via qr-decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):929–941, 2005.

[33] James C. Turner and Adrienne Keller. College Health Surveillance Network: Epidemiology and Health Care Utilization of College Students at U.S. 4-Year Universities. *Journal of American college health: J of ACH*, page 0, June 2015.

[34] Fei Wang and Jimeng Sun. Psf: a unified patient similarity evaluation framework through metric learning with weak supervision. *IEEE journal of biomedical and health informatics*, 19(3):1053–1060, 2015.

[35] Juwei Lu, Kostantinos N Plataniotis, and Anastasios N Venetsanopoulos. Face recognition using lda-based algorithms. *Neural Networks, IEEE Transactions on*, 14(1):195–200, 2003.

[36] Hui Gao and James W Davis. Why direct lda is not equivalent to lda. *Pattern Recognition*, 39(5):1002–1006, 2006.

[37] Zhihua Qiao, Lan Zhou, and Jianhua Z Huang. Effective linear discriminant analysis for high dimensional, low sample size data. In *Proceeding of the World Congress on Engineering*, volume 2, pages 2–4. Citeseer, 2008.

[38] Tony Cai and Weidong Liu. A direct estimation approach to sparse linear discriminant analysis. *Journal of the American Statistical Association*, 106(496):1566–1577, 2011.

[39] Christophe Croux, Peter Filzmoser, and Kristel Joossens. Classification efficiencies for robust linear discriminant analysis. *Statistica Sinica*, pages 581–599, 2008.

[40] Luke Tierney and Antonietta Mira. Some adaptive monte carlo methods for bayesian inference. *Statistics in medicine*, 18(1718):2507–2515, 1999.

[41] Umut A Acar, Alexander Ihler, Ramgopal Mettu, and Ozgür Sümer. Adaptive bayesian inference. In *Neural Information Processing Systems (NIPS)*, 2007.