applied sciences

MDPI

Article

Regularized Within-Class Precision Matrix Based PLDA in Text-Dependent Speaker Verification

Sung-Hyun Yoon [1], Jong-June Jeon [2] and Ha-Jin Yu [1,*]

[1] School of Computer Science, University of Seoul, Seoul 02054, Korea; ysh901108@naver.com
[2] Department of Statistics, University of Seoul, Seoul 02504, Korea; jj.jeon@gmail.com
* Correspondence: hjyu@uos.ac.kr; Tel.: +82-2-6490-2448

Received: 10 August 2020; Accepted: 17 September 2020; Published: 20 September 2020

Abstract: In the field of speaker verification, probabilistic linear discriminant analysis (PLDA) is the dominant method for back-end scoring. To estimate the PLDA model, the between-class covariance and within-class precision matrices must be estimated from samples. However, the empirical covariance/precision estimated from samples has estimation errors due to the limited number of samples available. In this paper, we propose a method to improve the conventional PLDA by estimating the PLDA model using the regularized within-class precision matrix. We use graphical least absolute shrinking and selection operator (GLASSO) for the regularization. The GLASSO regularization decreases the estimation errors in the empirical precision matrix by making the precision matrix sparse, which corresponds to the reflection of the conditional independence structure. The experimental results on text-dependent speaker verification reveal that the proposed method reduce the relative equal error rate by up to 23% compared with the conventional PLDA.

Keywords: graphical least absolute shrinking and selection operator (GLASSO); precision matrix; probabilistic linear discriminant analysis (PLDA); regularization; text-dependent speaker verification

1. Introduction

Automatic speaker verification (ASV) is a technique to verify a user's identity by comparing an utterance of a user (test utterance) with the reference utterance of a known target speaker (enrollment utterance). The procedure for ASV can be divided into two steps: front-end feature extraction and back-end scoring. In the front-end step, a fixed-size feature vector is extracted from a variable-length utterance, for both the enrollment and test utterance. The feature vector (called speaker embedding) should be extracted to represent the speaker information well. In the back-end step, a similarity score between the two speaker embeddings, one for the enrollment utterance and the other for the test utterance, is computed to accept or reject the identity claim [1].

Speaker verification can be divided into two categories: text-independent speaker verification (TI-SV) [2] and text-dependent speaker verification (TD-SV). In this paper, we focus on TD-SV only and explain based on TD-SV. The main difference between TI-SV and TD-SV is that whether the phrase of utterance is limited. In TI-SV, no limitation exists for the phrase, which enables users speak any types of phrases. TI-SV systems must compensate for phrase variability to improve the verification performance. In this case, sufficiently long utterances, for example longer than about 10 s, are required to effectively compensate for the phrase variability, or the performance would be significantly degraded. It means that users have to speak long enough, which makes the use of TI-SV systems inconvenient. Moreover, the longer utterance, the larger computational cost. These shortcomings can be solved by TD-SV. In TD-SV, the available lexicon is limited for a few kinds of phrases, and the phrases of both the enrollment and test utterances should be the same. Even though the limitation for the phrase makes TD-SV be less flexible than TI-SV, it enables TD-SV to show both the higher performance and lower

computational cost even with short utterances, for example, shorter than about 3 s. Because TD-SV has no need to compensate for the phrase variability, rather it should distinguish between not only speakers but also phrases. In addition, it is relatively easy to control the phrase variability in TD-SV because of the limitation for the phrase. Due to these advantages, TD-SV has been widely used in many real applications that require both the higher performance and short utterance, such as voice assistant [3,4]. One drawback of TD-SV is that it can be vulnerable to replay attacks which use recorded voices of the target user. However, currently there have been many active studies to prepare for the attacks and they have achieved high performance [5–12].

As mentioned above, speaker embedding is the feature vector on fixed-dimension subspace that represents speaker information contained in the utterance. Typically, the speaker embedding not only has class information (corresponding to speaker-and-phrase information in TD-SV) but also undesired information, such as session information [13]. Note that the speaker embedding in TD-SV actually contains not only speaker information and but also phrase information. Throughout this paper, nevertheless, we call it speaker embedding for convenience, rather than speaker-and-phrase embedding. Too much undesired information raises the within-class variability, which can degrade the speaker verification performance. However, it is challenging to completely remove only unwanted information in the embedding. In many cases, the score is computed in a more discriminative subspace to compensate for the within-class variability of the embedding [14].

In the field of ASV, probabilistic linear discriminant analysis (PLDA) [15] has become the dominant method for back-end scoring. It probabilistically models a discriminative subspace that compensates for the within-class variability. To estimate a PLDA model, the between-class covariance matrix and within-class precision matrix (the inverse of the within-class covariance matrix) must be estimated first. In practice, the empirical covariance/precision matrix is used instead of the true covariance/precision matrix because it is unknown. However, the empirical covariance/precision matrix has estimation errors because of the limited number of available samples (corresponding to the embeddings in our case). The error is increased if the number of samples is insufficient compared to the number of parameters, which can degrade performance.

In this paper, we propose a method to improve the performance of conventional PLDA by regularizing the within-class precision matrix used to estimate the PLDA model. The regularization of the within-class precision matrix is motivated by the need for the reduction of estimation errors contained in the empirical within-class precision matrix. We use graphical least absolute shrinking and selection operator (GLASSO) [16–18] to regularize the within-class precision matrix, which makes the precision matrix sparse. The reason for using the GLASSO for regularization, among many kinds of regularization methods, is that the GLASSO makes precision matrix sparse, which is based on our assumption that the true precision matrix has the conditional independence structure. Because the sparsity in the precision matrix implies the conditional independence of feature variables, the GLASSO can be understood as a de-noising operator to reflect the conditional independence structure in the underlying model. We focus on only the within-class covariance/precision, which is based on our assumption that empirical within-class covariance/precision has larger errors than empirical between-class covariance, because collecting sufficient utterances from the same class is relatively more difficult than collecting similar numbers of utterances from various classes. Therefore, it is expected that regularizing the within-class covariance/precision matrix may be more effective in terms of performance.

The remainder of this paper is organized as follows. Section 2 outlines the preliminaries related to our research. Section 3 describes the GLASSO and related theory. Section 4 introduces the proposed method. Section 5 presents the experiments and their results. Section 6 discusses some issues about text-independent speaker verification and one more option for our solution. Finally, Section 7 concludes the paper.

2. Preliminaries

2.1. I-Vector

Traditional speaker embeddings were mostly based on linear generative models in statistics [2,14]. The i-vector [14] is one of the speaker embeddings based on factor analysis and has been the state-of-the-art speaker embedding in ASV until the development of deep learning. It is a set of factors that explain the total variability of the Gaussian mixture model (GMM) supervector [19]. In the i-vector framework, an utterance-dependent supervector **M** is represented by the following:

$$\mathbf{M} = \mathbf{m} + \mathbf{Tx} \tag{1}$$

where **m** is an utterance-independent supervector (generally, a universal background model [20] supervector), **T** is a low-rank rectangular matrix that defines a total variability subspace, and **x** is a set of latent variables that follows a standard multivariate normal distribution $N(0, \mathbf{I})$. In practice, however, i-vectors exhibit non-Gaussian behavior [21,22]. In most cases, a simple length normalization [23] is applied to i-vector to reduce the non-Gaussian behavior.

The i-vector extractor (corresponding to the total variability matrix **T**) cannot be directly estimated to discriminate between classes because it is trained in an unsupervised manner. It means that the i-vector may have unnecessarily large within-class variability. Therefore, it is necessary to compensate for the within-class variability of the i-vector for robust performance [24]. The PLDA has been dominantly used to compensate for the within-class variability, and the i-vector/PLDA had been the state-of-the-art option in the field of ASV.

2.2. Deep Speaker Embeddings

Owing to remarkable development in deep learning, many studies have investigated the use of deep neural networks (DNNs) to extract more discriminative speaker embeddings. The DNNs have the advantages that they can represent complex nonlinear models and be directly optimized to discriminate between classes. Deep speaker embedding is extracted from a hidden layer of a speaker-and-phrase discriminate DNN and is expected to have more discriminative power. The deep speaker embeddings have become the state-of-the-art in the field of ASV [25].

Early deep speaker embeddings (called d-vectors) were based on a fully connected neural network (FCNN) [26,27]. The FCNNs were trained to classify frame-level acoustic features in a temporal context to the corresponding speaker and phrase. However, the FCNN cannot properly model the time-dependency of the frame-level features in a context. To overcome this drawback, the methods of using other kinds of DNNs, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), were proposed [28–30]. RNN is a neural network designed to model time-dependency of time-series data using additional recurrent connections. In this paper, we use the d-vector based on long short-term memory (LSTM) [31], which is an extension of RNN for modeling long term time dependencies efficiently by using gates mechanisms [32].

A residual network (ResNet) [33] is the network that has constant bypass weight connections between layers to optimize very deep networks efficiently. Motivated by the success of ResNet in the field of image recognition, many studies have employed ResNet to model deep speaker embedding (called the r-vector to distinguish from the original d-vector) and have shown remarkable performance [34–37]. In [38], the squeeze-and-excitation (SE) block was proposed, which is the building block of CNNs to recalibrate channel-wise responses adaptively by explicitly modeling the relationship between channels. The SE block has been successfully adopted in ResNet, which is called a squeeze-and-excitation residual network (SE-ResNet). In this study, we extract the r-vectors from SE-ResNet34 [6,11].

The x-vector [39,40] is the deep speaker embedding extracted from a model based on a time-delay neural network (TDNN) [41]. The TDNN is the network that computes an output at each time step

using the inputs from a small temporal context window similarly to CNNs. We can model long-term context information of input by stacking multiple TDNN layers, even if each TDNN layer has a small temporal context. Therefore, the TDNN-based model can efficiently capture long-term speaker information. Each layer of the TDNN models a small temporal context of the output from the previous layer. The output frames of the TDNN are aggregated over temporal pooling to capture long-term information. The x-vector/PLDA has shown state-of-the-art performance in TI-SV [25].

Like the i-vector, deep speaker embeddings have also been used with the PLDA and exhibit performance improvements. One of the main differences between the i-vector and deep speaker embeddings is that no assumption exists regarding the shape, such as the probability distribution and covariance structure, of deep speaker embeddings. This difference also affects our proposed method, which is described later.

2.3. Probabilistic Linear Discriminant Analysis (PLDA)

The PLDA is a generative probabilistic method that models between-class and within-class variabilities using latent variables. The goal of PLDA is to determine a more discriminative subspace that maximizes between-class variability and minimizes within-class variability. Some variants of PLDA exist [15,21,23,42,43]. In this paper, we use the PLDA implemented in the Kaldi toolkit [44], which is the two-covariance PLDA [43] based on [15]. In Kaldi's PLDA, a speaker embedding $\mathbf{x}$ is modeled as follows:

$$\mathbf{x} = \boldsymbol{\mu} + \mathbf{A}\mathbf{u} \tag{2}$$

$$\mathbf{u}|\mathbf{v} \sim N(\mathbf{v}, \mathbf{I}) \tag{3}$$

$$\mathbf{v} \sim N(0, \boldsymbol{\Psi}) \tag{4}$$

where $\boldsymbol{\mu}$ is the global mean of the embeddings in the original space, $\mathbf{A}$ is the PLDA projection matrix, $\mathbf{v}$ represents the class in the projected space, $\mathbf{u}$ represents an example of that class in the projected space, and $\boldsymbol{\Psi}$ is the between-class diagonal covariance in the projected space. The PLDA model parameters $\{\boldsymbol{\mu},\ \mathbf{A},\ \boldsymbol{\Psi}\}$ is estimated by the eigenvalue equation $\boldsymbol{\Phi}_w^{-1}\boldsymbol{\Phi}_b\mathbf{A} = \mathbf{A}\boldsymbol{\Psi}$, where $\boldsymbol{\Phi}_b$ and $\boldsymbol{\Phi}_w$ are the between-class covariance matrix and within-class covariance matrix, respectively.

The expectation-maximization (EM) [45] algorithm is used to estimate $\boldsymbol{\Phi}_b$ and $\boldsymbol{\Phi}_w$. Note that the EM algorithm is a greedy algorithm, which guarantees the convergence by updating the parameters iteratively. However, there is no guarantee to converge toward the global optimum. Therefore, the estimated PLDA model cannot guarantee the global optimum. It starts with the initial values of the between-class covariance $\boldsymbol{\Phi}_b^{(0)}$ and within-class covariance $\boldsymbol{\Phi}_w^{(0)}$ matrices, which can be directly computed from the training dataset, as follows:

$$\boldsymbol{\Phi}_b^{(0)} = \sum_{c=1}^{C} (\boldsymbol{\mu}^c - \boldsymbol{\mu})(\boldsymbol{\mu}^c - \boldsymbol{\mu})^T \tag{5}$$

$$\boldsymbol{\Phi}_w^{(0)} = \sum_{c=1}^{C} \frac{1}{N_c} \sum_{n=1}^{N_c} (\mathbf{x}_n^c - \boldsymbol{\mu}^c)(\mathbf{x}_n^c - \boldsymbol{\mu}^c)^T \tag{6}$$

where C is the number of speakers in the training dataset, $\boldsymbol{\mu}^c$ is the mean of the embeddings for the c-th speaker, N_c is the number of utterances for the c-th speaker, and $\mathbf{x}_n^c$ is the n-th embedding for the c-th speaker. A detailed explanation can be found in [15,46].

The log-likelihood ratio, which literally means the log ratio between two likelihoods and is used as the similarity score in our experiments, between two embeddings, $\mathbf{x}_1$ and $\mathbf{x}_2$, is computed by the following:

$$\log N\left(\bar{\mathbf{u}}_1 \middle| \frac{\boldsymbol{\Psi}}{\boldsymbol{\Psi}+\mathbf{I}}\bar{\mathbf{u}}_2,\ \mathbf{I} + \frac{\boldsymbol{\Psi}}{\boldsymbol{\Psi}+\mathbf{I}}\right) - \log N\left(\bar{\mathbf{u}}_1|0,\ \mathbf{I}+\boldsymbol{\Psi}\right) \tag{7}$$

where $\bar{\mathbf{u}}_i = \mathbf{u}_i / \sqrt{\frac{\mathbf{I}}{\Psi+\mathbf{I}}\mathbf{u}_i^2}$ is the length normalization of $\mathbf{u}_i$, $\mathbf{u}_i = \mathbf{A}^T(\mathbf{x}_i - \boldsymbol{\mu})$ is the projected embedding, the subscript $\cdot_i$ is an arbitrary index of embedding for distinguishing each other, and $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the Gaussian probability density function with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. An arbitrary embedding in the original space $\mathbf{x}_i$ is projected into the projected space through the transformation $\mathbf{u}_i = \mathbf{A}^T(\mathbf{x}_i - \boldsymbol{\mu})$. Note that $\mathbf{x}_i$ can be both $\mathbf{x}_1$ and $\mathbf{x}_2$. In other words, the same transformation $\mathbf{u}_i = \mathbf{A}^T(\mathbf{x}_i - \boldsymbol{\mu})$ is applied to $\mathbf{x}_i$, regardless of the kind of subscript $\cdot_i$. Therefore, $\mathbf{x}_i$ can correspond to both $\mathbf{x}_1$ and $\mathbf{x}_2$.

3. Graphical Least Absolute Shrinking and Selection Operator (GLASSO)

3.1. Gaussian Markov Random Field

Consider a D-dimensional random vector $\mathbf{x} = [x_1, \ldots, x_D]^T$ (corresponding to the embedding in our case), which is a set of D random variables x_i. The random vector $\mathbf{x}$ is a Markov random field (MRF; an undirected graphical model) if it satisfies Markov properties [47]. An undirected graph $G = (V, E)$, where V is a set of vertices and E is a set of edges, can describe the random vector. Each vertex in V corresponds to one of the random variables in $\mathbf{x}$, that is, $V = \{x_1, \ldots, x_D\}$. Each edge e_{ij} in E represents the dependency between x_i and x_j such that $i \neq j$. In undirected graphical models, all edges $e_{ij} \in E$ are unordered pairs, that is, $e_{ij} = e_{ji}$. The Markov property relates to conditional independence, and three kinds of Markov properties exist: the pairwise, local, and global Markov properties. In this study, we focus on the pairwise Markov property. The pairwise Markov property is that variables x_i and x_j are conditionally independent given all the other variables $\mathbf{x}_{-ij}$, $x_i \perp x_j | \mathbf{x}_{-ij}$, which is equivalent to stating that no edge e_{ij} exists in E [48].

The random vector $\mathbf{x}$ is a Gaussian Markov random field (GMRF; a Gaussian undirected graphical model) if it satisfies the Markov property and follows a multivariate normal distribution $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with mean $\boldsymbol{\mu} \in \mathbb{R}^D$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{D \times D}$. In other words, the GMRF is the MRF following a multivariate normal distribution. In the GMRF, the set of edges E is represented by the precision matrix $\boldsymbol{\Sigma}^{-1}$. The edge e_{ij} corresponds to the element in the i-th row and the j-th column of the precision matrix $\boldsymbol{\Sigma}_{ij}^{-1}$, and no edge e_{ij} exists if and only if $\boldsymbol{\Sigma}_{ij}^{-1} = 0$. Therefore, the following three statements are equivalent to each other: (i) there is no edge e_{ij}, (ii) $\boldsymbol{\Sigma}_{ij}^{-1} = 0$, and (iii) $x_i \perp x_j | \mathbf{x}_{-ij}$. To summarize, we can reflect the conditional independence structure to the variables by making $\boldsymbol{\Sigma}^{-1}$ sparse [47].

3.2. GLASSO

The GLASSO is a variable selection method to estimate a sparse precision matrix using the L_1 (lasso) penalty. In other words, it estimates a sparse undirected graphical model. Consider that we have samples (corresponding to speaker embeddings in our case) that follow a multivariate normal distribution $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Let $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$ be the true precision matrix and $\mathbf{S}$ be the empirical covariance matrix. The GLASSO-regularized precision matrix $\hat{\boldsymbol{\Theta}}$ is defined by maximizer of the L_1-penalized Gaussian log-likelihood:

$$\hat{\boldsymbol{\Theta}} = \underset{\boldsymbol{\Theta}}{\operatorname{argmax}} \{\log\det(\boldsymbol{\Theta}) - \operatorname{tr}(\mathbf{S}\boldsymbol{\Theta}) - \rho\|\boldsymbol{\Theta}\|_1\} \tag{8}$$

where $\det(\cdot)$ and $\operatorname{tr}(\cdot)$ are the determinant and trace of a matrix, respectively, ρ (> 0) is a regularization parameter, and $\|\cdot\|_1$ is the L_1 norm operator (sum of the absolute values of the elements). We omit diagonal elements from the penalty. Therefore, only off-diagonal elements are penalized.

The GLASSO is a biased estimator that shrinks all non-zero elements in the estimated precision matrix toward zero. A higher ρ results in (i) more regularization, (ii) lower estimation error with an accompanying higher bias in the estimated precision matrix, and (iii) a sparser estimated precision, and vice versa. Thus, the GLASSO requires the sparse assumption of the true precision matrix for the desired good asymptotic properties, such as selection consistency [49,50]. In addition, the selection

consistency requires a strict condition on the covariance matrix $\mathbf{S}$, the irrepresentable condition [49,50]. Typically, the GLASSO can detect the conditional independence structure on the considered covariates only when the covariates are not severely dependent. Therefore, a proper value of ρ must be selected for the performance, and simultaneously, the underlying covariance structure of the considered model should be considered. To summarize, the GLASSO regularization can reduce the estimation error in the precision matrix by pursuing a sparse structure, but the associated improvement depends on the underlying model.

In [51], solve (8) using convex duality, taking advantage of the fact that the optimization of (8) is a kind of convex optimization. In other words, [51] solve (8) by estimating $\mathbf{\Sigma}$, rather than $\mathbf{\Sigma}^{-1}$, using block coordinate descent algorithm, as follows. Let $\mathbf{W}$ be the regularization of $\mathbf{S}$. In general, $\mathbf{W}$ is initialized to $\mathbf{S} + \rho\mathbf{I}$. This algorithm optimizes each column (and corresponding row) of $\mathbf{W}$ iteratively until convergence. For each iteration, $\mathbf{W}$ and $\mathbf{S}$ are partitioned as follows:

$$\mathbf{W} = \begin{pmatrix} \mathbf{W}_{11} & \mathbf{w}_{12} \\ \mathbf{w}_{12}^T & w_{22} \end{pmatrix} \tag{9}$$

$$\mathbf{S} = \begin{pmatrix} \mathbf{S}_{11} & \mathbf{s}_{12} \\ \mathbf{s}_{12}^T & s_{22} \end{pmatrix} \tag{10}$$

for each i-th column, where $\mathbf{W}_{11}$ and $\mathbf{S}_{11}$ are the submatrix of $\mathbf{W}$ and $\mathbf{S}$ (obtained by excluding the i-th row and column), respectively, and $\mathbf{w}_{12}$ and $\mathbf{s}_{12}$ are the i-th column of $\mathbf{W}$ and $\mathbf{S}$ (except the i-th diagonal elements w_{22} and s_{22}), respectively. To optimize $\mathbf{w}_{12}$, $\hat{\boldsymbol{\beta}}$ is obtained using the following equation:

$$\hat{\boldsymbol{\beta}} = \min_{\boldsymbol{\beta}} \frac{1}{2} \|\mathbf{W}_{11}^{\frac{1}{2}}\boldsymbol{\beta} - \mathbf{W}_{11}^{-\frac{1}{2}}\mathbf{s}_{12}\|^2 + \rho\|\boldsymbol{\beta}\|_1 \tag{11}$$

and $\mathbf{w}_{12}$ is replaced by $\mathbf{W}_{11}\hat{\boldsymbol{\beta}}$ in each iteration. A more detailed explanation for the computation can be found in [17,51]. The complexity of the GLASSO algorithm is roughly $O\left(n^3\right)$ for reasonably sparse problems with n vertices [52]. Notice that the block coordinate descent algorithm does not guarantee convergence. Therefore, the reasonable conditions on $\mathbf{S}$ should be required for convergence [53], such as that the covariates in $\mathbf{S}$ is not severely dependent, as mentioned above.

4. The Proposed Method

4.1. GLASSO Applied PLDA

In this paper, we propose a method of applying the GLASSO to the PLDA (denoted as GLASSO-PLDA), where the GLASSO-regularized within-class precision matrix is used to estimate the PLDA model, instead of the original within-class precision matrix. Once the empirical between-class covariance matrix $\mathbf{\Phi}_b$ and the empirical within-class covariance matrix $\mathbf{\Phi}_w$ are estimated, we regularize the empirical within-class precision matrix $\mathbf{\Phi}_w^{-1}$ using the GLASSO (using Equation (8)):

$$\hat{\mathbf{\Phi}}_w^{-1} = \underset{\mathbf{\Theta}}{\operatorname{argmax}}\{\log\det(\mathbf{\Theta}) - \operatorname{tr}(\mathbf{\Phi}_w\mathbf{\Theta}) - \rho\|\mathbf{\Theta}\|_1\} \tag{12}$$

where $\hat{\mathbf{\Phi}}_w^{-1}$ is the regularized within-class precision matrix. The PLDA parameters are then estimated using $\hat{\mathbf{\Phi}}_w^{-1}$ instead of $\mathbf{\Phi}_w^{-1}$. That is, we solve the following eigenvalue equation $\hat{\mathbf{\Phi}}_w^{-1}\mathbf{\Phi}_b\mathbf{A} = \mathbf{A}\mathbf{\Psi}$. All the other processes are the same as those in the original PLDA. In addition, $\mathbf{\Phi}_w^{-1}$ and $\hat{\mathbf{\Phi}}_w^{-1}$ have the same number of parameters. Therefore, the GLASSO-PLDA does not affect the both computation amount and memory usage in evaluation phase. In other words, the computation amount and memory

usage of the GLASSO-PLDA for computing the log-likelihood ratio (7) is the same as those of the conventional PLDA.

The GLASSO-PLDA is based on our assumption that an optimal solution for the within-class precision matrix (as mentioned in Section 1, we focus on the within-class precision matrix only) exists at some point between a high variance (corresponding to low ρ) and high bias (corresponding to high ρ). Notice that there is a trade-off between estimation error and bias, as mentioned in Section 3.2. Naturally, the performance of the likelihood ratio test depends on a good estimation of the PLDA model parameters, which depends on a good estimation of the within-class precision matrix. Therefore, it is important to reduce the estimation error in the empirical within-class precision matrix by finding the optimal value of ρ. The optimal ρ can be considered as what can achieve the best performance on evaluation trials. In practice, it should be found with a validation dataset the domain of which is close to the target domain, because we do not know the target domain in advance.

According to our experiments described in Section 5.3, the GLASSO-PLDA exhibits performance improvement in TD-SV only if a prerequisite is satisfied. The prerequisite is that the within-class covariance and accompanying precision matrix of embeddings should be close to diagonal. Unless the prerequisite is satisfied, the GLASSO converges at a point far from the optimal solution, does not converge, or even fails to estimate. The failure of the estimation is due to that $\hat{\mathbf{\Phi}}_w^{-1}$ is ill-conditioned, which means that $\hat{\mathbf{\Phi}}_w^{-1}$ has the value of infinity or is not a number. We describe the effect of the prerequisite on the performance in Section 5.3.

4.2. Prerequisite: Close-to-Diagonal Within-Class Covariance/Precision Matrix

Even though a common normality assumption exists regarding the embeddings for both PLDA and GLASSO, we do not consider the normality assumption in our research. As mentioned in Section 2.1 and , we used four kinds of speaker embeddings, that is i-vector, d-vector, r-vector, and x-vector. Among these embeddings, only the i-vector satisfies the normality assumption. The i-vector has demonstrated performance improvement with both PLDA and GLASSO-PLDA. However, all deep speaker embeddings used in our research, which have no assumption of normality, also exhibited performance improvement with the PLDA. In contrast, the deep speaker embeddings show performance degradation or failure of the estimation with the GLASSO-PLDA. The experimental results will be described in Section 5.3. Some research has investigated making the deep speaker embeddings follow a normal distribution [54–56]. However, we found that deep speaker embeddings obtained using these methods are still not suitable for the GLASSO-PLDA (see Appendix A), and the embeddings obtained by [55] were corrupted and lost discriminative power. From this result, we conclude that the absence of the normality assumption on the embeddings does not matter, and regard all embeddings used in our experiments as following a multivariate normal distribution.

We focus on the diagonality of the within-class covariance/precision matrix. Close-to-diagonal covariance matrix of normal variables relates to that the covariates are not severely dependent, which is the condition for the GLASSO, as mentioned in Section 3.2. In addition, the closer the covariance matrix is to the diagonal, the closer the accompanying precision matrix is somewhat to the diagonal. For a quantitative comparison, we define the degree of the diagonality (denote as δ) of the matrix $\mathbf{\Theta}$ as the ratio of the L_1 norm of the covariance $\|\mathbf{\Theta}\|_1$ to the L_1 norm of the diagonal elements $\|\mathrm{diag}(\mathbf{\Theta})\|_1$:

$$\delta = \frac{\|\mathrm{diag}(\mathbf{\Theta})\|_1}{\|\mathbf{\Theta}\|_1}. \tag{13}$$

The higher δ means that $\mathbf{\Theta}$ is more close to the diagonal matrix, and δ satisfies $0 \leq \delta \leq 1$. As mentioned above, the prerequisite of the GLASSO-PLDA is that the empirical within-class covariance matrix $\mathbf{\Phi}_w$ and accompanying precision matrix $\mathbf{\Phi}_w^{-1}$ should be close to diagonal. Therefore, it is important to check whether $\mathbf{\Phi}_w$ and $\mathbf{\Phi}_w^{-1}$ of each kind of embedding are close to diagonal. Figures 1 and 2 illustrate $\mathbf{\Phi}_w$ and $\mathbf{\Phi}_w^{-1}$ of four kinds of embeddings, respectively. The diagonality δ of $\mathbf{\Phi}_w$ and $\mathbf{\Phi}_w^{-1}$

is 0.2179 (Figure 1a) and 0.2 (Figure 2a) for the i-vectors, 0.0262 (Figure 1b) and 0.0696 (Figure 2b) for the d-vectors, 0.0522 (Figure 1c) and 0.0794 (Figure 2c) for the r-vectors, and 0.019 (Figure 1d) and 0.0916 for the x-vectors (Figure 2d), respectively. As you can see, only $\mathbf{\Phi}_w$ and $\mathbf{\Phi}_w^{-1}$ of the i-vector (Figures 1a and 2a) are close to diagonal, and the others (Figures 1b–d and 2b–d) are far from diagonal. The covariances of the i-vector are assumed to be zero, that is, the features consisting of the i-vector are independent. In most cases, the empirical covariance matrix is close to the diagonal matrix due to the estimation error. Therefore, both $\mathbf{\Phi}_w$ and $\mathbf{\Phi}_w^{-1}$ of the i-vectors are close to diagonal. In contrast, both $\mathbf{\Phi}_w$ and $\mathbf{\Phi}_w^{-1}$ of all deep speaker embeddings are far from diagonal, even considering the estimation error. In practice, no assumption of the diagonality exists for the covariance of all deep speaker embeddings, and all kinds of covariances of all deep speaker embeddings are far from diagonal. It means that only i-vector satisfies the prerequisite of the GLASSO-PLDA. The characteristics in the two covariance structures of the i-vector and deep speaker embeddings lead to different results in applying the regularization method, the GLASSO, to the PLDA. In practice, as described in Section 5.3, only the i-vector, which satisfies the prerequisite, showed the performance improvements when using the GLASSO-PLDA. All the deep speaker embeddings, which have $\mathbf{\Phi}_w$ and $\mathbf{\Phi}_w^{-1}$ that are far from diagonal, showed the performance degradations when using the GLASSO-PLDA.

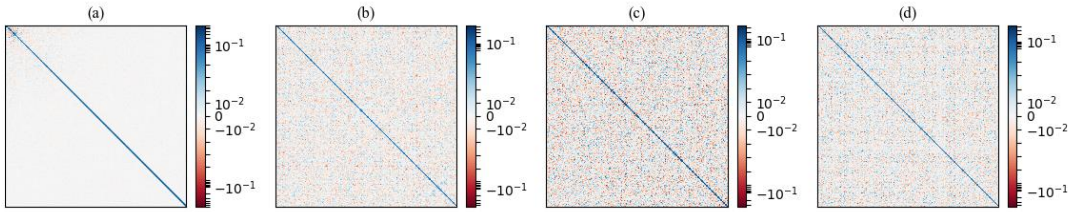

Figure 1. The empirical within-class covariance matrix of (**a**) i-vectors, (**b**) d-vectors, (**c**) r-vectors, and (**d**) x-vectors.

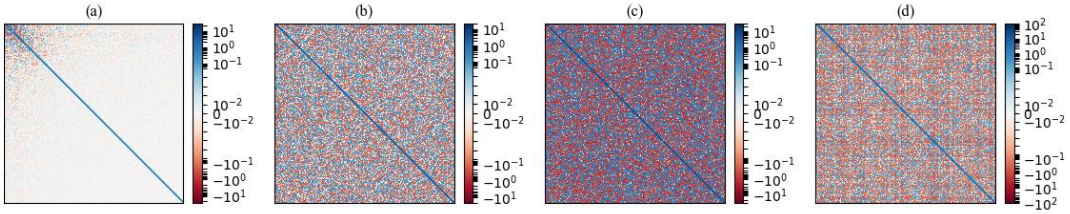

Figure 2. The empirical within-class precision matrix of (**a**) i-vectors, (**b**) d-vectors, (**c**) r-vectors, and (**d**) x-vectors.

Our goal of the use of the GLASSO is to reduce the estimation errors in the precision matrix. We aim to remove small noises in the precision matrix raised by the estimation errors. In the case of close-to-diagonal precision, like i-vector, the GLASSO-PLDA exhibited the performance improvement. Therefore, the GLASSO can remove many of the small noises effectively when the precision matrix is close to diagonal. However, in the case of far-from-diagonal precision, like deep speaker embeddings, the GLASSO-PLDA demonstrated the performance degradation or failure of the estimation of the PLDA parameters. The GLASSO also shrinks the principal elements that have relatively larger values toward zero, on the contrary a too-large bias toward zero, when the precision matrix is far from diagonal. Thus, the loss from a high bias is greater than the gain from the low estimation error for that case. This result may imply that the GLASSO can improve the performance only if the covariance/precision matrix is close to diagonal.

We assume that deep speaker embeddings also show the performance improvements with the GLASSO-PLDA, like i-vector, if we make the within-class covariance matrix $\mathbf{\Phi}_w$ and the accompanying precision matrix $\mathbf{\Phi}_w^{-1}$ close to diagonal. To make the $\mathbf{\Phi}_w$ and $\mathbf{\Phi}_w^{-1}$ of the deep speaker embeddings close to diagonal, we orthogonalize the deep speaker embeddings using principal component analysis

(PCA). The PCA transform makes the total covariance close to diagonal through orthogonalization. The diagonalization of the total covariance has the effect of making both within-class covariance and the accompanying precision matrix close to diagonal. We use the orthogonalized deep speaker embeddings for the GLASSO-PLDA instead of the original embeddings (denote as PCA-GLASSO-PLDA). We do not reduce the dimensionality in the PCA transform to avoid information loss. It is important to check whether $\mathbf{\Phi}_w$ and $\mathbf{\Phi}_w^{-1}$ actually become closer to diagonal after the PCA transform. Figures 3 and 4 reveal the within-class covariance and within-class precision matrix estimated from the transformed embeddings (denoted as $\mathbf{\Phi}_{w\_pca}$ and $\mathbf{\Phi}_{w\_pca}^{-1}$), respectively. After applying the PCA transform, all the within-class covariance/precision matrix (corresponding to $\mathbf{\Phi}_{w\_pca}$ and $\mathbf{\Phi}_{w\_pca}^{-1}$, respectively) become closer to diagonal. The diagonality δ of $\mathbf{\Phi}_{w\_pca}$ and $\mathbf{\Phi}_{w\_pca}^{-1}$ is 0.1509 (Figure 3a) and 0.1935 (Figure 4a) for the d-vectors, 0.2262 (Figure 3b) and 0.364 (Figure 4b) for the r-vectors, and 0.1728 (Figure 3c) and 0.1827 (Figure 4c) for the x-vectors, respectively. As described in Section 5.3, the orthogonalized deep speaker embeddings exhibited performance improvements with the GLASSO-PLDA (corresponding to the PCA-GLASSO-PLDA), as in the i-vector. Therefore, the empirical within-class covariance/precision matrix should be close to diagonal for the GLASSO regularization. Thus, if we find the transformation that makes the within-class covariance/precision matrix close to diagonal, the performance of the PLDA can improve by soft-thresholding the noises in the accompanying empirical within-class precision matrix using GLASSO regularization.

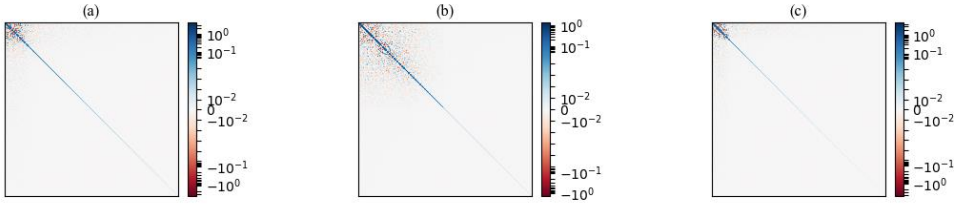

Figure 3. The empirical within-class covariance matrix of (**a**) d-vectors, (**b**) r-vectors, and (**c**) x-vectors. These matrices are estimated from the principal component analysis (PCA)-transformed embeddings.

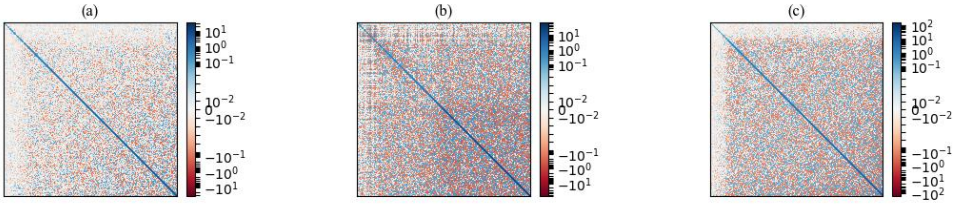

Figure 4. The empirical within-class precision matrix of (**a**) d-vectors, (**b**) r-vectors, and (**c**) x-vectors. These matrices are estimated from the PCA-transformed embeddings.

5. Experiments

5.1. Database

For the evaluation of the task of TD-SV, we used parts 1 and 2 of the robust speaker recognition (RSR) 2015 dataset [57]. Both parts consist of utterances from 300 speakers and are divided into background (50 male and 47 female speakers), development (50 male and 47 female speakers), and evaluation (57 male and 49 female speakers) subsets. The speakers from parts 1 and 2 are the same, and no speaker overlap exists across the subsets. For each part, each speaker utters 30 different phrases in nine different sessions. The average duration of utterances is 3.2 s for part 1, and 1.99 s for part 2, including silence. We used the background set to build gender-independent models, the development set for validation on gender-independent trials, and the evaluation set to evaluate the performance of the proposed system on the gender-dependent trials. Figures 1–4 were based on RSR part 1.

5.2. Experimental Setup

We extracted 400-dimensional i-vectors. For each utterance, 25-ms frames were extracted at 10-ms intervals. Preprocessing was performed for all frames in the following order: removing the direct current (DC) offset, pre-emphasis filtering with a coefficient of 0.97, and applying a Hamming window. A 60-dimensional (19 static + energy + delta + acceleration) mel-frequency cepstral coefficients (MFCCs) was extracted from each preprocessed frame. We applied utterance-level cepstral mean normalization (CMN) using a 300-frame sliding window, then voice activity detection (VAD) to remove silent frames. A gender-independent Gaussian mixture model universal background model (GMM-UBM) consists of 1024 mixture components with diagonal covariance, which was trained for 10 iterations. A gender-independent 400-dimensional i-vector extractor was trained for five iterations. Length normalization was applied to i-vectors.

The common configurations for extracting deep speaker embeddings were as follows. A 40-dimensional mel-filterbank feature was extracted from each preprocessed frame. The preprocessing is the same as that used for i-vector extraction. As in the i-vector extraction, the CMN and VAD were applied to the mel-filterbank feature. Except for extracting d-vectors, the sequences of mel-filterbank features of the utterances were truncated or padded along the time axis to have lengths of 250 and 150 for parts 1 and 2, respectively. For extracting d-vectors, the sequences of mel-filterbank features were handled by using distortion-free method [58]. The speaker-and-phrase discriminative networks have two softmax classifiers: one is the speaker classifier, and the other is the phrase classifier. Unless otherwise noted, all weights of the networks were initialized from the Glorot normal distribution [59] and those of the classifiers were orthogonalized with no bias. The AMSGrad [60], a variant of the Adam [61] optimizer, was used to minimize the cross-entropy loss with a learning rate of 10^{-3}. We trained 100 epochs with a mini-batch size of 32 and selected the best model by validation.

The extraction process for the d-vector is the same as in [58]. We extracted a 512-dimensional d-vector from a 2-layer LSTM. Each layer of the LSTM has 512 units. On top of the LSTM is a self-attention [62] layer with four attention heads, followed by a batch normalization [63] layer. The recurrent weights of the LSTM were orthogonalized. The biases of the forget gate of the LSTM were initialized to 1 [64], and the other biases were initialized to 0. The d-vector is the result of the batch normalization of the sum of the output of the attention layer.

We extracted the 256-dimensional r-vector from SE-ResNet34. Except for the SE blocks, SE-ResNet34 has the same structure as ResNet34 in [37] up to the last residual block. The layers were stacked on top of the last residual block in the following order: the statistical pooling layer to compute the mean and standard deviation along the time axis for each channel, the flattening layer, the 256-dimensional fully connected layer followed by a batch normalization layer, and the classifiers. We used the batch-normalized output vector as the r-vector.

We extracted the 512-dimensional x-vector. The architecture of the x-vector network is the same as the standard DNN in [29]. We applied batch normalization after each activation layer. The x-vector is the output of the layer segment 2 in [29].

The PLDA model was trained using each embedding for 10 iterations. For the GLASSO-PLDA, we first estimated the GLASSO-PLDA models with different values of ρ in the range 0 to 0.5 at intervals of 0.0005. The maximum number of iterations of the GLASSO was set to 100, and the tolerance for convergence (corresponding to the duality gap [65]) was set to 10^{-4}. We then computed equal error rates (EERs) for each GLASSO-PLDA model on the validation trials. The EER is the rate when the false positive rate and false negative rate are equal. We use EER as the performance metric for our experiments. For evaluation, we selected the best GLASSO-PLDA model based on the EERs of the validation trials.

All the acoustic features and linear models (corresponding to GMM, i-vector extractor, and PLDA) were implemented using the Kaldi toolkit. All the DNN-based models (corresponding to the extractors of all deep speaker embeddings) were implemented using PyTorch [66]. The GLASSO algorithm was implemented in scikit-learn [67].

5.3. Results

We first evaluated the EERs of the proposed method only with the i-vector, which satisfies the prerequisite (close-to-diagonal within-class covariance matrix) of the GLASSO-PLDA. Figure 5 displays the EERs on the validation trials of the RSR 2015 part 1 (Figure 5a) and part 2 (Figure 5b). The dashed red line depicts the EERs of the original PLDA (corresponding to the baseline), and the solid blue line depicts the EERs of the GLASSO-PLDA (corresponding to the proposed method). We confirmed that the GLASSO converged in all conditions, and the performances were improved with the proposed method. The same trend also can be observed in the evaluation trials. Figure 6 reveals the EERs on the evaluation trials of the RSR 2015 part 1 (Figure 6a,c) and part 2 (Figure 6b,d). Except for the interval of the regularization parameter $\rho > 0.0535$ on the male evaluation trials of RSR 2015 part 2 (Figure 6b), the proposed method also demonstrated performance improvements on the evaluation trials. Therefore, the proposed GLASSO-PLDA can improve the performance by detecting the optimum (sparse within-class precision matrix) when the sparse assumption of the true within-class precision matrix holds, and the prerequisite is satisfied.

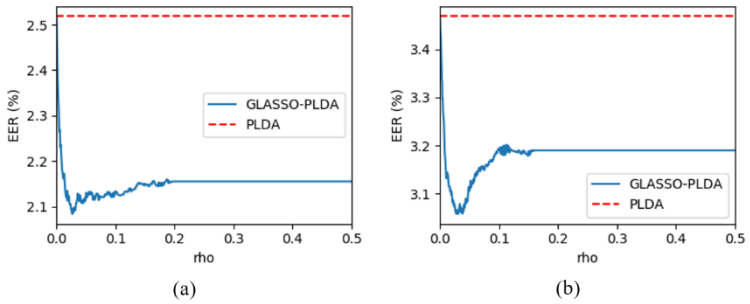

Figure 5. The equal error rates (EERs) (%) of the probabilistic linear discriminant analysis (PLDA) (dashed red line) and graphical least absolute shrinking and selection operator (GLASSO)-PLDA (solid blue line) according to ρ on the validation trials of the (**a**) RSR 2015 part 1 and (**b**) RSR 2015 part 2 datasets.

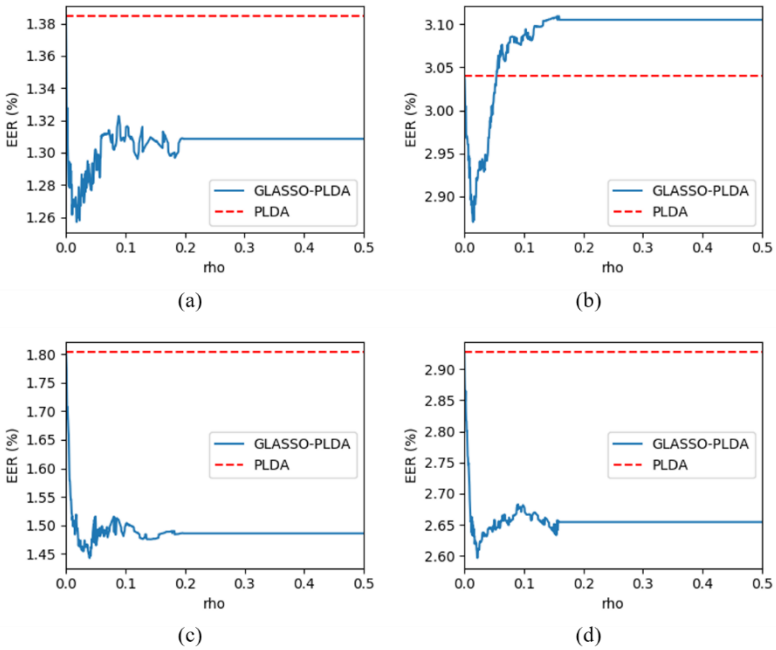

Figure 6. The EERs (%) of the PLDA (dashed red line) and GLASSO-PLDA (solid blue line) according to ρ on the evaluation trials of the RSR 2015 part 1 and part 2 datasets: (**a**) male trials of the part 1, (**b**) male trials of the part 2, (**c**) female trials of the part 1, and (**d**) female trials of the part 2.

Next, we performed experiments using deep speaker embeddings, which generally do not satisfy the prerequisite of the GLASSO-PLDA. Figure 7 lists the EERs of the validation trials of RSR 2015 part 1 for the (a) d-vector, (b) r-vector, and (c) x-vector. The dashed red line depicts the EERs of the original PLDA, and the solid blue and green lines depict the EERs of the GLASSO-PLDA and PCA-GLASSO-PLDA, respectively. Unlike for the i-vector, the GLASSO did not converge because the within-class covariance/precision matrices of all deep speaker embedding are far from diagonal. For this reason, the GLASSO-PLDA demonstrated performance degradation for all deep speaker embeddings, as mentioned in Section 4. In contrast, the GLASSO with PCA converged by making the within-class covariance/precision matrix close to diagonal. As a result, the PCA-GLASSO-PLDA displayed performance improvements for all deep speaker embeddings in all the conditions. The same trends can be observed in the evaluation trials of RSR 2015 part 1 (Figure 8), the validation trials of RSR 2015 part 2 (Figure 9), and the evaluation trials of RSR 2015 part 2 (Figure 10). There are no EER graphs of the GLASSO-PLDA for the x-vector on the trials of RSR 2015 part 2 (there is no solid blue line in Figures 9c and 10c,f), due to the failure of the estimation of the GLASSO-PLDA model caused by the failure of GLASSO regularization. These results indicate that the GLASSO-PLDA can improve performance when the within-class covariance/precision matrix is close to diagonal.

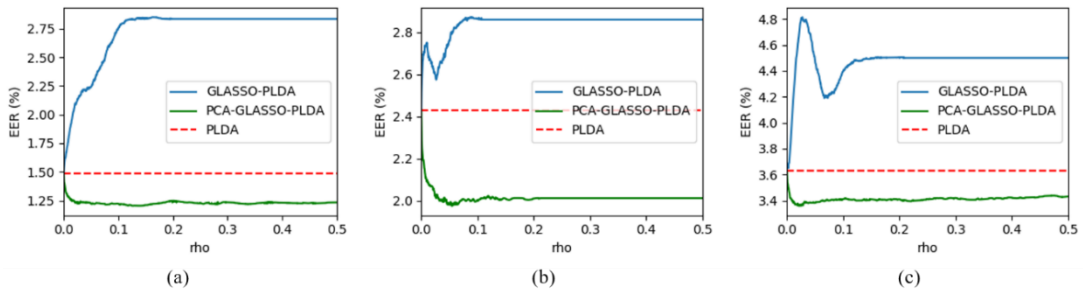

Figure 7. The EERs (%) of the PLDA (dashed red line), GLASSO-PLDA (solid blue line), and PCA-GLASSO-PLDA (solid green line) according to ρ on the validation trials of RSR 2015 part 1 for the (**a**) d-vector, (**b**) r-vector, and (**c**) x-vector.

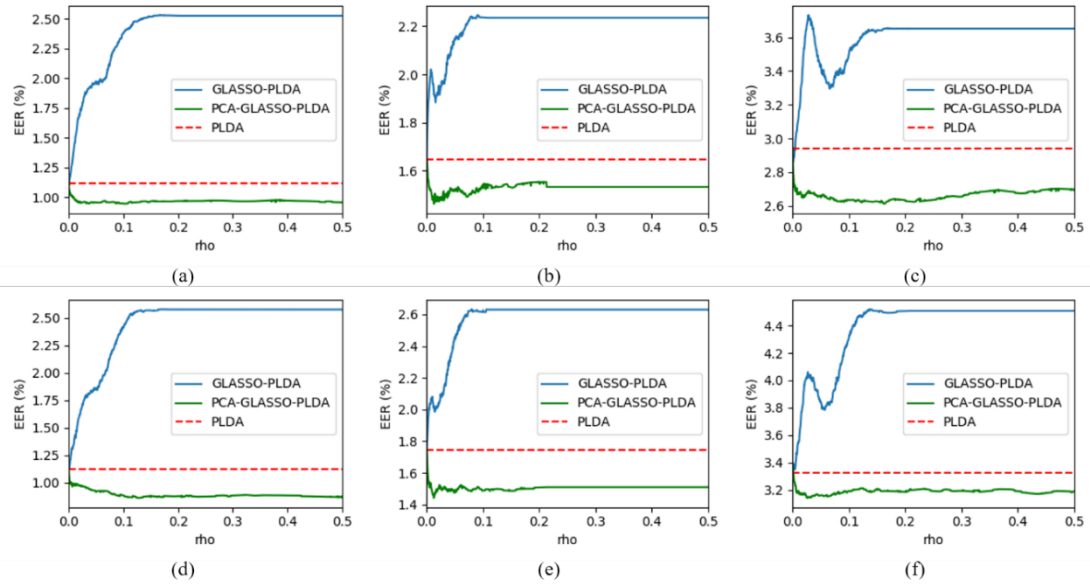

Figure 8. The EERs (%) of the PLDA (dashed red line), GLASSO-PLDA (solid blue line), and PCA-GLASSO-PLDA (solid green line) according to ρ on the evaluation trials of RSR 2015 part 1: (**a**) male trials for d-vector, (**b**) male trials for r-vector, (**c**) male trials for x-vector, (**d**) female trials for d-vector, (**e**) female trials for r-vector, and (**f**) female trials for x-vector.

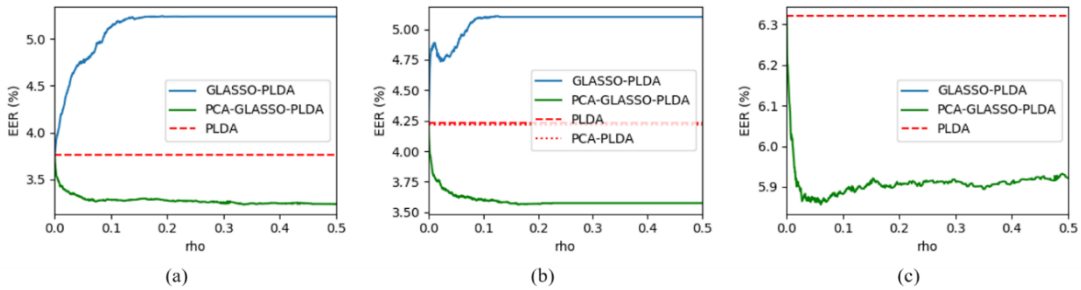

Figure 9. The EERs (%) of the PLDA (dashed red line), GLASSO-PLDA (solid blue line), PCA-PLDA (dotted red line), and PCA-GLASSO-PLDA (solid green line) according to ρ on the validation trials of RSR 2015 part 2 for the (**a**) d-vector, (**b**) r-vector, and (**c**) x-vector; EERs of the PCA-PLDA are not shown if the PLDA and PCA-PLDA have the same EER.

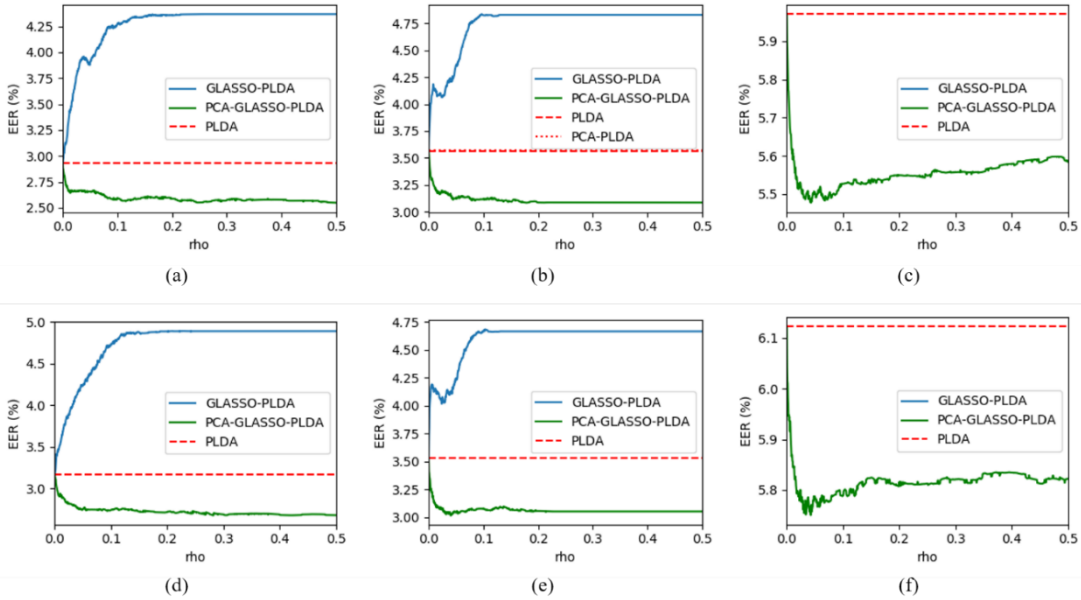

Figure 10. The EERs (%) of the PLDA (dashed red line), GLASSO-PLDA (solid blue line), PCA-PLDA (dotted red line), and PCA-GLASSO-PLDA (solid green line) according to ρ on the evaluation trials of RSR 2015 part 2: (**a**) male trials for d-vector, (**b**) male trials for r-vector, (**c**) male trials for x-vector, (**d**) female trials for d-vector, (**e**) female trials for r-vector, and (**f**) female trials for x-vector. EERs of the PCA-PLDA are not shown if the PLDA and PCA-PLDA have the same EER.

Tables 1 and 2 summarize the EERs of the baseline and proposed methods for each embedding on RSR 2015 parts 1 and 2, respectively. The proposed method is the GLASSO-PLDA for the i-vector and the PCA-GLASSO-PLDA for deep speaker embeddings. For the proposed method, the regularization parameter ρ was set to that of the lowest EER of the validation trials. In RSR 2015 part 1 trials, the proposed method revealed relative EER reductions of approximately 7% (for the x-vector) to 19% (for the d-vector) on the validation trials, reductions of approximately 7% (for the i-vector) to 13% (for the d-vector) on the male evaluation trials, and reductions of approximately 5% (for the x-vector) to 23% (for the d-vector) on the female evaluation trials. In RSR 2015 part 2, the proposed method similarly revealed relative EER reductions of approximately 7% (for the x-vector) to 16% (for the r-vector) on the validation trials, reductions of approximately 3% (for the i-vector) to 13% (for the r-vector) on the male evaluation trials, and reductions of approximately 6% (for the x-vector) to 15% (for the d-vector) on the female evaluation trials. Finally, we performed a score-level fusion of all proposed systems for each dataset. The weights for each score were estimated using logistic regression with the scores from the validation trials. With score-level fusion, we achieved significant performance improvements.

Table 1. EERs (%) of the baseline and proposed methods for each embedding on the validation and evaluation trials of RSR 2015 part 1.

Method		Embedding	EER (%)		
			Validation	Evaluation	
				Male	Female
Baseline		i-vector	2.5183	1.3845	1.8032
		d-vector	1.4874	1.1144	1.1249
		r-vector	2.4317	1.6478	1.7459
		x-vector	3.631	2.938	3.3259
Proposed with ρ of	0.027	i-vector	2.0842	1.2811	1.4627
	0.134	d-vector	1.2036	0.9644	0.8651
	0.055	r-vector	1.9761	1.5127	1.4949
	0.025	x-vector	3.3587	2.6728	3.1442
	Score fusion		**1.0089**	**0.7532**	**0.6875**

Table 2. EERs (%) of the baseline and proposed methods for each embedding on the validation and evaluation trials of RSR 2015 part 2. (* means the EER of the PCA-PLDA).

Method		Embedding	EER (%)		
			Validation	Evaluation	
				Male	Female
Baseline		i-vector	3.4689	3.0395	2.9271
		d-vector	3.7577	2.9342	3.1697
		r-vector	4.2289 4.2186 *	3.5635 3.5714 *	3.5306
		x-vector	6.3205	5.9704	6.1229
Proposed with ρ of	0.027	i-vector	3.0575	2.9434	2.6338
	0.134	d-vector	3.2286	2.5765	2.6918
	0.055	r-vector	3.5621	3.1058	3.0593
	0.025	x-vector	5.8561	5.509	5.7785
	Score fusion		**2.2855**	**1.7897**	**1.7363**

6. Discussion

In this section, we first evaluate the proposed method in TI-SV tasks. Next, we compare the performance of the proposed method with the banding method, which is more simple method than the GLASSO for making matrix sparse.

6.1. Evaluation in Text-Independent Speaker Verification

We explain the difference between TD-SV and TI-SV in terms of variability. The total variability of samples $\mathbf{\Sigma}$ can be decomposed as the sum of the between-class variability $\mathbf{\Sigma}_b$ and the within-class variability $\mathbf{\Sigma}_w$; that is, $\mathbf{\Sigma} = \mathbf{\Sigma}_b + \mathbf{\Sigma}_w$. In TD-SV, $\mathbf{\Sigma}_b$ contains the speaker variability $\mathbf{\Sigma}_{spk}$ and phrase variability $\mathbf{\Sigma}_{phr}$, and $\mathbf{\Sigma}_w$ contains other variabilities, such as channel variability $\mathbf{\Sigma}_{ch}$ and residual variability $\mathbf{\Sigma}_\epsilon$. Therefore, $\mathbf{\Sigma}_b = \mathbf{\Sigma}_{spk} + \mathbf{\Sigma}_{phr}$ and $\mathbf{\Sigma}_w = \mathbf{\Sigma}_{ch} + \mathbf{\Sigma}_\epsilon$. In TI-SV, in contrast, $\mathbf{\Sigma}_b$ contains only $\mathbf{\Sigma}_{spk}$, and $\mathbf{\Sigma}_w$ contains other variabilities $\mathbf{\Sigma}_{phr}$, $\mathbf{\Sigma}_{ch}$, and $\mathbf{\Sigma}_\epsilon$. Therefore, $\mathbf{\Sigma}_b = \mathbf{\Sigma}_{spk}$ and $\mathbf{\Sigma}_w = \mathbf{\Sigma}_{phr} + \mathbf{\Sigma}_{ch} + \mathbf{\Sigma}_\epsilon$. To summarize, the difference between TD-SV and TI-SV is whether $\mathbf{\Sigma}_w$ contains $\mathbf{\Sigma}_{phr}$ or not. For TD-SV, $\mathbf{\Sigma}_w = \mathbf{\Sigma}_{ch} + \mathbf{\Sigma}_\epsilon$ does not contain phrase variability $\mathbf{\Sigma}_{phr}$. For TI-SV, in contrast, $\mathbf{\Sigma}_w = \mathbf{\Sigma}_{phr} + \mathbf{\Sigma}_{ch} + \mathbf{\Sigma}_\epsilon$ contains $\mathbf{\Sigma}_{phr}$. Therefore, the GLASSO regularization of the empirical within-class precision matrix $\mathbf{\Phi}_w^{-1}$ does not affect $\mathbf{\Sigma}_{phr}$ for TD-SV but affects $\mathbf{\Sigma}_{phr}$ for TI-SV.

For the evaluation of the task of TI-SV, we used the VoxCeleb1 dataset [68], which is divided into development (148,642 utterances from 1211 speakers) and test (4874 utterances from 40 speakers) sets. The average duration of utterances is 8.2 s. We split the development set into training (144,990 utterances from 1183 speakers) and validation (3652 utterances from 28 speakers; speaker id10357-id10384) sets.

We created 37,904 validation (10,492 target and 27,412 impostor) trials using the validation set. We used the training set to build gender-independent models, the validation set for validation, and the test set to evaluate the performance of the proposed system. We used the VoxCeleb1 dataset to build and evaluate the i-vector-based system only, which satisfies the prerequisite of the GLASSO-PLDA. The configuration for extracting i-vectors was the same as that described in Section 5.2.

Figure 11 displays the EERs on the validation (Figure 11a) and evaluation (Figure 11b) trials of the VoxCeleb dataset. Unlike the results in the TD-SV tasks, the performances were degraded with the proposed method in the TI-SV tasks, even though the prerequisite was satisfied and the GLASSO converged. These results mean that the proposed method is suitable for only TD-SV tasks, where $\mathbf{\Sigma}_w = \mathbf{\Sigma}_{ch} + \mathbf{\Sigma}_\epsilon$ does not contain phrase variability $\mathbf{\Sigma}_{phr}$. In other words, the true within-class precision matrix is sparse only if $\mathbf{\Sigma}_w$ does not contain $\mathbf{\Sigma}_{phr}$. The optimums of both channel variability $\mathbf{\Sigma}_{ch}$ and residual variability $\mathbf{\Sigma}_\epsilon$ have conditional independence structure, but that of $\mathbf{\Sigma}_{phr}$ does not. Therefore, the sparse assumption of $\mathbf{\Phi}_w^{-1}$ does not hold and the proposed method is not suitable for TI-SV tasks, where $\mathbf{\Sigma}_w = \mathbf{\Sigma}_{phr} + \mathbf{\Sigma}_{ch} + \mathbf{\Sigma}_\epsilon$ contains phrase variability $\mathbf{\Sigma}_{phr}$.

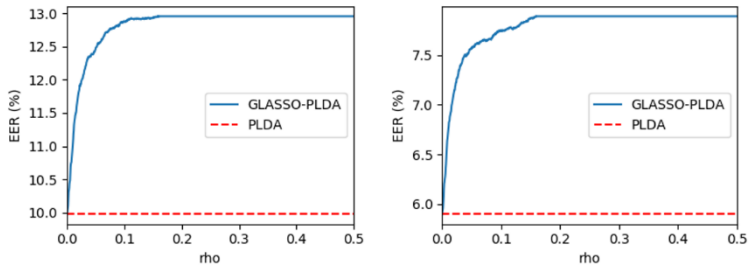

Figure 11. The EERs (%) of the PLDA (dashed red line) and GLASSO-PLDA (solid blue line) according to ρ on the (**a**) validation and (**b**) evaluation trials of the VoxCeleb dataset.

6.2. Comparison with Matrix Banding

In this section, we describe another option for our proposed method, matrix banding. Matrix banding is a simple method to make arbitrary matrix a band matrix [69]. The band matrix is a sparse matrix all non-zero elements of which are in at most some consecutive and diagonally bordered bands including main diagonal. In other words, all out-of-band elements of the band matrix are zero. Therefore, the matrix banding is the method that can make arbitrary matrix sparse with less computational burden than the GLASSO. We compare the performances of the proposed method (GLASSO-PLDA) with those of the matrix banding-based PLDA (denoted as banding-PLDA), in which the within-class precision matrix is regularized with the banding rather than the GLASSO. There is no reason for using the GLASSO-PLDA if the banding-PLDA generally shows better performances than the GLASSO-PLDA.

Tables 3 and 4 summarize the EERs of the proposed method and banding-PLDA for each embedding on RSR 2015 part 1 and 2, respectively. For the both proposed method and banding-PLDA, all deep speaker embeddings were orthogonalized using the PCA. Therefore, the EERs of the proposed method in Tables 3 and 4 are the same as those in Tables 1 and 2, respectively. For the banding-PLDA, we constrained the bands to symmetric because within-class precision matrix must be symmetric. k means the bandwidth for one side (left or right). The total bandwidth is $2k + 1$ because the bands are constrained to be symmetric. $k = 0$ means the diagonal matrix. The GLASSO-PLDA generally exhibited better performances than the banding-PLDA. In TD-SV, therefore, the GLASSO, which is based on the L_1-penalized Gaussian log-likelihood, is more effective than the banding.

Table 3. EERs (%) of the proposed and banding methods for each embedding on the validation and evaluation trials of RSR 2015 part 1.

Method		Embedding	EER (%)		
			Validation	Evaluation	
				Male	Female
Proposed		i-vector	**2.0842**	**1.2811**	**1.4627**
		d-vector	**1.2036**	**0.9644**	0.8651
		r-vector	**1.9761**	**1.5127**	**1.4949**
		x-vector	3.3587	2.6728	**3.1442**
Banding-PLDA with k of	0	i-vector	2.1904	1.3757	1.5431
	0	d-vector	1.2398	1.0033	**0.8529**
	0	r-vector	2.0544	1.5728	1.5796
	11	x-vector	**3.3551**	**2.6337**	3.2162

Table 4. EERs (%) of the proposed and banding methods for each embedding on the validation and evaluation trials of RSR 2015 part 2.

Method		Embedding	EER (%)		
			Validation	Evaluation	
				Male	Female
Proposed		i-vector	**3.0575**	**2.9434**	**2.6338**
		d-vector	**3.2286**	2.5765	2.6918
		r-vector	**3.5621**	**3.1058**	**3.0593**
		x-vector	**5.8561**	**5.509**	**5.7785**
Banding-PLDA with k of	0	i-vector	3.173	3.1342	2.669
	0	d-vector	3.2393	**2.5571**	**2.6877**
	0	r-vector	3.7074	3.1803	3.1319
	9	x-vector	5.9019	5.5572	5.8016

7. Conclusions

In this paper, we improved the conventional PLDA by proposing the GLASSO-PLDA, in which the GLASSO-regularized within-class precision matrix was used to estimate the PLDA model. The GLASSO makes empirical within-class precision matrices sparse. It has the effects of reducing the estimation error in the within-class precision matrices and of reflecting a conditional independence structure on the variables. We assumed that the empirical within-class precision matrices would have large errors due to the limited amount of data and expected that the reduction of the estimation error would lead to performance improvement. From the experimental results on the trials on a public database RSR 2015 parts 1 and 2, we found that the GLASSO-PLDA demonstrated the performance improvement when the within-class covariance/precision matrix of the embedding is close to diagonal. That is, the performance of the PLDA can be improved using GLASSO regularization on the empirical within-class precision matrix when the covariance/precision matrix is close to diagonal. With system fusion, we also have achieved significant performance improvements in the task of TD-SV. The GLASSO-PLDA can be directly applied to the TD-SV systems based on the conventional PLDA without changing the structure of the systems. Therefore, it can be applied to any kinds of applications that uses the TD-SV systems based on the PLDA.

In the future, we will apply the GLASSO-PLDA onto noisy condition, where the within-class variability $\mathbf{\Sigma}_w = \mathbf{\Sigma}_{ch} + \mathbf{\Sigma}_\epsilon + \mathbf{\Sigma}_{noise}$ contains not only both channel variability $\mathbf{\Sigma}_{ch}$ and residual variability $\mathbf{\Sigma}_\epsilon$ but also noise variability $\mathbf{\Sigma}_{noise}$. Notice that the within-class variability on clean condition, $\mathbf{\Sigma}_w = \mathbf{\Sigma}_{ch} + \mathbf{\Sigma}_\epsilon$, does not contain $\mathbf{\Sigma}_{noise}$. In detail, we will evaluate the performance of the GLASSO-PLDA based TD-SV system that is trained using only clean utterances on various noisy conditions, which is the setting close to the environment where real applications are used. This experiment is to confirm whether

the GLASSO-PLDA can compensate for the $\mathbf{\Sigma}_w = \mathbf{\Sigma}_{ch} + \mathbf{\Sigma}_\epsilon + \mathbf{\Sigma}_{noise}$ without noisy training utterances. As shown in our experiments, the optimums of both $\mathbf{\Sigma}_{ch}$ and $\mathbf{\Sigma}_\epsilon$ have conditional independence structure. If the optimum of $\mathbf{\Sigma}_{noise}$ has also conditional independence structure; that is, if the sparse assumption of the within-class precision matrix also holds in noisy condition, the GLASSO-PLDA would bring performance improvements in the condition. It means that the GLASSO-PLDA can compensate for the noise variability even without using noisy utterances.

Author Contributions: Conceptualization, S.-H.Y.; methodology, J.-J.J.; formal analysis, S.-H.Y. and J.-J.J.; investigation, S.-H.Y.; writing—original draft preparation, S.-H.Y.; writing—review and editing, J.-J.J. and H.-J.Y.; and supervision, H.-J.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Projects for Research and Development of Police Science and Technology under the Center for Research and Development of Police Science and Technology and the Korean National Police Agency funded by the Ministry of Science, ICT and Future Planning (PA-J000001-2017-101).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Appendix A

In this appendix, we describe the effect of making the deep speaker embeddings follow a normal distribution [54–56]. Figures A1 and A2 list the EERs of the PLDA and Gaussian-constrained (GC)-GLASSO-PLDA on the trials of RSR 2015 parts 1 and 2, respectively. We used d-vectors as speaker embeddings, which generally exhibited good performance in our experiments. The GC-GLASSO-PLDA means that the embeddings were extracted from the DNN trained using the GC training method [54–56]. However, the GC embeddings are still unsuitable for GLASSO-PLDA, because the GLASSO-PLDA with the GC embeddings actually showed performance degradations, as shown in Figures A1 and A2. If the GC embeddings are suitable for the GLASSO-PLDA, the GLASSO-PLDA with the GC embeddings would show performance improvements. Therefore, we can claim that the GC embeddings are not suitable for the GLASSO-PLDA, regardless of whether the GC embeddings actually follow a normal distribution. It means that the GC training cannot make the embeddings follow a Gaussian distribution, or the prerequisite of the GLASSO-PLDA may not be related to the normality assumption of embeddings.

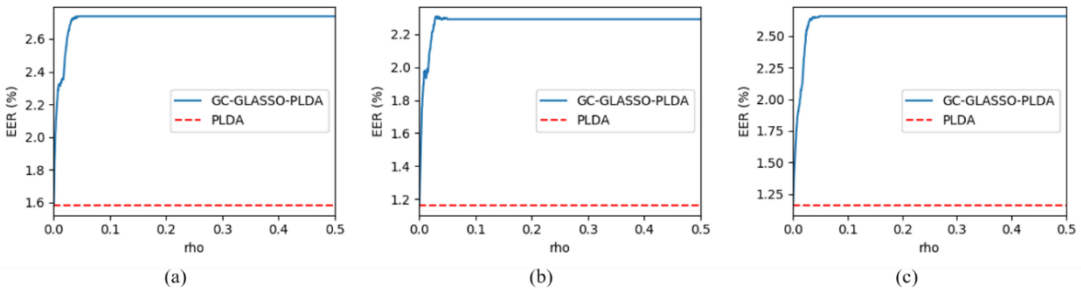

Figure A1. The EERs (%) of the PLDA (dashed red line) and GC-GLASSO-PLDA (solid blue line) according to ρ on the (**a**) validation, (**b**) male evaluation, and (**c**) female evaluation trials of RSR 2015 part 1 for the d-vector.

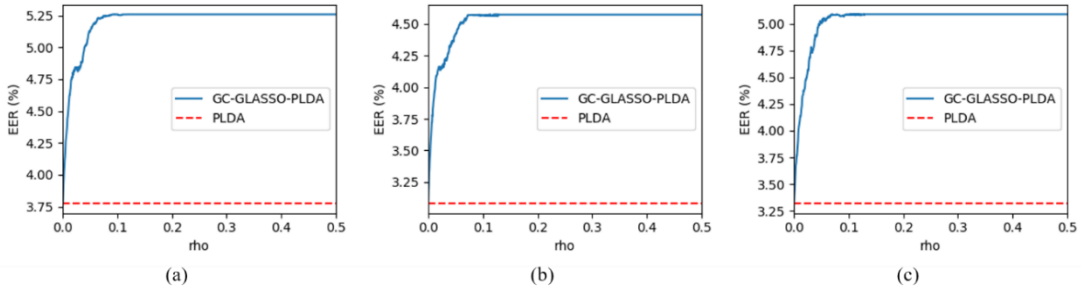

Figure A2. The EERs (%) of the PLDA (dashed red line) and GC-GLASSO-PLDA (solid blue line) according to ρ on the (**a**) validation, (**b**) male evaluation, and (**c**) female evaluation trials of RSR 2015 part 2 for the d-vector.

References

1. Hamidi, M.; Satori, H.; Laaidi, N.; Satori, K. Conception of speaker recognition methods: A review. In Proceedings of the 2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), Meknes, Morocco, 19–20 March 2020; pp. 1–6.
2. Kinnunen, T.; Li, H. An overview of text-independent speaker recognition: From features to supervectors. *Speech Commun.* **2010**, *52*, 12–40. [CrossRef]
3. Access the Google Assistant with Your Voice. Available online: https://support.google.com/assistant/answer/7394306 (accessed on 7 August 2020).
4. Talk to Bixby Using Voice Wake–Up. Available online: https://www.samsung.com/us/support/answer/ANS00080448 (accessed on 7 August 2020).
5. Yoon, S.-H.; Koh, M.-S.; Park, J.-H.; Yu, H.-J. A new replay attack against automatic speaker verification systems. *IEEE Access* **2020**, *8*, 36080–36088. [CrossRef]
6. Yoon, S.-H.; Yu, H.-J. Multiple points input for convolutional neural networks in replay attack detection. In Proceedings of the 2020 IEEE 45th International Conference on Acoustic, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 6444–6448.
7. Wu, H.; Liu, S.; Meng, H.; Lee, H. Defense against adversarial attacks on spoofing countermeasures of ASV. In Proceedings of the 2020 45th IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 6564–6568.
8. Chen, T.; Kumar, A.; Nagarsheth, P.; Sivaraman, G.; Khoury, E. Generalization of audio deepfake detection. In Proceedings of the Odyssey Speaker and Language Recognition Workshop, Tokyo, Japan, 1–5 November 2020; pp. 132–137.
9. Wang, Q.; Lee, K.A.; Koshinaka, T. Using multi-resolution feature maps with convolutional neural networks for anti-spoofing in ASV. In Proceedings of the Odyssey Speaker and Language Recognition Workshop, Tokyo, Japan, 1–5 November 2020; pp. 138–142.
10. Monteiro, J.; Alam, J.; Falk, T.H. A multi-condition training strategy for countermeasures against spoofing attacks to speaker recognizers. In Proceedings of the Odyssey Speaker and Language Recognition Workshop, Tokyo, Japan, 1–5 November 2020; pp. 296–303.
11. Yoon, S.-H.; Koh, M.-S.; Yu, H.-J. Phase spectrum of time-flipped speech signals for robust spoofing detection. In Proceedings of the Odyssey Speaker and Language Recognition Workshop, Tokyo, Japan, 1–5 November 2020; pp. 319–325.
12. Chettri, B.; Kinnunen, T.; Benetos, E. Subband modeling for spoofing detection in automatic speaker verification. In Proceedings of the Odyssey Speaker and Language Recognition Workshop, Tokyo, Japan, 1–5 November 2020; pp. 341–348.
13. Dehak, N. Discriminative and Generative Approaches for Long- and Short-Term Speaker Characteristics Modeling: Application to Speaker Verification. Ph.D. Thesis, École de Technologie Supérieure, Montreal, QC, Canada, 2009.
14. Dehak, N.; Kenny, P.; Dehak, R.; Dumouchel, P.; Ouellet, P. Front-end factor analysis for speaker verification. *IEEE Trans. AudioSpeechLang. Process.* **2010**, *19*, 788–798. [CrossRef]

15. Ioffe, S. Probabilistic linear discriminant analysis. In Proceedings of the 9th European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; pp. 531–542.
16. Yuan, M.; Lin, Y. Model selection and estimation in the Gaussian graphical model. *Biometrika* **2007**, *94*, 19–35. [CrossRef]
17. Friedman, J.; Hastie, T.; Tibshirani, R. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* **2008**, *9*, 432–441. [CrossRef] [PubMed]
18. Witten, D.M.; Friedman, J.H.; Simon, N. New insights and faster computations for the graphical lasso. *J. Comput. Graph. Stat.* **2011**, *20*, 892–900. [CrossRef]
19. Campbell, W.M.; Sturim, D.E.; Reynolds, D.A. Support vector machines using GMM supervectors for speaker verification. *IEEE Signal. Process. Lett.* **2006**, *13*, 308–311. [CrossRef]
20. Reynolds, D.A.; Quatieri, T.F.; Dunn, R.B. Speaker verification using adapted Gaussian mixture models. *Digit. Signal. Process.* **2000**, *10*, 19–41. [CrossRef]
21. Kenny, P. Bayesian speaker verification with heavy-tailed priors. In Proceedings of the Odyssey Speaker and Language Recognition Workshop, Brno, Czech Republic, 28 June–1 July 2010; p. 14.
22. Matejka, P.; Glembek, O.; Castaldo, F.; Alam, M.J.; Plchot, O.; Kenny, P.; Burget, L.; Cernocky, J. Full-covariance UBM and heavy-tailed PLDA in i-vector speaker verification. In Proceedings of the 2011 IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP), Prague, Czech Republic, 22–27 May 2011; pp. 4828–4831.
23. Garcia-Romeo, D.; Espy-Wilson, C.Y. Analysis of i-vector length normalization in speaker recognition systems. In Proceedings of the Interspeech, Florence, Italy, 27–31 August 2011; pp. 249–252.
24. Yoon, S.-H.; Koh, M.-S.; Yu, H.-J. Fuzzy restricted Boltzmann machine based probabilistic linear discriminant analysis for noise-robust text-dependent speaker verification on short utterance. *IAENG Int. J. Comput. Sci.* **2020**, *47*, 468–480.
25. Villalba, J.; Chen, N.; Snydner, D.; Garcia-Romeo, D.; McCree, A.; Sell, G.; Borgstrom, J.; Garcia-Perera, L.P.; Richardson, F.; Dehak, R.; et al. State-of-the-art speaker recognition with neural network embeddings in NIST SRE18 and speakers in the wild evaluations. *Comput. Speech Lang.* **2020**, *60*, 101026. [CrossRef]
26. Variani, E.; Lei, X.; McDermott, E.; Moreno, I.L.; Gonzalez-Dominguez, J. Deep neural networks for small footprint text-dependent speaker verification. In Proceedings of the 2014 IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 4052–4056.
27. Liu, Y.; Qian, Y.; Chen, N.; Fu, T.; Zhang, Y.; Yu, K. Deep feature for text-dependent speaker verification. *Speech Commun.* **2015**, *73*, 1–13. [CrossRef]
28. Chen, Y.H.; Lopez-Moreno, I.; Sainath, T.N.; Visontai, M.; Alvarez, R.; Parada, C. Locally-connected and convolutional neural networks for small footprint speaker recognition. In Proceedings of the Interspeech, Dresden, Germany, 6–10 September 2015; pp. 1136–1140.
29. Sak, H.; Senior, A.; Beaufays, F. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In Proceedings of the Interspeech, Singapore, 14–18 September 2014; pp. 338–342.
30. Heigold, G.; Moreno, I.; Bengio, S.; Shazzer, N. End-to-end text-dependent speaker verification. In Proceedings of the 2016 IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP), Lujiazui, China, 20–25 March 2016; pp. 5115–5119.
31. Hochreiter, S.; Schmidhuber, J. Long short term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
32. Bengio, Y.; Frasconi, P.; Simard, P. The problem of learning long-term dependencies in recurrent networks. In Proceedings of the IEEE International Conference on Neural Networks, San Francisco, CA, USA, 28 March–1 April 1993; pp. 1183–1188.
33. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
34. Chung, J.S.; Nagrani, A.; Zisserman, A. VoxCeleb2: Deep speaker recognition. In Proceedings of the Interspeech, Hyderabad, India, 2–6 September 2018; pp. 1086–1090.
35. Hajibabaei, M.; Dai, D. Unified hypersphere embedding for speaker recognition. *arXiv* **2018**, arXiv:1807.08312.
36. Bhattacharya, G.; Alam, J.; Kenny, P. Deep speaker recognition: Modular or monolithic? In Proceedings of the Interspeech, Graz, Austria, 15–19 September 2019; pp. 1143–1147.
37. Zeinali, H.; Wang, S.; Silnova, A.; Matejka, P.; Plchot, O. BUT system description to VoxCeleb speaker recognition challenge. *arXiv* **2019**, arXiv:1910.12592.

38. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 7132–7141.
39. Snyder, D.; Garcia-Romeo, D.; Sell, G.; Povey, D.; Khudanpur, S. Deep neural network embeddings for text-independent speaker verification. In Proceedings of the Interspeech, Stockholm, Sweden, 20–24 August 2017; pp. 999–1003.
40. Snyder, D.; Garcia-Romeo, D.; Sell, G.; Povey, D.; Khudanpur, S. X-vectors: Robust DNN embeddings for speaker recognition. In Proceedings of the 2018 IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 5329–5333.
41. Waibel, A.; Hanazawa, T.; Hinton, G.; Shikano, K.; Lang, K.J. Phoneme recognition using time-delay neural networks. *IEEE Trans. Acoust. SpeechSignal Process.* **1989**, *37*, 328–339. [CrossRef]
42. Prince, S.J.D.; Elder, J.H. Probabilistic linear discriminant analysis for inferences about identity. In Proceedings of the 2007 IEEE 11th International Conference on Computer Vision, Rio De Janeiro, Brazil, 14–21 October 2007; pp. 1–8.
43. Brummer, N.; de Villiers, E. The speaker partitioning problem. In Proceedings of the Odyssey Speaker and Language Recognition Workshop, Brno, Czech Republic, 28 June–1 July 2010; p. 34.
44. Povey, D.; Ghoshal, A.; Boulianne, G.; Burget, L.; Glembek, O.; Goel, N.; Hannemann, M.; Motlicek, P.; Qian, Y.; Schwarz, P.; et al. The Kaldi speech recognition toolkit. In Proceedings of the IEEE 2011 Workshop on Automatic Speech Recognition and Understanding, Big Island, HI, USA, 11–15 December 2011.
45. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum likelihood from incomplete data via EM algorithm. *J. R. Stat. Soc. Ser. B Methodol.* **1977**, *39*, 1–22.
46. Ding, K. A note on Kaldi's PLDA implementation. *arXiv* **2018**, arXiv:1804.00403v1.
47. Rue, H.; Held, L. *Gaussian Markov Random Fields: Theory and Applications*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2005.
48. Murphy, K.P. Undirected graphical models (Markov random fields). In *Machine Learning: A Probabilistic Perspective*; MIT Press: Cambridge, MA, USA, 2012; pp. 661–705.
49. Meinshausen, N.; Buhlmann, P. High-dimensional graphs and variable selection with the lasso. *Ann. Stat.* **2006**, *34*, 1436–1462. [CrossRef]
50. Zhao, P.; Yu, B. On model selection consistency of lasso. *J. Mach. Learn. Res.* **2006**, *7*, 2541–2563.
51. Banerjee, O.; Ghaoui, L.E.; d'Aspremont, A. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *J. Mach. Learn. Res.* **2008**, *9*, 485–516.
52. Mazumder, R.; Hastie, T. Exact covariance thresholding into connected components for large-scale graphical lasso. *J. Mach. Learn. Res.* **2012**, *13*, 781–794. [PubMed]
53. Spall, J.C. Cyclic seesaw process for optimization and identification. *J. Optim. Theory Appl.* **2012**, *154*, 187–208. [CrossRef]
54. Li, L.; Tang, Z.; Shi, Y.; Wang, D. Gaussian-constrained training for speaker verification. In Proceedings of the 2019 IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP), Brigton, UK, 12–17 May 2019; pp. 6036–6040.
55. Zhang, Y.; Li, L.; Wang, D. VAE-based regularization for deep speaker embedding. In Proceedings of the Interspeech, Graz, Austria, 15–19 September 2019; pp. 4020–4024.
56. Gu, B.; Guo, W. Gaussian speaker embedding learning for text-independent speaker verification. *arXiv* **2020**, arXiv:2001.04585.
57. Larcher, A.; Lee, K.A.; Ma, B.; Li, H. Text-dependent speaker verification: Classifiers, databases and RSR2015. *Speech Commun.* **2014**, *60*, 56–77. [CrossRef]
58. Yoon, S.-H.; Yu, H.-J. A simple distortion-free method to handle variable length sequences for recurrent neural networks in text dependent speaker verification. *Appl. Sci.* **2020**, *10*, 4092. [CrossRef]
59. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 13–15 May 2017; pp. 2616–2620.
60. Reddi, S.J.; Kale, S.; Kumar, S. On the convergence of Adam and beyond. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
61. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations, Banff, AL, Canada, 14–16 April 2014.

62. Lin, Z.; Feng, M.; Santos, C.S.; Yu, M.; Xiang, B.; Zhou, B.; Bengio, Y. A structured self-attentive sentence embedding. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
63. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 7–9 July 2015.
64. Jozefowicz, R.; Zaremba, W.; Sutskever, I. An empirical exploration of recurrent neural network architectures. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 7–9 July 2015.
65. Duchi, J.; Gould, S.; Koller, D. Projected subgradient methods for learning sparse Gaussian. In Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence (UAI), Helsinki, Finland, 9–12 July 2008; pp. 153–160.
66. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic differentiation in PyTorch. In Proceedings of the Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
67. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
68. Nagrani, A.; Chung, J.S.; Zisserman, A. VoxCeleb: A large-scale speaker identification dataset. In Proceedings of the Interspeech, Stockholm, Sweden, 20–24 August 2017; pp. 2616–2620.
69. Golub, G.H.; Van Loan, C.F. *Matrix Computations*; Johns Hopkins University Press: Baltimore, MD, USA; London, UK, 1996.