Fast Scoring for Mixture of PLDA in I-Vector/PLDA Speaker Verification

Man-Wai Mak
Center for Signal Processing, Dept. of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hong Kong SAR
E-mail: enmwmak@polyu.edu.hk

Abstract—With the ubiquitous of mobile phones, users of speaker verification systems will perform authentication anywhere at anytime. As a result, practical speaker verification systems need to deal with utterances of different noise levels. Recently, an SNR-dependent mixture of PLDA model was proposed to deal with such practical situation. However, the scoring function of this model is significantly more complex than the conventional one. This paper proposes a method to reduce the computation burden of this mixture PLDA model. The idea is based on the observation that for most utterances, the posterior probabilities of SNR are very sparse so that it is possible to consider the top Gaussian only during scoring. The method effectively reduces the computational complexity from $O(K^2D^3)$ to $O(D^3)$, where $K$ and $D$ are the number of mixtures and i-vector dimension, respectively. Experimental results based on NIST 2012 SRE suggest that the proposed method can reduce computation time by 60% with very minor degradation in performance.

I. INTRODUCTION

In the past decades, a number of methods have been proposed to reduce the effect of background noise in speaker verification systems. Some of these methods address the problem in the front-end or during feature extraction stage, e.g., [1]–[4]. There were also attempts to use speech enhancement techniques to reduce the effect of noise [5]. While the effectiveness of these feature-based approaches has been demonstrated, recent researches have found that techniques that operate on the backend classification stage are more promising. Among them, the joint factor analysis (JFA) [6] and i-vector/PLDA framework [7], [8] have been by far the most successful.

The i-vector/PLDA framework comprises two stages of factor analyses and dimension reduction. In the first stage, the acoustic characteristics of an utterance is represented by a low-dimensional vector called the i-vector that lives in the subspace of the GMM-supervector space [9] that is formed by stacking the mean vectors of a universal background model (UBM) [10]. In other words, the acoustic variabilities of utterances are modelled by a factor analyzer in which the latent space is of much lower dimension than the GMM-supervector space. Given an utterance, its spectral features (typically MFCC) are aligned with the UBM in a frame-by-frame basis. Then, the posterior probabilities (also known as responsibility) of individual mixtures in the UBM are estimated to compute the zero- and first-order sufficient statistics. Based on the sufficient statistics, the posterior density of the latent factors of the factor analysis model is computed and the posterior mean is considered as the i-vector. The space in which the i-vector lives is called the total variability space [7], [11].

Because the total variability space accounts for both speaker and other variabilities – such as channel, reverberation, and noise – a second stage of dimension reduction and normalization is required to suppress the channel effects. State-of-the-art speaker verification systems typically use a supervised factor analyzer called probabilistic linear discriminant analysis (PLDA) [12] to further suppress these variabilities. Some systems [7], [13], [14] also apply linear discriminant analysis (LDA) [15] and within-class covariance normalization (WCCN) [16] to reduce the dimension of i-vectors and to normalise their covariance before applying PLDA. It has been found that vector-length normalization is a simple but effective way to make the i-vectors more amenable to Gaussian PLDA modelling [17].

Recent methods to address noise robustness in speaker verification systems are typically built on top of the i-vector/PLDA framework. For example, in [18]–[21], multi-condition training was applied. Clean and noisy utterances are pooled together to train a PLDA model so that it becomes more robust to noisy test utterances. In [22], multiple PLDA models are trained, one for each condition. Hasan and Hansen [23] performed mixture of probabilistic PCA on feature space so that the posterior means of the mixture-dependent acoustic factors can be incorporated into an i-vector extractor. This idea has been further enhanced by replacing the UBM by a mixture of acoustic factor analyzers for i-vector extraction [24]. Recently, Lei et al. [25] proposed adapting a clean UBM to noisy utterances using vector Taylor series. I-vectors are then extracted based on the noise-adapted UBM. The idea is to clean up the i-vectors so that they become independent of additive and convolutive noise. Li and Mak [26] proposed an SNR-invariant PLDA model by introducing an SNR factor and an SNR subspace to the conventional PLDA model. The results show that the SNR factor is very effective in suppressing the variability in i-vectors caused by SNR variations in the utterances.

II. MIXTURE OF SNR-DEPENDENT PLDA

In most practical situations, a speaker verification system needs to deal with utterances of different noise levels because users may use the system in different acoustic environments,
e.g., offices, streets, restaurants, and subway stations, etc. As a result, the utterances received by the system may have a wide range of SNR. To tackle the varying noise levels, Mak [27] argued that the SNR of utterances should be divided into a number of regions so that the utterances in each region can be modelled more accurately by an SNR-dependent PLDA model. Based on this idea, Mak [27] proposed a mixture model called SNR-dependent mixture of PLDA or mPLDA in short.

A. Model Parameters

In mPLDA, i-vectors are modelled by a mixture of SNR-dependent factor analyzers with parameters

$$\theta = \{ \Delta, \omega \} = \{ \lambda_k, \omega_k \}$$

$$= \{ \pi_k, \mu_k, \sigma_k, m_k, V_k, \Sigma_k \}$$

$$= \{ \pi_k, \mu_k, \sigma_k \}$$

where $\lambda_k = \{ \pi_k, \mu_k, \sigma_k \}$ contains the prior probability, mean and standard deviation of the SNR in the $k$-th group, and $\omega_k = \{ m_k, V_k, \Sigma_k \}$ comprises the mean i-vector, factor loading matrix, and residual covariance of the $k$-th factor analyzer corresponding to the $k$-th SNR group. The EM formulations for estimating $\theta$ in Eq. 1 can be found in [27].

B. Mixture Alignments

Denote $y_k$’s as the indicator variables specifying which of the factor analyzers is responsible for generating the i-vector $x$, and denote $\ell$ as the SNR of the corresponding utterance. Then, the posterior probability of $y_k$ is

$$\gamma_\ell(y_k) \equiv P(y_k = 1 | \ell, \Lambda) = \frac{\pi_k N((\ell | \mu_k, \sigma_k^2)}{\sum_{k=1}^{K} \pi_k N((\ell | \mu_k, \sigma_k^2))}. \tag{2}$$

Eq. 2 implies that the alignments of i-vectors in the mixture model are purely based on the posterior probabilities of SNR. This property gives the SNR-dependent mixture of PLDA an advantage over the conventional mixture of factor analysis [28] in that the i-vector clusters obtained by the EM algorithm is more prominent.

C. Likelihood Ratio Scores

Given target-speaker’s i-vector $x_s$ and test i-vector $x_t$ and the SNR $\ell_s$ and $\ell_t$ (in dB) of the corresponding utterances, the same-speaker marginal likelihood is

$$p(x_s, x_t, \ell_s, \ell_t | \text{same-speaker})$$

$$= p(\ell_s)p(\ell_t)p(x_s, x_t | \ell_s, \ell_t, \text{same-speaker})$$

$$= p_{st} \sum_{k_s=1}^{K} \sum_{k_t=1}^{K} \int p(x_s, x_t, y_{k_s}, y_{k_t} = 1, y_{k_t} = 1, z | \theta_s, \ell_s, \ell_t)dz$$

$$= p_{st} \sum_{k_s=1}^{K} \sum_{k_t=1}^{K} \int p(x_s, x_t | y_{k_s}, y_{k_t} = 1, y_{k_s} = 1, z | \theta_s, \ell_s, \ell_t)dz$$

$$\times p(y_{k_s}, y_{k_t} | \ell_s, \ell_t)$$

$$\times p(\ell_s)p(\ell_t)$$

$$\times \mathcal{N} \left( \begin{pmatrix} x_s^T & x_t^T \end{pmatrix}^T \left[ \begin{pmatrix} m_{k_s}^T & m_{k_t}^T \end{pmatrix}^T, \Sigma_{k_s} + \Sigma_{k_t} \right] \right)$$

$$\hat{V}_{k_s, k_t} = [V_{k_s}^T, V_{k_t}^T]^T, \Sigma_k = \text{diag}\{ \Sigma_{k_s}, \Sigma_{k_t} \}$$

where $p_{st} = p(\ell_s)p(\ell_t)$, $\hat{V}_{k_s, k_t} = [V_{k_s}^T, V_{k_t}^T]^T$, $\Sigma_k = \text{diag}\{ \Sigma_{k_s}, \Sigma_{k_t} \}$ and

$$\gamma_{\ell_s, \ell_t}(y_{k_s}, y_{k_t}) \equiv P(y_{k_s} = 1, y_{k_t} = 1 | \ell_s, \ell_t, \Lambda)$$

$$= \frac{\pi_{k_s} \pi_{k_t} N((\ell_s | \mu_{k_s}, \sigma_{k_s}^2) \text{diag}\{ \sigma_{k_s}^2, \sigma_{k_t}^2 \})}{\sum_{k_s=1}^{K} \sum_{k_t=1}^{K} \pi_{k_s} \pi_{k_t} N((\ell_s | \mu_{k_s}, \mu_{k_t}) \text{diag}\{ \sigma_{k_s}^2, \sigma_{k_t}^2 \})} \tag{3}$$

Similarly, the different-speaker marginal likelihood is

$$p(x_s, x_t, \ell_s, \ell_t | \text{diff-speaker}) = p(x_s, x_t | \text{Spk } s)p(\ell_s, \ell_t | \text{Spk } t),$$

where

$$p(\ell_s, \ell_t | \text{Spk } s) = p(\ell_s) \sum_{k_s=1}^{K} \int p(x_s, y_{k_s} = 1, z | \theta_s, \ell_s)dz$$

$$= p(\ell_s) \sum_{k_s=1}^{K} \gamma_{\ell_s}(y_{k_s}) N \left( x_s | m_{k_s}, V_{k_s} \Sigma_{k_s} + m_{k_s} ^T \right),$$

and similarly for $p(x_s, x_t | \text{Spk } t)$. Therefore, the likelihood ratio $S_{\text{mPLDA}}$ is given by Eq. 4 at the bottom of next page.

Note that Eq. 4 is likely to cause numerical problems if they are evaluated directly because the determinant of covariance matrices $\Sigma_k$ could exceed the double-precision representation. This problem, however, can be avoided by computing the logarithm of determinant and noting the identity: $|aA| = a^D |A|$, where $a$ is a scalar and $A$ is a $D \times D$ matrix. Thus, we can rewrite Eq. 4 as Eq. 5 shown at the bottom of next page, where $\Delta_{k_s, k_t} = V_{k_s} \Sigma_{k_s} + \Sigma_{k_t}, \Lambda_{k_s} = V_{k_s} \Sigma_{k_s}, \Sigma_{k_s, k_t} = \text{diag}\{ \Sigma_{k_s}, \Sigma_{k_t} \}$, and

$$D(a|b) = (a - b)^T S_a^{-1} (a - b), \tag{7}$$

where $S = \text{cov}(a, a)$. In this work, $\alpha = 5$. Note that because Eq. 5 is derived from Bayes’ rule, mPLDA does not require score normalization.

D. Complexity Analysis

Note that the determinants in Eq. 5 can be pre-computed, so as the covariance matrices $\Delta_{k_s, k_t}$ and $\Lambda_{k_s}$. As a result, the major computation burden in the scoring function lies in the computation of Mahalanobis distance $D(\cdot|\cdot)$. More precisely, the computational complexity of the numerator and denominator of Eq. 5 are $O(K^2 (2D)^3)$ and $O(K^2 D^3)$, respectively, where $D$ is the dimensionality of i-vectors (after LDA+WCCN in our case) and $K$ is the number of mixtures. Therefore, the overall complexity is $O(K^2 D^3)$. Table I summarises the computational complexity of the three scoring methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Computational Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLDA</td>
<td>$O(D^3)$</td>
</tr>
<tr>
<td>mPLDA</td>
<td>$O(K^2 D^3)$</td>
</tr>
<tr>
<td>Fast mPLDA</td>
<td>$O(D^3)$</td>
</tr>
</tbody>
</table>

Table I: Computational complexity of the likelihood-ratio scoring function in PLDA, mPLDA, and Fast mPLDA.
III. Fast Scoring for mPLDA

When comparing with the computational complexity of the original PLDA ($O(D^3)$), the computational complexity of mPLDA is $K^2$ times as much. Our results suggest that this will be a burden of mPLDA when K is larger than 2. It is therefore, imperative to find a method to reduce the scoring complexity of mPLDA.

A. Sparseness Analysis of SNR Posteriors

In Eq. 5, it is necessary to evaluate the likelihood for each combination of $k_s$ and $k_t$. This is the major computation burden in mPLDA, especially when the number of mixture is large. If the posterior probabilities of SNR $[\gamma_{\ell_s, \ell_t}(y_s, y_t)]$ are sparse, we may drop the combinations of $(k_s, k_t)$ that lead to small posterior $\gamma_{\ell_s, \ell_t}(y_s, y_t)$ when computing the likelihood.

Fig. 1(a) shows the average SNR posteriors for $K = 3$, sorted in descending order of $\gamma_{\ell_s, \ell_t}(y_s, y_t)$. There are totally 9 combinations of $(k_s, k_t)$ when $K = 3$. Evidently, the maximum average posterior dominates among the 9 combinations and is significantly larger than the first runner-up. Fig. 1(b) shows the individual posterior probabilities of SNR for 150 combinations of target-speaker utterances’ SNR $(\ell_s)$ and test-utterances’ SNR $(\ell_t)$. The figure further confirms the dominance of the winner among the 9 combinations.

B. Scoring Function

In the extreme case, we may only keep the combination that leads to maximum posterior. Based on this idea, Eq. 4 reduces to Eq. 6 at the bottom of this page, where

$$(k_s, k_t) = \arg \max_{(k_s, k_t)} \gamma_{\ell_s, \ell_t}(y_s, y_t).$$

A comparison between Eq. 6 and Eq. 5 reveals that the complexity has been reduced by $K^2$ times.

Eq. 6 can be written as

$$\log S_{\text{mPLDA}}(x_s, x_t) = \log \left[ \frac{\gamma_{k_s, k_t} \mathcal{N}\left( x_s | x_t, m_{k_s}, m_{k_t}, \Phi_{k_s}, \Psi_{k_s}, \Phi_{k_t}, \Psi_{k_t} \right)}{\gamma_{k_s, k_t} \mathcal{N}\left( x_s | x_t, m_{k_s}, m_{k_t}, \Phi_{k_s}, \Psi_{k_s}, 0, \Phi_{k_t} \right)} \right].$$

where

$$\Phi_{k_s} = V_{k_s} V_{k_s}^T + \Sigma_{k_s},$$

$$\Psi_{k_t} = V_{k_t} V_{k_t}^T,$$

$$\Phi_{k_t} = V_{k_t} V_{k_t}^T + \Sigma_{k_t}.$$
Expanding Eq. 10, we have
\[
\log S_{\text{fast-mPLDA}}(s_t, x_t) = \log \gamma_{k_t,k_t} - \log \gamma_{k_t} - \log x_k - \frac{1}{2} x_k^T Q_{k_t,k_t} x_k + \frac{1}{2} m_{k_t}^T P_{k_t,k_t} m_{k_t} \\
+ \frac{1}{2} x_k^T Q_{k_t,k_t} (s_t + 2m_{k_t}) + \frac{1}{2} m_{k_t}^T P_{k_t,k_t} (s_t + 2m_{k_t}) \\
+ x_k^T P_{k_t,k_t} (x_t + m_{k_t}) + x_t^T P_{k_t,k_t} m_{k_t} \\
- \frac{1}{2} \log |D_1| + \frac{1}{2} \log |D_2| \\
- \frac{1}{2} m_{k_t}^T Q_{k_t,k_t} m_{k_t} - \frac{1}{2} m_{k_t}^T Q_{k_t,k_t} m_{k_t} \\
- \frac{1}{2} m_{k_t}^T P_{k_t,k_t} m_{k_t} - \frac{1}{2} m_{k_t}^T P_{k_t,k_t} m_{k_t}.
\]
Note that the last six terms in Eq. VI are independent of \(x_s\) and \(x_t\). Therefore, they can be pre-computed before scoring. In the sequel, we refer to the scoring function in Eq. VI as fast mPLDA scoring.

Note also that the conventional PLDA scoring is a special case of Eq. VI where \(k_t = k_t = K = 1\) and \(m_{k_t} = m_{k_t} = 0\), i.e., there is only one mixture and therefore the SNR posterior \(\gamma\) always equal to 1.0. Specifically, the scoring function for PLDA is
\[
\log S_{\text{PLDA}}(s_t, x_t) = \frac{1}{2} x_s^T Q x_s + \frac{1}{2} m_t^T P x_t + \log \sum_{k=1}^{K} e^{-\frac{1}{2} x_k^T Q x_k + m_k^T P x_k}.
\]
where we have dropped the subscripts for \(P\) and \(Q\) for clarity. Similar to fast mPLDA, \(P\) and \(Q\) can be pre-computed using Eq. 11.

IV. EXPERIMENTS
A. Speech Data and Acoustic Features
The male speech files in the core set of NIST 2012 Speaker Recognition Evaluation (SRE) [30] were used for performance evaluation. In 2012 SRE, noise was added to the test segments of common conditions 3 and 4, resulting in the SNR distribution shown in Fig. 2. Because mPLDA is designed to improve the robustness of PLDA systems under noisy environments, this paper focuses on these two common conditions. Table II shows the acoustic conditions of the test segments in these common conditions. Enrollment utterances with length less than 10 seconds and the summed-channel utterances were removed. However, we ensured that all target speakers have at least one utterance for enrollment. The speech files in NIST 2005–2010 SREs were used as development data for training gender-dependent UBMs, total variability matrices, LDA-WCCN [7], PLDA and mPLDA models.

Speech regions in the speech files were extracted by using a two-channel VAD [31]. 19 MFCCs together with energy plus their 1st- and 2nd- derivatives were extracted from the speech regions, followed by cepstral mean normalization and feature warping [4] with a window size of 3 seconds. A 60-dim acoustic vector was extracted every 10ms, using a Hamming window of 25ms. For each clean training file, we randomly select one out of the 30 noise files from the PRISM dataset [32] and added the noise waveform to the file at an SNR of 6dB and 15dB using the FaNT tool [33].

<table>
<thead>
<tr>
<th>TABLE II</th>
<th>CONDITIONS OF TEST SEGMENTS IN CC3 AND CC4 OF NIST 2012 SRE.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Condition</td>
<td>Test-segment Conditions</td>
</tr>
<tr>
<td>CC3</td>
<td>Interview speech with added noise</td>
</tr>
<tr>
<td>CC4</td>
<td>Phone call speech with added noise</td>
</tr>
</tbody>
</table>

B. SNR Measurements
To measure the “actual” SNR of speech files (including the original and noise contaminated ones), we used the voltmeter function of FaNT and the speech/non-speech decisions of our VAD [31], [34] as follows. Given a speech file, we passed the waveform to the G.712 frequency weighting filter in FaNT and
then estimated the speech energy using the voltmeter function 
(sv-p56.c from the ITU-T Software Tool Library [35]).
Then, we extracted the non-speech segments based on the
VAD’s decisions and passed the non-speech segments to
the voltmeter function to estimate the noise energy.
The difference between the signal and noise energies in the log domain gives
the measured SNR of the file.

\[ \gamma \]

C. PLDA and Mixture of PLDA

The i-vector system is based on gender-dependent UBMs
with 1024 mixtures and total variability matrices with 500
total factors. Microphone and telephone utterances from NIST
2005–2008 SREs were used for training the UBMs and total
variability matrices. Following [14], within-class covariance
normalization (WCCN) [16] and i-vector length normalization
[17] were applied to the 500-dimensional i-vectors. Then,
linear discriminant analysis (LDA) [15] and WCCN were
applied to reduce the dimension to 200 before training the
PLDA and mixture of PLDA models with 150 latent variables.

To train the mPLDA models for CC4, we pooled the
6dB (tel), 15dB (tel), and original (tel+mic) speech files in
2006–2010 SRE—excluding speakers with less than two utterances—into a single training set.
The EM algorithm specified in [27] were used to train a mixture PLDA models
with \( K = 3 \) and \( K = 4 \). The number of speaker factors
(\( M \)) was set to 150 in all cases. As shown in Table II,
the test segments in CC3 comprises interview speech with
added noise. As a result, we only used microphone enrollment
utterances to train the PLDA and mPLDA models for CC3.

As for PLDA scoring, we followed the conventional PLDA
scoring function [17] as specified in Eq. 14. For mPLDA
scoring and fast mPLDA scoring, we applied Eq. 5 and Eq. VI,
respectively.

V. RESULTS AND DISCUSSIONS

Tables III(a) and III(b) show the breakdown of computation
time within each invocation of the Matlab functions that
implement mPLDA scoring (Eq. 5) and fast mPLDA scoring
(Eq. VI). The computation time was measured by Matlab’s
Profiler when \( K \) was set to 3, average over all the trials in CC4.
When estimating the computation time for Eq. 5 and Eq. VI,
it is assumed that the posterior probabilities \( \gamma \)’s have already
been computed. I-vector preprocessing includes the time to
perform i-vector whitening, length-normalization, LDA, and
WCCN projection. All measurements were done on an Intel
Quad CPU Q9550 running at 2.83GHz.

For mPLDA, the computation of likelihoods in Eq. 5 takes
over 60% of the overall time. This is because when \( K = 3 \),
there are 9 Mahalanobis distances in the numerator and 6 Ma-
halanobis distances in the denominator. For fast mPLDA,
on the other hand, the computation of SNR posteriors consumes
43% of the overall time. This suggests that the scoring function
in Eq. VI is very efficient. The computation saving comes
from omitting the likelihoods with small SNR posterior \( \gamma \).
Comparing the overall time in Tables III(a) and III(b) reveals
that fast mPLDA reduces the scoring time by more than 60%.

Table IV shows the EER and minimum DCF (minDCF)
achieved by PLDA (baseline), mPLDA, and fast mPLDA in
CC3 and CC4 of NIST 2012 SRE. Also shown are the scoring
time (in seconds) to perform all trials using different methods.
Results show that the EER of mPLDA is significantly lower
than that of PLDA, although the former is slightly inferior
in terms of minDCF. While the computational complexity
of PLDA and fast mPLDA is the same (see Table I), the actual
scoring time of fast mPLDA is still significantly longer than
that of PLDA. The reason is that computing the SNR posterior
probabilities takes time, as shown in Table III. Table IV also
demonstrates that the fast scoring approach proposed in this
paper reduces the overall scoring time by more than 60%.

VI. CONCLUSIONS

This paper proposes to speed up the scoring process of SNR-
dependent mixture of PLDA. This is achieved by omitting the
computation of the likelihood terms when their corresponding
SNR posterior is small. In the extreme case, only the likelihood
whose SNR posterior is the largest is considered. It was found
that the SNR posteriors are sparse so that even for this extreme
case, the loss in performance is minor but the scoring time can
be cut by half. In future work, it is interesting to consider not
only the top SNR posterior but also a few runner-ups when
evaluating the scoring function to see if it is possible to reduce
scoring time without sacrificing verification performance.

ACKNOWLEDGMENT

This work was in part supported by The RGC of Hong Kong
SAR, Grant No. PolyU 152117/14E.
**TABLE IV**

<table>
<thead>
<tr>
<th>Method</th>
<th>$K$</th>
<th>CC3</th>
<th>Time (sec.)</th>
<th>CC4</th>
<th>Time (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLDA</td>
<td></td>
<td>EER(%)</td>
<td>Time (sec.)</td>
<td>EER(%)</td>
<td>Time (sec.)</td>
</tr>
<tr>
<td></td>
<td>0.255</td>
<td>5.6%</td>
<td>0.372</td>
<td>4.79</td>
<td>6098</td>
</tr>
<tr>
<td>mPLDA</td>
<td>2</td>
<td>0.216</td>
<td>0.238</td>
<td>2.295</td>
<td>331</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.206</td>
<td>0.238</td>
<td>2.39</td>
<td>3529</td>
</tr>
<tr>
<td>Fast mPLDA</td>
<td>4</td>
<td>0.238</td>
<td>0.206</td>
<td>3.10</td>
<td>2286</td>
</tr>
</tbody>
</table>

**TABLE III**

I-vector preprocessing time and computation time in different parts of (a) mPLDA in Eq. 5 and (b) Fast mPLDA in Eq. VI. The I-vector preprocessing time includes whitening, length-normalization, LDA, and WCCN. The computation time was obtained by using MATLAB. In both cases, the time required for computing $S_{mPLDA}(x_i, x_i)$ and $S_{Fast-mPLDA}(x_i, x_i)$ assumes that $\gamma_{x, t}(y_{k, y_1})$, $\gamma_{y_1}(y_{k, y_1})$ and $\gamma_{t, y_1}(y_{k, y_1})$ have been computed.

<table>
<thead>
<tr>
<th>Function</th>
<th>Time (ms)</th>
<th>% of Scoring Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_{mPLDA}(x_i, x_i)$ in Eq. 5</td>
<td>2.295</td>
<td>61.3%</td>
</tr>
<tr>
<td>I-vector Preprocessing</td>
<td>0.678</td>
<td>18.1%</td>
</tr>
<tr>
<td>$\gamma_{x, t}(y_{k, y_1})$</td>
<td>0.343</td>
<td>9.2%</td>
</tr>
<tr>
<td>$\gamma_{y_1}(y_{k, y_1})$</td>
<td>0.216</td>
<td>5.8%</td>
</tr>
<tr>
<td>Other operations and overhead</td>
<td>0.209</td>
<td>5.6%</td>
</tr>
<tr>
<td>Overall</td>
<td>3.741</td>
<td>100%</td>
</tr>
</tbody>
</table>

(a) mPLDA

<table>
<thead>
<tr>
<th>Function</th>
<th>Time (ms)</th>
<th>% of Scoring Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-vector Preprocessing</td>
<td>0.520</td>
<td>37.0%</td>
</tr>
<tr>
<td>$\gamma_{x, t}(y_{k, y_1})$</td>
<td>0.372</td>
<td>26.5%</td>
</tr>
<tr>
<td>$S_{Fast-mPLDA}(x_i, x_i)$ in Eq. VI</td>
<td>0.206</td>
<td>20.6%</td>
</tr>
<tr>
<td>$\gamma_{x, t}(y_{k, y_1})$ &amp; $\gamma_{y_1}(y_{k, y_1})$</td>
<td>0.203</td>
<td>14.4%</td>
</tr>
<tr>
<td>Other operations and overhead</td>
<td>0.106</td>
<td>7.5%</td>
</tr>
<tr>
<td>Overall</td>
<td>1.407</td>
<td>100%</td>
</tr>
</tbody>
</table>

(b) Fast mPLDA

**REFERENCES**


