Non-Iterative Parameter Estimation for Total Variability Model Using Randomized Singular Value Decomposition

Ruchir Travadi and Shrikanth Narayanan

University of Southern California

travadi@usc.edu, shri@sipi.usc.edu

Abstract

In this paper, we address the problem of parameter estimation for the Total Variability Model (TVM) [1]. Typically, the estimation of the Total Variability Matrix requires several iterations of the Expectation Maximization (EM) algorithm [2], and can be considerably demanding computationally. As a result, fast and efficient parameter estimation remains a key challenge facing the model. We show that it is possible to reduce the Maximum Likelihood parameter estimation problem for TVM into a Singular Value Decomposition (SVD) problem by making some suitably justified approximations in the likelihood function. By using randomized algorithms for efficient computation of the SVD, it becomes possible to accelerate the parameter estimation task remarkably. In addition, we show that this method is able to increase the efficiency of the ivector extraction procedure, and also lends some interpretability to the extracted ivectors.

Index Terms: Total Variability Model, ivector

1. Introduction

Total Variability Model [1] has been widely used in the domain of audio signal processing as a framework for obtaining a fixed-dimensional representation for variable length sequences. It has been successfully applied in a variety of applications including speaker recognition [1], language identification [3, 4], acoustic model adaptation for speech recognition [5, 6], and also for inferring paralinguistic information such as cognitive load [7].

Despite its popularity, training a TVM is computationally intensive since it requires several iterations of EM algorithm. Several ideas have been proposed in order to improve the efficiency of EM training and ivector extraction, such as pre-normalization of statistics [8], constant component alignment and orthogonalization [9], factorized subspace estimation [10] and Variational Bayes algorithms [11]. However, these methods are still based on EM and require several iterations.

In this paper, we propose a new algorithm based on SVD to overcome this limitation. We construct an approximation to the model likelihood function, and show that it can be maximized by computing an SVD for a single matrix. The approximations we make are reasonable when the training data is long in duration. Although the dimensionality of this matrix is very large, randomized algorithms can be used in order to obtain the SVD very efficiently [12]. This eliminates the need for iteration, and enables us to reduce the complexity of the training procedure by an order of magnitude.

In addition, SVD provides a ranking of different subspace dimensions in the form of their corresponding singular values, which leads to further advantages. First, it eliminates the need for repeated training when experimentation is necessary over different values of subspace dimensionality k: the solution can be obtained by obtaining the SVD for largest k and then truncating it as necessary. Moreover, the first few subspace dimensions correspond to the most important factors of variation in the data. Exploring correlations between them and physically observable quantities lends some degree of interpretability to the extracted ivector dimensions.

The remainder of this paper is organized as follows: we propose an algorithm for approximate likelihood maximization based on SVD in section 2, followed by a discussion of its advantages in section 3. Experimental results on RATS Language Identification (LID) corpus have been reported in section 4 followed by conclusion.

2. Parameter Estimation for TVM

Let $\mathbf{X} = \{\mathbf{X}_u\}_{u=1}^{U}$ be the collection of acoustic feature vectors in a dataset comprising of U utterances, where $\mathbf{X}_u = \{\boldsymbol{x}_{ut}\}_{t=1}^{T_u}$ denotes the feature vector sequence from a specific utterance u. Let D be the dimensionality of each feature vector: $\boldsymbol{x}_{ut} \in \mathbb{R}^D$.

In the Total Variability Model (TVM), it is assumed that with every utterance u, there is an associated vector $\boldsymbol{w}_u \in \mathbb{R}^K$, known as the *ivector* for that utterance, such that the conditional distribution of $\boldsymbol{x}_{ut}$ given $\boldsymbol{w}_u$ is a Gaussian Mixture Model (GMM) with parameters $\{p_c, \boldsymbol{\mu}_{uc} = \boldsymbol{\mu}_c + \mathbf{T}_c\boldsymbol{w}_u, \boldsymbol{\Sigma}_c\}_{c=1}^{C}$ where $p_c \in \mathbb{R}, \boldsymbol{\mu}_c \in \mathbb{R}^D, \mathbf{T}_c \in \mathbb{R}^{D\times K}$ and $\boldsymbol{\Sigma}_\mathbf{c} \in \mathbb{R}^{D\times D}$. The prior distribution for $\boldsymbol{w}_u$ is assumed to be standard normal:

$$f(\boldsymbol{w}_u) = \mathcal{N}(\mathbf{0}, \mathbf{I})$$

Let $\boldsymbol{M}_0, \boldsymbol{M}_u \in \mathbb{R}^{CD}$ denote vectors consisting of stacked global and utterance-specific component means $\boldsymbol{\mu}_c$ and $\boldsymbol{\mu}_{uc}$ respectively. Then, TVM can be summarized as:

$$\boldsymbol{M}_u = \boldsymbol{M}_0 + \mathbf{T}\boldsymbol{w}_u$$

where $\mathbf{T} \in \mathbb{R}^{CD\times K}$ is given as: $\mathbf{T} = \left[\ \mathbf{T}_1^\mathsf{T} \vdots \ldots \vdots \mathbf{T}_C^\mathsf{T}\ \right]^\mathsf{T}$ The parameters p_c and $\boldsymbol{M}_0$ are typically obtained by training a GMM, also known as a Universal Background Model (UBM), on all available speech data. Let $\Theta = \{\mathbf{T}, \boldsymbol{\Sigma}\}$ denote the remaining parameters. Our focus is on efficient estimation of Θ.

2.1. Expectation Maximization

The problem of estimating the matrices $\mathbf{T}$ and $\boldsymbol{\Sigma}$ is usually set up as a Maximum Likelihood problem:

$$\Theta^* = \text{argmax}_\Theta \log P(\mathbf{X}|\Theta)$$

An estimate Θ^* that achieves a local maximum of the likelihood function is then obtained using the Expectation Maximization (EM) algorithm by treating the ivectors $\boldsymbol{w}_u$ as hidden variables.

2.2. Related work

There has been recent work on replacing EM algorithm with PCA in [13], which is similar to our approach, but there are a few key differences. While their approach is motivated indirectly by drawing parallels between linear Gaussian model view of PCA and EM updates for TVM, we motivate ours directly from likelihood maximization for TVM. Moreover, the projection matrix in their case is computed by eigendecomposition of the covariance matrix, as opposed our approach of using randomized SVD, which is remarkably faster [12].

2.3. Proposed Algorithm: Singular Value Decomposition

To avoid an iterative procedure, we marginalize the likelihood over ivectors $\boldsymbol{w}_u$, rather than treating them as hidden variables.

2.3.1. Likelihood Function

Just like in the derivation of the EM updates [2], we start by assuming that the component associations $\mathbf{C}$ are observed, where

$$\mathbf{C} = \{\mathbf{C}_u\}_{u=1}^{U}, \mathbf{C}_u = \{c_{ut}\}_{t=1}^{T_u}, c_{ut} \in \{1, \ldots, C\}$$

Hence, the problem we consider is that of maximizing

$$\mathcal{L}(\Theta) = \log P(\mathbf{X}|\mathbf{C}, \Theta) = \sum_{u=1}^{U} \log P(\mathbf{X}_u|\mathbf{C}_u, \Theta) \quad (1)$$

$P(\mathbf{X}_u|\mathbf{C}_u, \Theta)$ can be obtained by marginalizing over $\boldsymbol{w}_u$:

$$P(\mathbf{X}_u|\mathbf{C}_u, \Theta) = \int_{\boldsymbol{w}_u} P(\mathbf{X}_u|\mathbf{C}_u, \boldsymbol{w}_u, \Theta) f(\boldsymbol{w}_u) d\boldsymbol{w}_u \quad (2)$$

$P(\mathbf{X}_u|\mathbf{C}_u, \boldsymbol{w}_u, \Theta)$ is given by the Total Variability Model:

$$P(\mathbf{X}_u|\mathbf{C}_u, \boldsymbol{w}_u, \Theta) = \prod_{c=1}^{C} \prod_{t:c_{ut}=c} \mathcal{N}(\boldsymbol{x}_{ut};\, \boldsymbol{\mu}_c + \mathbf{T}_c \boldsymbol{w}_u, \boldsymbol{\Sigma}_c)$$

By separating the terms involving $\mathbf{T}$ from rest of the expression, it is possible to factorize $P(\mathbf{X}_u|\mathbf{C}_u, \boldsymbol{w}_u, \Theta)$ as follows:

$$P(\mathbf{X}_u|\mathbf{C}_u, \boldsymbol{w}_u, \Theta) = g(\mathbf{X}_u, \Theta)\, h(\mathbf{X}_u, \Theta), \text{ where:}$$

$$g(\mathbf{X}_u, \Theta) = \prod_{c=1}^{C} \prod_{t:c_{ut}=c} \mathcal{N}(\boldsymbol{x}_{ut};\, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$$

$$h(\mathbf{X}_u, \Theta) = \exp\left[\boldsymbol{w}_u^\top \left(\mathbf{T}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{F}_u - \frac{1}{2} \mathbf{T}^\top \boldsymbol{\Sigma}^{-1} \mathbf{N}_u \mathbf{T} \boldsymbol{w}_u\right)\right]$$

$$\boldsymbol{F}_u = \left[\ \boldsymbol{F}_{u1}^\top \,\vdots\, \ldots \,\vdots\, \boldsymbol{F}_{uC}^\top\ \right]^\top\!\!, \ \boldsymbol{F}_{uc} = \sum_{t:c_{ut}=c} (\boldsymbol{x}_{ut} - \boldsymbol{\mu}_c) \quad (3)$$

where $\boldsymbol{\Sigma}, \mathbf{N}_u$ are $CD \times CD$ block diagonal matrices with c^{th} block given by $\boldsymbol{\Sigma}_c$ and $N_{uc}\mathbf{I}$ respectively, and $N_{uc} = \sum_{t:c_{ut}=c} \delta(c_{ut} - c)$. Substituting this expression for $P(\mathbf{X}_u|\mathbf{C}_u, \boldsymbol{w}_u, \Theta)$ back in (2), and eventually substituting the obtained expression back in (1), we get:

$$\mathcal{L}(\Theta) = \mathcal{L}_1(\Theta) + \mathcal{L}_2(\Theta) + \mathcal{L}_3(\Theta) \quad (4)$$

where: $$\mathcal{L}_1(\Theta) = \sum_{u=1}^{U} \log g(\mathbf{X}_u, \Theta),$$

$$\mathcal{L}_2(\Theta) = \frac{1}{2} \sum_{u=1}^{U} \boldsymbol{F}_u^\top \boldsymbol{\Sigma}^{-1} \mathbf{T} \left(\mathbf{I} + \mathbf{T}^\top \boldsymbol{\Sigma}^{-1} \mathbf{N}_u \mathbf{T}\right)^{-1} \mathbf{T}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{F}_u,$$

$$\mathcal{L}_3(\Theta) = -\frac{1}{2} \sum_{u=1}^{U} \log |\mathbf{I} + \mathbf{T}^\top \boldsymbol{\Sigma}^{-1} \mathbf{N}_u \mathbf{T}|$$

The approach we follow for obtaining $\mathbf{T}^*, \boldsymbol{\Sigma}^*$ that maximize the likelihood given by (4) is as follows:

- Obtain $\mathbf{T}^*(\boldsymbol{\Sigma})$ as a function of $\boldsymbol{\Sigma}$ by maximizing $\mathcal{L}(\mathbf{T}, \boldsymbol{\Sigma})$ with respect to $\mathbf{T}$ while treating $\boldsymbol{\Sigma}$ as constant
- Obtain $\boldsymbol{\Sigma}^*$ by maximizing $\mathcal{L}(\mathbf{T}^*(\boldsymbol{\Sigma}), \boldsymbol{\Sigma})$
- Finally, $\Theta^* = \{\mathbf{T}^*(\boldsymbol{\Sigma}^*), \boldsymbol{\Sigma}^*\}$

2.3.2. Estimating $\mathbf{T}^*(\boldsymbol{\Sigma})$

Since $\mathcal{L}_1(\Theta)$ is constant with respect to $\mathbf{T}$, we have:

$$\mathbf{T}^*(\boldsymbol{\Sigma}) = \arg\max_{\mathbf{T}} J(\mathbf{T}), \text{ where } J(\mathbf{T}) = \mathcal{L}_2(\Theta) + \mathcal{L}_3(\Theta)$$

Let the Cholesky decomposition of $\boldsymbol{\Sigma}^{-1}$ be $\mathbf{L}\mathbf{L}^\top$. Then, we can express $\mathcal{L}_1, \mathcal{L}_2$ in terms of quantities $\widetilde{\boldsymbol{F}}_u, \widetilde{\mathbf{T}}_u$ as:

$$\widetilde{\boldsymbol{F}}_u = \mathbf{N}_u^{-\frac{1}{2}} \mathbf{L}^\top \boldsymbol{F}_u,\ \widetilde{\mathbf{T}}_u = \mathbf{N}_u^{\frac{1}{2}} \mathbf{L}^\top \mathbf{T} \quad (5)$$

$$\mathcal{L}_2(\Theta) = \frac{1}{2} \sum_{u=1}^{U} \left(\widetilde{\boldsymbol{F}}_u^\top \widetilde{\mathbf{T}}_u \left(\mathbf{I} + \widetilde{\mathbf{T}}_u^\top \widetilde{\mathbf{T}}_u\right)^{-1} \widetilde{\mathbf{T}}_u^\top \widetilde{\boldsymbol{F}}_u\right)$$

$$\mathcal{L}_3(\Theta) = -\frac{1}{2} \sum_{u=1}^{U} \log |\mathbf{I} + \widetilde{\mathbf{T}}_u^\top \widetilde{\mathbf{T}}_u| \quad (6)$$

At this point, the maximization can be greatly simplified if we can separate terms involving $\mathbf{T}$ from the summation over u. To that end, we make the approximation $N_{uc} \approx T_u p_c$, which is a reasonable approximation to make if T_u is large enough, since $N_{uc} \to T_u p_c$ as $T_u \to \infty$ by the Law of Large Numbers. For this approximation, and for large T_u, we get:

$$\widetilde{\mathbf{T}}_u \approx \sqrt{T_u}\, \widetilde{\mathbf{T}},\ \ \widetilde{\mathbf{T}} = \mathbf{P}^{\frac{1}{2}} \mathbf{L}^\top \mathbf{T}, \ \text{ and } \ \frac{1}{T_u}\mathbf{I} \approx \frac{1}{T}\mathbf{I}$$

where $\mathbf{P}$ is a $CD \times CD$ block diagonal matrix with c^{th} block given by $p_c\,\mathbf{I}$, and T is the average utterance length. Then, by using the invariance of matrix trace under cyclic permutations, (6) can be simplified as:

$$\mathcal{L}_2(\Theta) = \frac{1}{2} \operatorname{Tr}\left[\left(\sum_{u=1}^{U} \widetilde{\boldsymbol{F}}_u \widetilde{\boldsymbol{F}}_u^\top\right) \widetilde{\mathbf{T}} \left(\frac{1}{T}\mathbf{I} + \widetilde{\mathbf{T}}^\top \widetilde{\mathbf{T}}\right)^{-1} \widetilde{\mathbf{T}}^\top\right]$$

$$\mathcal{L}_3(\Theta) = -\frac{1}{2} \sum_{u=1}^{U} \log |\mathbf{I} + T_u \widetilde{\mathbf{T}}^\top \widetilde{\mathbf{T}}|$$

Let $\widetilde{\mathbf{F}}$ denote a matrix containing $\widetilde{\boldsymbol{F}}_u$ as columns then:

$$\left(\sum_{u=1}^{U} \widetilde{\boldsymbol{F}}_u \widetilde{\boldsymbol{F}}_u^\top\right) = \widetilde{\mathbf{F}}\widetilde{\mathbf{F}}^\top$$

Let the Singular Value Decomposition (SVD) for $\widetilde{\mathbf{F}}, \widetilde{\mathbf{T}}$ be:

$$\widetilde{\mathbf{F}} = \mathbf{U}_F \mathbf{D}_F \mathbf{V}_F^\top,\ \ \widetilde{\mathbf{T}} = \mathbf{U}_T \mathbf{D}_T \mathbf{V}_T^\top$$

where $\mathbf{D}_F \in \mathbb{R}^{CD \times U}$ and $\mathbf{D}_T \in \mathbb{R}^{CD \times K}$ are diagonal with sorted entries: $d_{F_1} \geq \ldots d_{F_{CD}}, d_{T_1} \geq \ldots d_{T_K}$, and $\mathbf{U}_F, \mathbf{U}_T, \mathbf{V}_F, \mathbf{V}_T$ are orthonormal: To find $\mathbf{T}^*$, we need to find $\mathbf{U}_T^*, \mathbf{D}_T^*, \mathbf{V}_T^*$ that maximize the likelihood. Define:

$$\mathbf{S}_F = \widetilde{\mathbf{F}}\widetilde{\mathbf{F}}^\top,\ \widetilde{\mathbf{D}}_T = \mathbf{D}_T \left(\frac{1}{T}\mathbf{I} + \mathbf{D}_T^\top \mathbf{D}_T\right)^{-1} \mathbf{D}_T^\top$$

Then, $\mathcal{L}_2, \mathcal{L}_3$ can be expressed as:

$$\mathcal{L}_2(\Theta) = \frac{1}{2} \operatorname{Tr}\left[\mathbf{S}_F\, \mathbf{U}_T \widetilde{\mathbf{D}}_T \mathbf{U}_T^\top\right]$$

$$\mathcal{L}_3(\Theta) = -\frac{1}{2} \sum_{u=1}^{U} \log |\mathbf{I} + T_u \mathbf{D}_T^\top \mathbf{D}_T|$$

Neither $\mathcal{L}_2(\Theta)$ nor $\mathcal{L}_3(\Theta)$ depend on $\mathbf{V}_T$, so without loss of generality, we can take $\mathbf{V}_T^* = \mathbf{I}$. $\mathcal{L}_3(\Theta)$ does not depend on $\mathbf{U}_T$, so we can obtain $\mathbf{U}_T^*$ by maximizing $\mathcal{L}_2(\Theta)$. To that end, we use the following result:

Theorem ([14, Theorem 4.1]). *Let A, B be $n \times n$ Hermitian matrices, with eigenvalues α_i, β_i respectively, both similarly ordered: $\alpha_1 \geq \cdots \geq \alpha_n, \beta_1 \geq \cdots \geq \beta_n$. Then:*

$$\max_{U \text{unitary}} \operatorname{Tr}\left[AU^\mathsf{T}BU\right] = \sum_{i=1}^{n} \alpha_i \beta_i$$

Substituting $A \leftarrow \mathbf{S}_F, U \leftarrow \mathbf{U}_T^\mathsf{T}, B \leftarrow \widetilde{\mathbf{D}}_T$, in the statement of the theorem, we get:

$$\max_{\mathbf{U}_T \text{ unitary}} \mathcal{L}_2(\Theta) = \frac{1}{2} \operatorname{Tr}\left[\mathbf{D}_F \mathbf{D}_F^\mathsf{T} \widetilde{\mathbf{D}}_T\right]$$

The maximum value can be achieved by setting $\mathbf{U}_T^* = \mathbf{U}_F$. Substituting it back in $J(\mathbf{T})$, we get the following expression in terms of d_{Tk}:

$$J(\mathbf{T}) = \frac{1}{2} \sum_{k=1}^{K} \left[\frac{T d_{F_k}^2 d_{T_k}^2}{1 + T d_{T_k}^2} - \sum_{u=1}^{U} \log(1 + T_u d_{T_k}^2)\right]$$

Taking the derivative with respect to d_{Tk}, then making approximations $(1 + Td_{T_k}^2)^2 \approx T^2 d_{T_k}^4 + 2Td_{T_K}^2$ and $1 + T_u d_{T_K}^2 \approx T_u d_{T_k}^2$, and setting to zero, we get:

$$d_{T_k}^* \approx \begin{cases} \sqrt{\frac{d_{F_k}^2}{UT} - \frac{2}{T}} & \text{if } d_{F_k}^2 \geq 2U \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

To summarize, we get:

$$\widetilde{\mathbf{T}}^*(\mathbf{\Sigma}) = \mathbf{U}_T^* \mathbf{D}_T^*, \ \mathbf{T}^*(\mathbf{\Sigma}) = \mathbf{\Sigma} \mathbf{L} \mathbf{P}^{-\frac{1}{2}} \widetilde{\mathbf{T}}^*(\mathbf{\Sigma}) \tag{8}$$

where $\mathbf{U}_T^* = \mathbf{U}_F$, and the entries of $\mathbf{D}_T^*$ are given by (7).

2.3.3. *Estimating $\mathbf{\Sigma}^*$*

Let $\bar{d}_{F_k} = \max(d_{F_k}, \sqrt{2U})$. Substituting the obtained $\mathbf{T}^*(\mathbf{\Sigma})$ in the likelihood, we get the following expression for $2J(\mathbf{T})$:

$$\sum_{k=1}^{K} \left(d_{F_k}^2 \frac{\bar{d}_{F_k}^2 - 2U}{\bar{d}_{F_k}^2 - U} - \sum_{u=1}^{U} \log\left[1 + T_u \left(\frac{\bar{d}_{F_k}^2}{UT} - \frac{2}{T}\right)\right]\right)$$

Here, some approximations are necessary to simplify the expression. The first term grows almost linearly with d_{Fk}^2, whereas the other terms reduce logarithmically, so when we maximize the likelihood, we expect the linear term to dominate, and push d_{Fk} towards a high value. Expecting a large d_{Fk}, we approximate $\bar{d}_{F_k}^2 - 2U \approx \bar{d}_{F_k}^2 - U$, and drop the logarithmic terms since their contribution can be rendered negligible in comparison to the first for large d_{Fk}. Since we expect $\mathbf{F}$ to be low rank, $\widetilde{\mathbf{F}}$ would also be low rank, and the summation in the first term can be increased up to index CD instead of K without a significant effect. Making these approximations, we can get:

$$2J(\mathbf{T}) \approx \sum_{k=1}^{CD} d_{F_k}^2 = \operatorname{Tr}(\widetilde{\mathbf{F}}\widetilde{\mathbf{F}}^\mathsf{T}) = \operatorname{Tr}\left[\mathbf{\Sigma}^{-1} \sum_{u=1}^{U} \mathbf{N}_u^{-1} \boldsymbol{F}_u \boldsymbol{F}_u^\mathsf{T}\right]$$

Substituting in (4), we get the following objective function to minimize with respect to $\mathbf{\Sigma}$: $\left(\text{where } N_c = \sum_{u=1}^{U} N_{uc}\right)$

$$\sum_{c=1}^{C} \operatorname{Tr}\left[\mathbf{\Sigma}_c^{-1} \left(\mathbf{S}_{Xc} - \sum_{u=1}^{U} \frac{1}{N_{uc}} \boldsymbol{F}_{uc} \boldsymbol{F}_{uc}^\mathsf{T}\right)\right] + N_c \log|\mathbf{\Sigma}_c|,$$

$$\mathbf{S}_{Xc} = \sum_{u=1}^{U} \mathbf{S}_{uc}, \ \mathbf{S}_{uc} = \sum_{t:c_{ut}=c} (\boldsymbol{x}_{ut} - \boldsymbol{\mu}_c)(\boldsymbol{x}_{ut} - \boldsymbol{\mu}_c)^\mathsf{T} \tag{9}$$

Minimizing with respect to $\mathbf{\Sigma}_c$, we get:

$$\mathbf{\Sigma}_c^* = \frac{1}{N_c} \left(\mathbf{S}_{Xc} - \sum_{u=1}^{U} \frac{1}{N_{uc}} \boldsymbol{F}_{uc} \boldsymbol{F}_{uc}^\mathsf{T}\right) \tag{10}$$

By algebraic manipulation, (10) can be rearranged to a Positive Semi-Definite (PSD) matrix:

$$\begin{gathered} \mathbf{\Sigma}_c^* = \frac{1}{N_c} \sum_{u=1}^{U} \sum_{t:c_{ut}=c} (\boldsymbol{x}_{ut} - \bar{\mathbf{X}}_{uc})(\boldsymbol{x}_{ut} - \bar{\mathbf{X}}_{uc})^\mathsf{T}, \\ \bar{\mathbf{X}}_{uc} = \frac{1}{N_{uc}} \sum_{t:c_{ut}=c} \boldsymbol{x}_{ut} \end{gathered} \tag{11}$$

2.3.4. *Summary of SVD Estimation*

Overall, the estimation procedure using SVD has been summarized in the procedure below. Although our derivation is based on hard component alignment for simplicity, soft posteriors can be used in practice for computing the statistics.

```
Procedure TVM-SVD-Estimation
Input  : Feature Vectors $\boldsymbol{x}_{ut}$, Posteriors $\gamma_{ut}$
Output: Total Variability Matrix $\mathbf{T}$, Covariance Matrix $\mathbf{\Sigma}$
1: for $u = 1$ to $U$ do
2:     Collect statistics $\mathbf{N}_u, \boldsymbol{F}_u, \mathbf{S}_u$            ▷ Eq (3),(9)
3: Get $\mathbf{\Sigma}$ from $\mathbf{S}_u$, $\boldsymbol{F}_u$                      ▷ Eq (10)/(11)
4: Normalize the statistics $\boldsymbol{F}_u$ to get $\widetilde{\boldsymbol{F}}_u$              ▷ Eq (5)
5: Get SVD of $\widetilde{\mathbf{F}}$ using randomized algorithms       ▷ Sec 3.1
6: Get the Total Variability Matrix $\mathbf{T}$                ▷ Eq (8)
```

3. Advantages of SVD Estimation

3.1. Computational Complexity of Parameter Estimation

Although we have shown that we can obtain the Total Variability Matrix by computing the SVD of matrix $\widetilde{\mathbf{F}}$, it is not immediately clear that this would be a better choice than EM computationally, especially because the matrix $\widetilde{\mathbf{F}}$ is of dimension $CD \times U$, which is typically fairly large. Fortunately, there are randomized algorithms available that can solve this problem very efficiently. A summary of various algorithms and the associated bounds can be found in [12]. To illustrate the efficiency of these algorithms, we highlight a few key points:

- Although the outcome is probabilistic, the probability of failure is a user specified parameter, and can be rendered negligible (say 10^{-15}), with a nominal impact on the computational resources required
- Most of the computationally intensive steps in these algorithms are parallelizable, allowing the advantage of exploiting a large number of parallel nodes, if available
- The computations do not require loading the entire matrix to memory, and can be modified to require only a single pass over the matrix stored on a disk

3.2. Computational Complexity of ivector Estimation

The MAP estimate of the ivector given model parameters Θ and utterance statistics $\boldsymbol{F}_u, \mathbf{N}_u$ is given as:

$$\boldsymbol{w}_u^* = \left(\mathbf{I} + \mathbf{T}^\mathsf{T} \mathbf{\Sigma}^{-1} \mathbf{N}_u \mathbf{T}\right)^{-1} \mathbf{T}^\mathsf{T} \mathbf{\Sigma}^{-1} \boldsymbol{F}_u \tag{12}$$

By making similar approximations to those made in section 2.3.2, we can potentially simplify (12) to:

$$\boldsymbol{w}_u^* = \frac{1}{\sqrt{T_u}} \left(\frac{1}{T_u}\mathbf{I} + \widetilde{\mathbf{T}}^{\mathsf{T}}\widetilde{\mathbf{T}} \right)^{-1} \widetilde{\mathbf{T}}^{\mathsf{T}} \widetilde{\boldsymbol{F}}_u \qquad (13)$$

Since $\widetilde{\mathbf{T}}^{\mathsf{T}}\widetilde{\mathbf{T}} = \mathbf{D}_T^{\mathsf{T}}\mathbf{D}_T$ is diagonal, matrix inversion is greatly simplified and enables much faster ivector extraction.

3.3. Interpretability of ivectors

Consider for example the singular values of the matrix $\widetilde{\boldsymbol{F}}$ obtained for the Wall Street Journal (WSJ) si284 corpus, shown in Figure 1. It is apparent that most of the variability in the matrix is explained by the first few subspace dimensions.

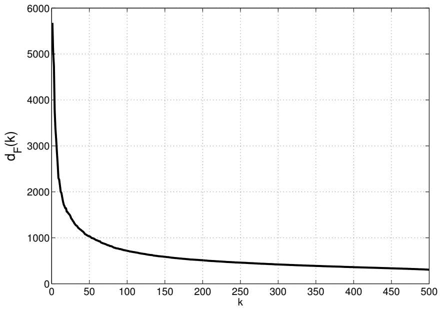

Figure 1: Singular Values of $\widetilde{\boldsymbol{F}}$

For further illustration, first two dimensions of the extracted ivectors for the WSJ test set eval92, are shown in figure 2. Different speakers are shown by markers of different color, male and female speakers are shown by circular and triangular markers respectively. It is clear that just the first two ivector dimensions already capture quite a lot of the speaker variability, and yield fairly congregated clusters.

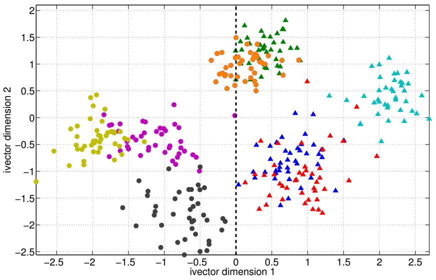

Figure 2: First two ivector dimensions for WSJ eval92 data

Moreover, observing the speaker gender reveals yet another interesting fact: with the exception of a few utterances from one speaker, the value of the first ivector dimension is negative for male utterances and positive for female utterances, indicating that it is implicitly encoding gender information. In other words, the model has effectively identified gender as the most important factor of acoustic variability in the data, even though no labels are provided during training. By similarly exploring correlations between speaker metadata and the first few ivector dimensions, it might become possible to assign them interpretable notions, while also understanding the relative impact of different factors on acoustic variability.

4. Experimental Results

We conducted experiments for Language Identification (LID) on the DARPA Robust Automatic Transcription of Speech (RATS) database. The database consists of audio recordings of varying lengths, and corrupted by different noise types of varying SNRs. Train and test data splits were chosen according to the description in [7]. The database contains audio from six classes: five target languages and a class corresponding to 10 non-target languages. The data used for training the Total Variability Matrix consists of 96000 recordings (16000 from each class) of 30s each. In addition, there are four datasets, consisting of utterances with length 10s, 5s, 3s, and 1s respectively. For each of these splits, a labeled training set of 96000 utterances is available for classifier training, and a test set of 2000 utterances of the same length is used to evaluate the performance.

Table 1: LID Results

	EM		**SVD-1**		**SVD-2**	
T_θ	33.5 hrs		**30 min**		**30 min**	
T_w	1.13 sec		**0.12 sec**		1.13 sec	
Dur	*EER*	*ACC*	*EER*	*ACC*	*EER*	*ACC*
10	**8.30**	83.50	8.78	**83.60**	9.26	81.15
5	**10.04**	**80.80**	10.40	80.55	10.94	78.05
3	**13.41**	72.75	13.59	**74.25**	14.97	70.55
1	**21.59**	56.45	22.13	**58.35**	22.61	54.65

For each frame, we obtained 20-dimensional MFCC vectors, concatenated with delta coefficients. A UBM of 2048 components, and an ivector model of 400 dimensions were trained on the 30s dataset. For classification, we trained a Support Vector Machine (SVM) with a fifth order polynomial kernel. Within Class Covariance Normalization (WCCN) is used prior to classifier training for compensating unwanted sources of variability.

The baseline model was obtained by training the Total Variability Matrix using EM algorithm [2], implemented using the Kaldi toolkit [15]. The results have been summarized in Table 1. SVD-1 refers to ivector extraction given by (13), whereas SVD-2 refers to ivector extraction given by (12). The performance results for EM and SVD-1 are comparable, with EM being slightly better in terms of Equal Error Rate (EER), but SVD-1 being slightly better in terms of classification accuracy (ACC). The exact but slower version, given by SVD-2, is also slightly worse, suggesting that maintaining a consistency in terms of the approximations made during parameter estimation and ivector extraction is beneficial.

In terms of time taken for parameter estimation (T_θ), however, there is a stark difference between the EM algorithm and SVD estimation. Using 16 processors in parallel, the total time taken for parameter estimation using the EM algorithm (excluding the time required for feature extraction and posterior computation) was 120,517s (roughly **33.5 hours**). Using the same number of processors, the total time required for SVD estimation was 1807s (roughly **30 minutes**). It should also be noted that we implemented the proto algorithm described in [12], primarily because of its simplicity. However, many faster algorithms have been described in [12], which have the potential to reduce the computational requirement even further. In addition, EM was also much slower compared to SVD-1 in terms of average time taken for ivector extraction (T_w), owing primarily to the computation of the inverse matrix in (12).

5. Conclusion

We have presented an algorithm based on randomized SVD that significantly reduces the computation time required for TVM parameter estimation and ivector computation. It also opens up a potential opportunity for gaining insights about interpretability of the extracted ivectors.

6. References

[1] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-end factor analysis for speaker verification,” *IEEE Trans. Audio, Speech & Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[2] P. Kenny, G. Boulianne, and P. Dumouchel, “Eigenvoice modeling with sparse training data,” *Speech and Audio Processing, IEEE Transactions on*, vol. 13, no. 3, pp. 345–354, 2005.

[3] D. Martınez, O. Plchot, L. Burget, O. Glembek, and P. Matejka, “Language recognition in ivectors space,” *Proceedings of Interspeech, Firenze, Italy*, pp. 861–864, 2011.

[4] N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, and R. Dehak, “Language recognition via i-vectors and dimensionality reduction,” in *INTERSPEECH*, 2011, pp. 857–860.

[5] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, “Speaker adaptation of neural network acoustic models using i-vectors,” in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013, pp. 55–59.

[6] A. W. Senior and I. Lopez-Moreno, “Improving DNN speaker independence with i-vector inputs.” in *ICASSP*, 2014, pp. 225–229.

[7] M. Van Segbroeck, R. Travadi, C. Vaz, J. Kim, M. P. Black, A. Potamianos, and S. S. Narayanan, “Classification of cognitive load from speech using an i-vector framework.” in *INTERSPEECH*, 2014, pp. 751–755.

[8] M. Li, A. Tsiartas, M. Van Segbroeck, and S. S. Narayanan, “Speaker verification using simplified and supervised i-vector modeling,” in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7199–7203.

[9] O. Glembek, L. Burget, P. Matějka, M. Karafiát, and P. Kenny, “Simplification and optimization of i-vector extraction,” in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 4516–4519.

[10] S. Cumani and P. Laface, “Factorized sub-space estimation for fast and memory effective i-vector extraction,” *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 1, pp. 248–259, 2014.

[11] P. Kenny, “A small footprint i-vector extractor.” in *Odyssey*, 2012, pp. 1–6.

[12] N. Halko, P.-G. Martinsson, and J. A. Tropp, “Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions,” *SIAM review*, vol. 53, no. 2, pp. 217–288, 2011.

[13] M. Omar, “Fast approximate i-vector estimation using PCA,” in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4495–4499.

[14] I. Coope and P. Renaud, “Trace inequalities with applications to orthogonal regression and matrix nearness problems.” *JIPAM. Journal of Inequalities in Pure and Applied Mathematics*, vol. 10, no. 4, 2009.

[15] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz *et al.*, “The Kaldi speech recognition toolkit,” 2011.