修士論文

テンソル分解に基づく音声表現とその言語識別・話者識別への応用

2016 年 2 月 4 日

指導教員　峯松 信明 教授

電気系工学専攻

37-146454　鈴木 颯

内容梗概

音声認識の技術は、現在では様々なアプリケーションにおいて利用されるようになった。特に、カーナビゲーションシステムやスマートフォンにおける Siri、しゃべってコンシェル等の秘書機能アプリケーション等、情報端末に対してユーザーが話しかけることで操作を行なうものが日常的に利用されている。但し、これらに搭載されている音声認識システムの多くは認識対象が入力音声の発話内容のみであり、想定される発話内容のパターンも限られている。しかし近年では、コンピュータとの対話や一般的な日常会話、複数人の会話等を想定したより高度な音声インターフェースに対してニーズが高まってきている。このような音声インターフェースを実現するためには、発話内容だけでなく「いつ、誰が、何を、どのように話したか」等の音声の持つ属性を認識する技術が必要となる。本研究では、音声から話者を推定するタスクである話者識別と、言語を推定するタスクである言語識別の二つのタスクに着目し、識別精度の向上を目指した。音声は話者や収録環境等の条件によって多様に変化し、これら非言語的特徴の変動による識別性能低下が二つのタスクにおける課題の一つとなっている。入力音声から識別に有効な特徴量を抽出し、抽出された特徴量を基に適切に識別する二つのプロセスにおいてこれらの課題の解決を目指した。

近年、言語識別・話者識別分野ではこの課題に着目して提案された i-vector が標準的な特徴量表現の手法になっているが、これは各発話を混合ガウス分布 (Gaussian Mixture Model; GMM) でモデル化し、GMM の各分布の平均ベクトルを連結した GMM supervector (GMM-SV) を因子分析に基づき次元圧縮することによって得られる特徴量である。また話者識別分野では、GMM-SV 表現に内在する問題に着目し、i-vector のアプローチを拡張したテンソル分解に基づく話者情報表現が提案されている。行および列がそれぞれ GMM の各分布と平均ベクトルに対応するような行列によって一発話を表現し、多数話者分の行列をテンソルとして扱い、テンソル解析を導入することで話者情報を表現している。この手法では、GMM によって複数要因からの音響的変動を捉え、それらの持つ関係性を明示的に考慮した上でより適切に分離することをねらっている。本研究ではテンソル分解に基づく音声表現を言語識別・話者識別の二つのタスクに適用し、その効果を実験的に検証した。実験結果から、テンソル分解に基づく音声表現が特に話者識別に有効であることが分かった。言語識別では有効性が確認できなかったが、話者識別・言語識別のタスクの違いに基づき、両者において識別に有効な特徴量表現が異なる可能性について指摘した。

また、近年の言語識別・話者識別分野における識別手法に関しては、Probabilistic Linear Discriminant Analysis (PLDA) が標準的となっている。PLDA はベクトルで表現される特徴量を想定した識別器であるが、テンソル分解に基づく音声特徴量のように行列で表現される特徴量の識別を行なう際には、行列の各行と各列の関係性を考慮することでより適切な識別が可能になることが期待される。そこで本研究では、テンソル分解に基づく音声特徴量により適した識別手法として、PLDA を行列変量に拡張した Matrix Variate PLDA (MV-PLDA) を提案し、これについても言語識別・話者識別の二つのタスクにおいて識別性能を実験的に検証した。言語識別・話者識別においては MV-PLDA の有効性については確認できず、MV-PLDA の制約の強さが識別に悪影響を及ぼしている可能性を示唆する結果となった。

目次

図目次

表目次

第1章

序論

1.1　研究の背景

音声認識の技術は、現在では様々なアプリケーションにおいて利用されるようになった。特に、カーナビゲーションシステムやスマートフォンにおける Siri[1]、しゃべってコンシェル [2] 等の秘書機能アプリケーション等、情報端末に対してユーザーが話しかけることで操作を行なうものが日常的に利用されている。但し、これらに搭載されている音声認識システムの多くは認識対象が入力音声の発話内容のみであり、想定される発話内容のパターンも限られている。しかし近年では、コンピュータとの対話や一般的な日常会話、複数人の会話等を想定したより高度な音声インターフェースに対してニーズが高まってきている。このような音声インターフェースを実現するためには、発話内容だけでなく「いつ、誰が、何を、どのように話したか」という、音声の持つ属性を認識する技術が必要となる [1]。このような技術の応用例として、自動議事録作成アプリケーションや音声翻訳、国際コールセンターにおけるオペレータ切り替えが挙げられる [2, 3, 4]。自動議事録作成アプリケーションでは音声から「誰が話したのか」を推定する必要がある。また、これを多言語対応にする場合や音声翻訳、国際コールセンターにおける言語毎のオペレータ切り替えでは音声から「どの言語が話されたのか」を推定する必要がある。音声から話者を推定するタスクを話者識別、言語を推定するタスクを言語識別と呼ぶ [5, 6]。本研究では言語識別・話者識別の二つのタスクに着目し、識別精度の向上を目指す。いずれのタスクにおいても、識別性能を向上させるためには二つの点が重要となる。一つ目は入力音声から識別に有効な特徴を抽出すること、二つ目は抽出された特徴量を基に適切に識別することである。音声は話者や収録環境等の条件によって多様に変化し、これら非言語的特徴の変動による識別性能低下が二つのタスクにおける課題の一つとなっている。

近年、言語識別・話者識別分野ではこの課題に着目して提案された i-vector が標準的な特徴量表現の手法になっているが [7, 8]、これは各発話を混合ガウス分布 (Gaussian Mixture Model; GMM) でモデル化し、GMM の各分布の平均ベクトルを連結した GMM supervector (GMM-SV) を因子分析に基づき次元圧縮することによって得られる特徴量である [9]。また話者識別分野では、GMM-SV 表現に内在する問題に着目し、i-vector のアプローチを拡張したテンソル分解に基づく話者情報表現が提案されている [10]。[10] では行および列がそれぞれ GMM の各分布と平均ベクトルに対応するような行列によって一発話を表現し、多数話者分の行列をテンソルとして扱い、これに対しテンソル解析を導入することで話者情報を表現している。この手法では、GMM によって複数要因からの音響的変動を捉え、それらの持つ関係性を明示的に考慮した上でより適切に分離することをねらっており、話者識別だけでなく言語識別にも適用可能であると考えられる。[10] では基礎的な検討のみに留まるため、本研究ではテンソル分解に基づく音声の特徴量表現を言語識別・話者識別の二つのタスクに適用し、その効果を改めて実験的に検証する。

また、近年の言語識別・話者識別分野における識別手法に関しては、Probabilistic Linear Discriminant Analysis (PLDA) が標準的となっている [11, 12]。これはクラスに依存する潜在変数を用いてクラス内の変動とクラス間の変動を記述する生成モデルであり、同じく生成モデルである GMM に基づいて計算される i-vector の識別によく用いられる。PLDA はベクトルで表現される特徴量を想定した識別器であるが、テンソル分解に基づく音声特徴量のように行列で表現される特徴量の識別を行なう際には、行列の各行と各列の関係性を考慮することでより適切な識別が可能になることが期待される。そこで本研究では、テンソル分解に基づく音声特徴量により適した識別手法として、PLDA を行列変量に拡張した Matrix Variate PLDA (MV-PLDA) を提案し、

[1] https://www.apple.com/jp/ios/siri/
[2] https://www.nttdocomo.co.jp/service/information/shabette_concier/

これについても言語識別・話者識別の二つのタスクにおいて識別性能を実験的に検証する。

1.2　本論文の構成

本論文は、全 5 章から構成される。まず第 2 章では、従来の言語・話者情報表現の手法として GMM-SV, i-vector とそれらに関連する手法と要素技術について述べたのち、従来の識別手法について述べる。第 3 章では GMM-SV, i-vector の問題点について述べ、それを解決するテンソル分解に基づく言語・話者情報表現の手法と、新しい識別手法として MV-PLDA について述べる。第 4 章では、言語識別実験と話者識別実験の二つの実験を行ない、提案手法による識別性能を検証したのち、言語識別・話者識別のタスクの違いに基づいて二つの実験結果について考察する。最後に、第 5 章で本論文をまとめ、今後の課題について述べる。

第2章

従来の言語・話者情報表現と識別

2.1　はじめに

音声が入力されその言語・話者が識別されるまでには、入力音声からの特徴量抽出とその識別という大きく分けて二つのプロセスを経る。近年の話者識別は、混合ガウス分布 (Gaussian Mixture Model; GMM) に基づく話者モデルと最尤基準による識別が基礎となるが [6]、その後 Support Vector Machine (SVM) が識別手法として導入され、GMM をそのまま識別に用いるのではなく特徴量に落としこむ手法に変遷していく [9]。話者情報表現の手法として GMM に基づく GMM supervector (GMM-SV) や GMM-SV に対して次元圧縮を施すことによって得られる i-vector が提案されてからはこれらが標準的に用いられるようになった [8]。

一方言語識別については、類似したタスクである方言分類も含めると音素の出現頻度に基づく手法 [13] や非言語的特徴の変動に対して頑健な音声表現である音声の構造的表象を用いた手法 [14] 等の様々な手法が検討されてきたが、近年は話者識別と同様の変遷を辿っており [15]、現在では i-vector が言語情報表現においても標準的な手法となっている [7]。

本章では、従来の言語・話者情報表現と識別の手法について説明する。まず、2.2 節では音声から抽出される基礎的な音響特徴量について述べ、2.6 節からはこれを基礎としたよりハイレベルな話者・言語情報表現の従来手法として GMM-SV と i-vector、さらに i-vector に関連する手法として声質変換分野で提案された Eigenvoice に基づく話者情報表現について述べる。次に、2.7 節からは従来の識別手法として Support Vector Machine (SVM) と Probabilistic Linear Discriminant Analysis (PLDA) について述べる。

2.2　音響特徴量

2.2.1　ケプストラム

音声信号は、声帯における音源の生成、声道による音響変化、放射による音響変化の 3 つの過程を経て他者に聴取される（ソースフィルタモデル）。発話内容や話者の声質は主に声道形状に表れ、物理的には音声信号のスペクトル包絡に表れる。これらを捉える特徴量として、ケプストラムが音声認識において広く用いられている。

ケプストラムは、音声信号のパワースペクトルの対数に逆フーリエ変換を施したものとして定義される。図 2.1 に音声信号からのケプストラム抽出の流れを示す。まず音声波形にハミング窓やハニング窓等の窓関数を乗ずることによって数十 ms 程度のフレームを切り出し、それに対して離散フーリエ変換 (Discrete Fourier Transform; DFT) を施して対数パワースペクトルを抽出する。次に対数パワースペクトルに逆離散フーリエ変換 (Inverse DFT; IDFT) を施すことで、そのフレームにおけるケプストラムが得られる。窓を時間方向にシフトさせ、各フレームに対してこれらの操作を行なうことで音声波形がケプストラムベクトル系列に変換される。ケプストラムの低次成分に対してフーリエ変換を施すとスペクトル包絡が得られる。

2.2.2　聴覚的特性に基づくケプストラム

人間の聴感覚特性は周波数に対してほぼ対数的であり、低い周波数ではその分解能が高く、高い周波数ほど低くなることが知られている。この周波数感度はメル尺度と呼ばれ、周波数 f[Hz] に対して

$$f_{mel}(f) = 2595 \log_{10}(1 + \frac{f}{700}) \tag{2.1}$$

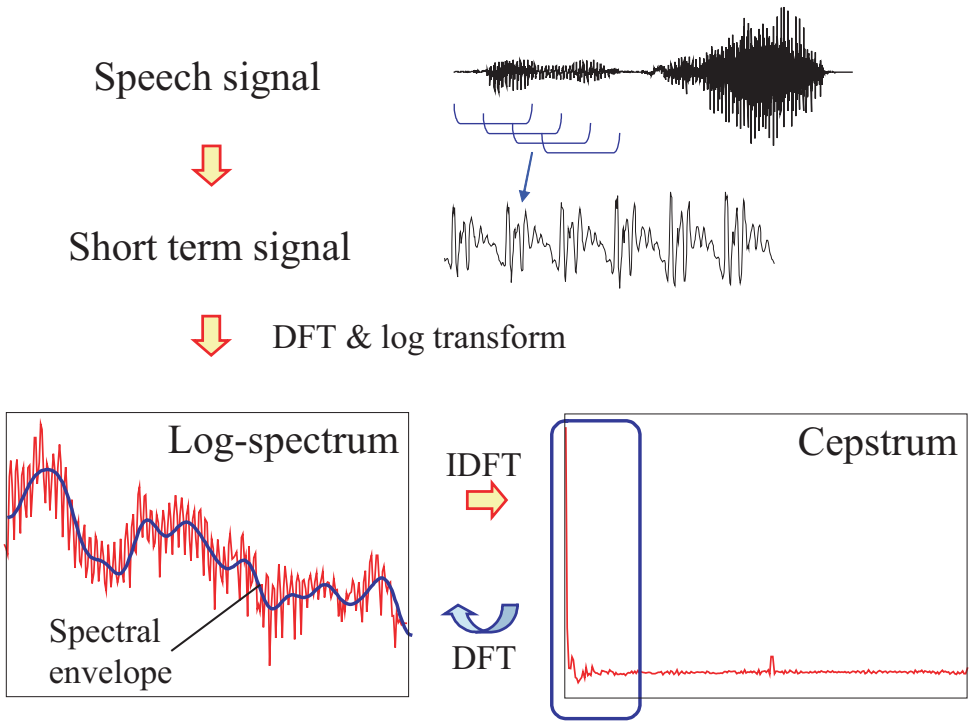

図 2.1: 音声信号からのケプストラム抽出

で近似される。ケプストラムにメル尺度を反映させた音響特徴量として Mel-Frequency Cepstrum Coefficient (MFCC) があり、現在では音声認識において標準的な音響特徴量となっている。

対数パワースペクトルに対し、式 (2.1) で表される周波数ウォーピングを施すことでメル化対数パワースペクトルが得られ（図 2.2 参照）、これに対して逆コサイン変換を施すことで MFCC が得られる [16]。

2.2.3　デルタケプストラムと Shifted Delta Cepstrum (SDC)

ケプストラムに加えて、音声信号の時間的な変化を記述する音響特徴量として各フレームにおけるケプストラムの時間微分（速度成分）とさらにその時間微分（加速度成分）がよく用いられる。前者はデルタ (Δ) ケプストラム、後者はデルタデルタ (ΔΔ) ケプストラムと呼ばれる。一方、言語識別・話者識別分野では複数のフレームにおけるデルタケプストラムを連結した Shifted Delta Cepstrum (SDC) が用いられることも多い [17]。フレーム t における SDC の計算の概略を 図 2.3 に示す。SDC は N, d, P, k の四つのパラメータを持ち、N はケプストラムの次元数、d はデルタケプストラムを計算する際に考慮する隣接フレーム数、P はデルタケプストラムを計算するブロックのシフト幅、k はブロックの個数を表す。各ブロックで計算されたデルタケプストラム $\Delta c(t), \Delta c(t+P), \cdots, \Delta c(t+(k-1)P)$ を連結することで、フレーム t における kN 次元の SDC が得られる。

2.2.4　ケプストラムにおける非言語的特徴の正規化

ケプストラムは声道形状を反映した音響特徴量であり、発話内容に加えて音声の収録環境や話者性等の非言語的特徴を含んでいる。これら非言語的特徴に対する正規化の手法の代表的なもの

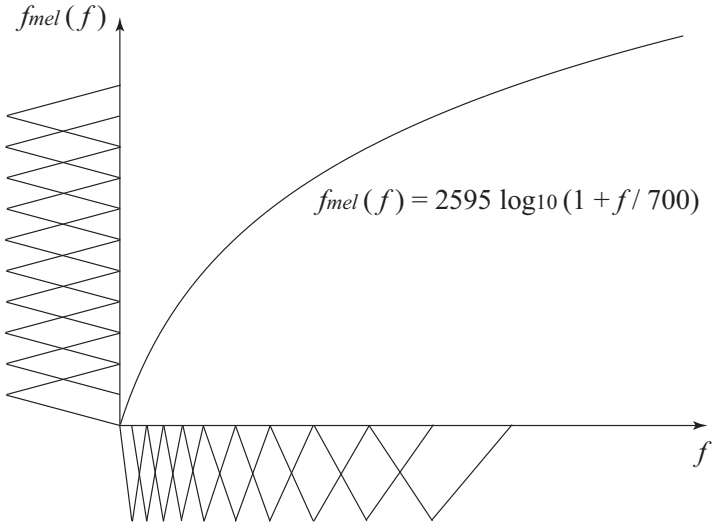

図 2.2: メル周波数とその軸上に等間隔で配置された三角窓

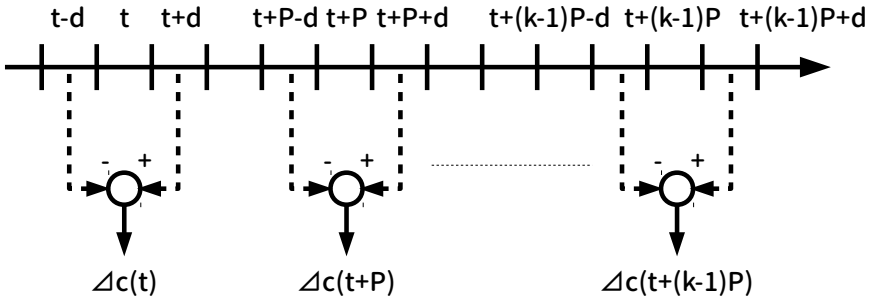

図 2.3: フレーム t における SDC の計算

としてケプストラム平均正規化 (Cepstrum Mean Normalization; CMN)、声道長正規化 (Vocal Tract Length Normalization; VTLN) がある。本節ではこの二つの手法について述べる。

i)　ケプストラム平均正規化

ケプストラム平均正規化 (Cepstrum Mean Normalization; CMN) は、スペクトルに対する乗算で表現される変動を正規化する手法である [18]。CMN では、音声のケプストラム系列から特定の時間幅毎に平均ケプストラムを差し引くことで正規化を行なう。音響機器の特性の変動はスペクトルに対する乗算で表現され、スペクトルに対する乗算はケプストラムに対する加算と等価であるため、ケプストラムの平均を差し引くことで変動を消去することができる。

ii)　声道長正規化

声道長正規化 (Vocal Tract Length Normalization; VTLN) は、話者の声道長の違いに起因するケプストラムの線形変換性の変動を正規化する手法である [19]。話者間の声道長の違いはフォルマント周波数を上下させるため、これが静的なバイアスとして音声特徴量を変化させる。標準的な VTLN では、これを周波数ウォーピングによってモデル化し、入力音声の不特定話者音響モ

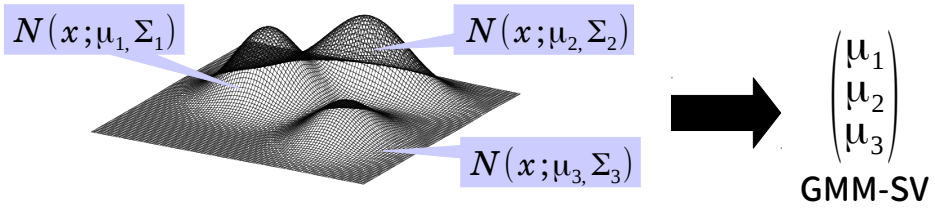

図 2.4: GMM-SV の抽出

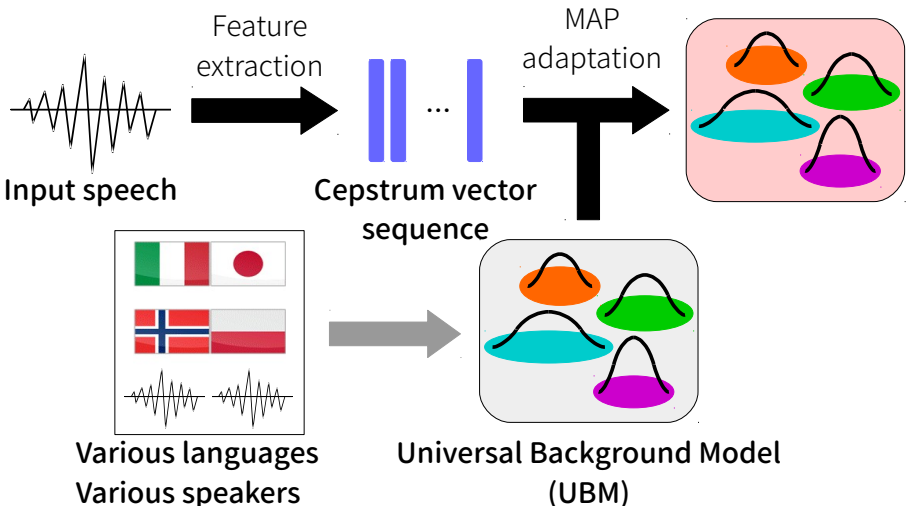

図 2.5: GMM の MAP 推定

デルに対する尤度を最大化するようにウォーピング係数を推定し、その結果を用いて正規化を行なっている [20]。

2.3　GMM-supervector (GMM-SV)

2.3.1　概要

GMM-SV は、一発話を混合ガウス分布 (Gaussian Mixture Model; GMM) によってモデル化し、GMM を構成する各ガウス分布の平均ベクトルを一列に連結した特徴量である（図 2.4 参照）[9]。GMM は以下のように複数のガウス分布の重み付き線形和で表される多峰性の確率分布である（図 2.4 左参照）。D 次元ベクトル $\boldsymbol{x}$ を確率変数とした多次元 GMM の確率密度関数 $p(\boldsymbol{x})$ は以下の式で表される。

$$p(\boldsymbol{x}) = \sum_{m=1}^{M} w_m \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) \tag{2.2}$$

$$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) = \frac{1}{(\sqrt{2\pi})^D \sqrt{|\boldsymbol{\Sigma}_m|}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_m)^\top \boldsymbol{\Sigma}_m^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_m)\right\} \tag{2.3}$$

M は混合数、w_m は混合重み、$\boldsymbol{\mu}_m$ は D 次元の平均ベクトル、$\boldsymbol{\Sigma}_m$ は D 行 D 列の分散共分散行列であり、混合重み w_m の和は 1 となる。GMM の各分布は音素や単音のような音響的な特徴

を捉えていることが期待できるため、GMM-SV はそれらの音響的な特徴を発話単位で表現した特徴量と考えることができる（ここで、音素はある言語においてその言語を理解する人が区別している音声の（最小）単位を指し、単音は音素を実際に発声したときの音、すなわち言語に依存しない（最小）単位を指す）。

2.3.2　発話 GMM の推定

i)　概要

GMM-SV では一発話の GMM の推定を行なうが、一発話から直接 GMM を推定すると、GMM の各分布のインデックスの付け方が発話間で統一されず、GMM-SV の各次元が持つ言語的意味も発話によって異なってしまうことになる。これを避けるため、あらかじめ様々な音声から統計的に話者・言語非依存の Universal Background Model (UBM) を構築しておき、UBM を事前分布として入力発話に対する最大事後確率基準による推定 (MAP 推定) を行なうことで、発話 GMM を得る（図 2.5 参照）。

ii)　GMM-UBM の最尤推定

GMM-UBM は、話者・言語非依存のモデル、すなわち人間の声そのものを表す GMM である。大量の話者・言語のデータを用いることで、話者及び言語属性が隠れ変数として扱われ、統計的には話者・言語非依存モデルとして GMM が推定される。

最尤推定はモデルパラメータ推定法の一つであり、入力データ $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N\}$ が得られたとき、以下の式に示すように尤度最大化に基づいてモデルパラメータ $\hat{\theta}$ を推定する。

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, p(\boldsymbol{X}|\theta) \tag{2.4}$$

式 (2.4) は対数尤度最大化の式に変形でき、$\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N$ が独立であるという仮定のもとではさらに以下のように変形できる。

$$\begin{aligned}
\hat{\theta} &= \underset{\theta}{\operatorname{argmax}} \ln p(\boldsymbol{X}|\theta) && (2.5)\\
&= \underset{\theta}{\operatorname{argmax}} \ln p(\boldsymbol{x}_1|\theta)p(\boldsymbol{x}_2|\theta)\cdots p(\boldsymbol{x}_N|\theta) && (2.6)\\
&= \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \ln p(\boldsymbol{x}_i|\theta) && (2.7)
\end{aligned}$$

となる。従って、GMM の最尤推定は以下のようになる。

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \ln \left\{ \sum_{m=1}^{M} w_m \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) \right\} \tag{2.8}$$

式 (2.8) を直接解くことは困難であるため、Expectation-Maximization (EM) アルゴリズムを用いる。EM アルゴリズムは θ が収束するまで繰り返し計算を行なうアルゴリズムであり、手順は以下のようになる。

1. $w_m, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m$ を初期化する
2. 現在のパラメータを用いて負担率 $\gamma_{m,i}$ を更新する (E-Step)

3. 更新した $\gamma_{m,i}$ を用いて $w_m, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m$ を更新する (M-Step)
4. 前回のステップからの変化率が閾値よりも小さければ収束したとして終了、そうでなければ手順 2 に戻る

E-Step, M-Step における更新式は以下のようになる。
E-Step:

$$\gamma_{m,i} = \frac{w_m \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}{\sum_{m'=1}^{M} w_{m'} \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_{m'}, \boldsymbol{\Sigma}_{m'})} \tag{2.9}$$

M-Step:

$$N_m = \sum_{i=1}^{N} \gamma_{m,i} \tag{2.10}$$

$$\hat{w}_m = \frac{N_m}{N} \tag{2.11}$$

$$\hat{\boldsymbol{\mu}}_m = \frac{1}{N_m} \sum_{i=1}^{N} \gamma_{m,i} \boldsymbol{x}_i \tag{2.12}$$

$$\hat{\boldsymbol{\Sigma}}_m = \frac{1}{N_m} \sum_{i=1}^{N} \gamma_{m,i} (\boldsymbol{x}_i - \boldsymbol{\mu}_m)(\boldsymbol{x}_i - \boldsymbol{\mu}_m)^\top \tag{2.13}$$

iii)　UBM を事前分布とした発話 GMM の MAP 推定

MAP 推定はモデルパラメータ推定法の一つであり、以下のように学習データ $\boldsymbol{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$ が与えられたときのモデルパラメータの事後確率を最大化するような推定を行なう。

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, p(\theta|\boldsymbol{X}) \tag{2.14}$$

ベイズの定理を用いて変形し、対数をとると上の式は以下のようになる。

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, p(\boldsymbol{X}|\theta) p(\theta) \tag{2.15}$$

$$= \underset{\theta}{\operatorname{argmax}} [\ln p(\boldsymbol{X}|\theta) + \ln p(\theta)] \tag{2.16}$$

すなわち、モデルパラメータの事後確率 $p(\theta|\boldsymbol{X})$ の最大化は対数尤度 $\ln p(\boldsymbol{X}|\theta)$ と対数事前確率 $\ln p(\theta)$ の和の最大化に等しい。従って、MAP 推定は最尤推定を事前分布の方向へ抑制するような推定になっていることが分かる。

以下に MAP 推定 による GMM の各分布の平均ベクトルの推定手順を示す。データ $\boldsymbol{x}_i$ が観測されたとき、これが GMM の m 番目のガウス分布から生成される確率 $\gamma_{m,i}$ は以下のように計算される。

$$\gamma_{m,i} = p(m|\boldsymbol{x}_i) = \frac{w_m \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}{\sum_{m=1}^{M} w_m \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)} \tag{2.17}$$

従って、学習データ $\boldsymbol{X}$ のうち m 番目のガウス分布から生成される確率的サンプル数 N_m と 確率的平均ベクトル $\boldsymbol{e}_m$ はそれぞれ $\gamma_{m,i}$ を用いて以下のように計算される。

$$N_m = \sum_{n=1}^{N} \gamma_{m,n} \tag{2.18}$$

$$\boldsymbol{e}_m = \frac{1}{N_m} \sum_{n=1}^{N} \gamma_{m,n} \boldsymbol{x}_n \tag{2.19}$$

これらを用いて、平均ベクトル $\boldsymbol{\mu}_m$ は以下のように更新される。

$$\hat{\boldsymbol{\mu}}_m = \frac{\tau}{N_m + \tau} \boldsymbol{\mu}_m + \frac{N_m}{N_m + \tau} \boldsymbol{e}_m \tag{2.20}$$

理論的には τ は UBM を推定するときの学習データのうち、m 番目のガウス分布から生成される確率的サンプル数であるが、実際上は固定値を与えることが多い。式 (2.20) から、$\hat{\boldsymbol{\mu}}_m$ は最尤推定により求まるパラメータと事前分布のパラメータの内挿であることが分かる。

このように UBM を事前分布とした MAP 推定を用いて各発話の GMM を推定することで、一発話という少量のサンプルから安定に推定できるだけでなく、発話 GMM の各分布のインデックスと UBM の各分布のインデックスの対応付けが保たれる。これにより、GMM-SV が発話間で比較可能な特徴量となる。

2.4　i-vector

一発話から抽出された GMM-SV $\boldsymbol{M}$ は、因子分析に基づき UBM の GMM-SV $\boldsymbol{m}$ と、発話内容や話者・収録環境の変化による音声のばらつきをモデル化した低次元空間への射影行列 $\boldsymbol{T}$ (Total variability matrix) を用いて

$$\boldsymbol{M} = \boldsymbol{m} + \boldsymbol{T}\boldsymbol{w} \tag{2.21}$$

と分解される。この $\boldsymbol{w}$ が、i-vector と呼ばれる [8]。GMM-SV には発話内容・話者性・収録環境の成分が混在しており、言語識別においては話者性・収録環境の成分、話者識別においては発話内容・収録環境の成分がノイズとなりうる。一方 i-vector では、Total variability matrix による低次元空間への射影によってこれらの成分の除去を行なっている。

Total variability matrix による射影は Principal Component Analysis (PCA) に相当する。話者識別分野における i-vector は、GMM-SV に対する PCA という観点から考えると、次に述べる声質変換分野で提案された Eigenvoice に基づく話者情報表現と基本的に同一視できる。

2.5　Eigenvoice に基づく話者情報表現

2.5.1　概要

Eigenvoice は音声認識におけるモデル適応法として提案された技術であるが [21]、これを GMM に基づく声質変換に適用した固有声変換法 (Eigenvoice Conversion; EVC) が声質変換分野で提案されている [22]。GMM に基づく声質変換では、入力話者の特徴量 $\boldsymbol{X}_t$ と出力話者の特徴量 $\boldsymbol{Y}_t$ の結合 GMM を用いて入出力の変換のモデル化を行なうが、EVC ではこの結合 GMM の出力話者側の平均ベクトルを事前学習用の S 人の話者の特徴を用いて表した EV-GMM $\boldsymbol{\lambda}^{(EV)}$ に基づいて声質変換を行なう。

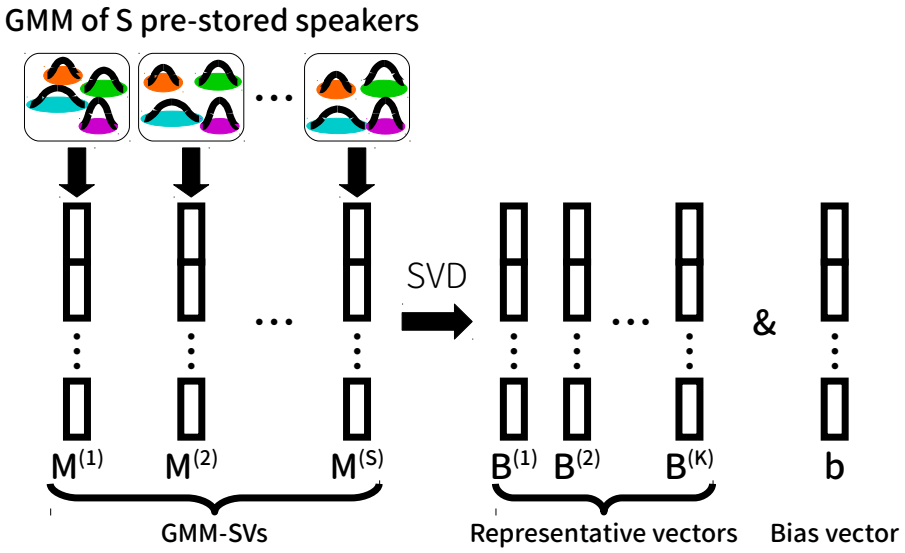

図 2.6: Eigenvoice に基づく話者空間の構築

2.5.2　入力話者と出力話者の結合 GMM

入力話者の特徴量 $\boldsymbol{X}_t$ と出力話者の特徴量 $\boldsymbol{Y}_t$ の結合 GMM は、$\boldsymbol{X}_t$ と $\boldsymbol{Y}_t$ を連結した $[\boldsymbol{X}_t^\top, \boldsymbol{Y}_t^\top]^\top$ を確率変数とする GMM として以下のように表される。

$$p(\boldsymbol{X}_t, \boldsymbol{Y}_t) = \sum_{m=1}^{M} w_m \mathcal{N}(\boldsymbol{X}_t, \boldsymbol{Y}_t; \boldsymbol{\mu}_m^{(X,Y)}, \boldsymbol{\Sigma}_m^{(X,Y)}) \tag{2.22}$$

$$\boldsymbol{\mu}_m^{(X,Y)} = \begin{bmatrix} \boldsymbol{\mu}_m^{(X)} \\ \boldsymbol{\mu}_m^{(Y)} \end{bmatrix}, \boldsymbol{\Sigma}_m^{(X,Y)} = \begin{bmatrix} \boldsymbol{\Sigma}_m^{(XX)} & \boldsymbol{\Sigma}_m^{(XY)} \\ \boldsymbol{\Sigma}_m^{(YX)} & \boldsymbol{\Sigma}_m^{(YY)} \end{bmatrix} \tag{2.23}$$

EVC では、$\boldsymbol{\mu}_m^{(Y)}$ を事前学習用の S 人の話者の特徴を用いて表す。

2.5.3　Eigenvoice に基づく話者空間の構築と話者情報表現

i)　SVD による話者空間の基底の計算

まず、事前学習用話者毎に GMM を学習し、GMM-SV を抽出する。S 個の GMM-SV に対して特異値分解 (Singular Value Decomposition; SVD) を行なうことでバイアスベクトルと $K(\leqq S)$ 個の基底ベクトルを求め、これによって話者空間を構築する（図 2.6 参照）。

SVD は行列 $\boldsymbol{A} \in \mathcal{R}^{M \times N}$ に対する以下のような分解を指す。

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top \tag{2.24}$$

但し、$\boldsymbol{U} \in \mathcal{R}^{M \times M}, \boldsymbol{S} \in \mathcal{R}^{M \times N}, \boldsymbol{V} \in \mathcal{R}^{N \times N}$ で $\boldsymbol{U}, \boldsymbol{V}$ は直交行列、$\boldsymbol{S}$ は K 個の特異値からなる対角行列である。ここで、K は $\boldsymbol{A}$ のランクを表す $(K \leqq \min(M, N))$。本研究では、M は GMM-SV の次元数、N は話者空間の構築に用いるデータ数（発話数）を表す。SVD は実対称行列に対する固有値分解の拡張になっており、任意の実行列に対して適用可能である。PCA によって射影された空間を主部分空間と呼ぶが、$\boldsymbol{A}$ が各行がデータを表す標準化されたデータ行列であ

る場合、$\boldsymbol{V}$ は主部分空間の基底を各列に持つ。すなわち、SVD を用いることで PCA を行なうことができる。

SVD によって求めたバイアスベクトルと K 個の基底ベクトルを用いて、出力話者の GMM-SV $\boldsymbol{M}^{(tar)} = [\boldsymbol{\mu}_1^{(tar)^\top}, \cdots, \boldsymbol{\mu}_M^{(tar)^\top}]^\top$ を以下のようにバイアスベクトルと基底ベクトルの線型結合で表す。

$$\boldsymbol{M}^{(tar)} = \boldsymbol{B}\boldsymbol{w} + \boldsymbol{b} \tag{2.25}$$

但し $\boldsymbol{B}$ は $K(\leqq S)$ 個の基底ベクトルからなる行列である。このように出力話者の GMM-SV が K 次元の重みベクトル $\boldsymbol{w}$ によって制御されるため、$\boldsymbol{w}$ が出力話者を表現していると考えることができる。事前学習用話者の情報を活用しているため、データが少量でも効率的に出力話者を表現することが可能になっている。

ii)　最尤基準に基づく重みベクトルの計算

重みベクトル $\boldsymbol{w}$ は、出力話者の音声データを用いて最尤基準に基づいて推定することができる。出力話者の特徴量系列を $\boldsymbol{Y}^{(tar)}$ とすると、$\boldsymbol{w}$ は以下のように推定できる。

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmax}} \int p(\boldsymbol{X}, \boldsymbol{Y}^{(tar)} | \boldsymbol{\lambda}^{(EV)}, \boldsymbol{w}) \mathrm{d}\boldsymbol{X} \tag{2.26}$$

$$= \underset{\boldsymbol{w}}{\operatorname{argmax}}\, p(\boldsymbol{Y}^{(tar)} | \boldsymbol{\lambda}^{(EV)}, \boldsymbol{w}) \tag{2.27}$$

ここで、出力の確率密度分布は GMM で表されるため、以下の補助関数を導入して EM アルゴリズムによって重みベクトルの最適化を行なう。

$$Q(\boldsymbol{w}, \hat{\boldsymbol{w}}) = \sum_m p(m|\boldsymbol{Y}^{(tar)}, \boldsymbol{\lambda}^{(EV)}, \boldsymbol{w}) \log p(\boldsymbol{Y}^{(tar)}, m | \boldsymbol{\lambda}^{(EV)}, \hat{\boldsymbol{w}}) \tag{2.28}$$

GMM の混合数を M とすると、式 (2.28) より $\hat{\boldsymbol{w}}$ に関する更新式は以下のようになる。

$$\hat{\boldsymbol{w}} = \left[\sum_{m=1}^{M} \overline{\gamma}_m^{(tar)} \boldsymbol{B}_m^\top \boldsymbol{\Sigma}_m^{(YY)^{-1}} \boldsymbol{B}_m \right]^{-1} \left[\sum_{m=1}^{M} \boldsymbol{B}_m^\top \boldsymbol{\Sigma}_m^{(YY)^{-1}} \overline{\boldsymbol{Y}}_m^{(tar)} \right] \tag{2.29}$$

$$\overline{\gamma}_m^{(tar)} = \sum_{t=1}^{T} \gamma_{m,t} \tag{2.30}$$

$$\overline{\boldsymbol{Y}}_m^{(tar)} = \sum_{t=1}^{T} \gamma_{m,t} (\boldsymbol{Y}_t^{(tar)} - \boldsymbol{b}_m^{(0)}) \tag{2.31}$$

$$\gamma_{m,t} = P(m | \boldsymbol{Y}_t^{(tar)}, \boldsymbol{\lambda}^{(EV)}, \boldsymbol{w}) \tag{2.32}$$

但し $\boldsymbol{b}_m^{(0)} \in \mathcal{R}^{D\times 1}$ であり、$\boldsymbol{b} = [\boldsymbol{b}_1^{(0)^\top}, \cdots, \boldsymbol{b}_M^{(0)^\top}]^\top \in \mathcal{R}^{DM\times 1}$ となっている。

iii)　MAP 基準に基づく重みベクトルの計算

前節では最尤基準に基づく重みベクトルの計算について述べたが、MAP 基準に基づいて計算することも可能である。重みベクトルの最尤推定の式 (2.27) を MAP 推定の式に書き直すと以下のようになる。

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmax}}\, p(\boldsymbol{\lambda}^{(EV)}, \boldsymbol{w} | \boldsymbol{Y}^{(tar)}) \tag{2.33}$$

$$= \underset{\boldsymbol{w}}{\operatorname{argmax}}\, p(\boldsymbol{Y}^{(tar)} | \boldsymbol{\lambda}^{(EV)}, \boldsymbol{w}) p(\boldsymbol{w}) \tag{2.34}$$

i-vector と同様に事前分布を $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ と仮定した場合、重みベクトル $\hat{\boldsymbol{w}}_n$ の更新式は以下のようになる。

$$\hat{\boldsymbol{w}} = \left[\left(\sum_{m=1}^{M} \overline{\gamma}_m^{(tar)} \boldsymbol{B}_m^\top \boldsymbol{\Sigma}_m^{(YY)^{-1}} \boldsymbol{B}_m \right) + \boldsymbol{I} \right]^{-1} \left[\sum_{m=1}^{M} \boldsymbol{B}_m^\top \boldsymbol{\Sigma}_m^{(YY)^{-1}} \overline{\boldsymbol{Y}}_m^{(tar)} \right] \tag{2.35}$$

2.6　EVC における話者正規化学習に基づく話者情報表現

2.6.1　EVC における話者正規化学習

EVC の性能をより高める手法として、EVC に対する話者正規化学習 (Speaker Adaptive Training; SAT) が提案されている [23]。SAT は本来、話者適応のための適応元モデルとして標準的な話者モデルを構築する手法として提案されたものである。これは、適応元のモデルとして複数話者間の発話情報を持つ不特定話者モデルよりも、標準的な性質を持つある仮想的な話者モデルを用いることでより適切な話者適応が行なえるという考えに基づいている。[23] では SAT を話者単位で行なっているが、これを発話の発話単位で行なうことも可能である。本節では発話単位の SAT について述べる。

Eigenvoice における発話単位の SAT は、以下のように全ての学習データに対するモデルの尤度を最大化する基準で基底とバイアス、学習データの各発話の重みベクトルを推定する。

$$\hat{\boldsymbol{\lambda}}(\hat{\boldsymbol{w}}_1^N) = \underset{\boldsymbol{\lambda}, \boldsymbol{w}_1^N}{\operatorname{argmax}} \prod_{n=1}^{N} \prod_{t_n=1}^{Tn} p(\boldsymbol{Y}_{t_n}^{(n)} | \boldsymbol{\lambda}(\boldsymbol{w}_n)) \tag{2.36}$$

ここで、$\boldsymbol{\lambda}(\boldsymbol{w}_n)$ は重みベクトル $\boldsymbol{w}_n$ で表される n 番目の学習データ（発話）に適応した GMM である。$\boldsymbol{w}_1^N$ は全ての学習データの重みベクトル $(\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_N)$ のセットである。式 (2.36) を計算するために以下の補助関数を導入する。

$$Q(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}}) = \sum_{n=1}^{N} \sum_{m=1}^{M} \overline{\gamma}_m^{(n)} \log p \left(\boldsymbol{Y}^{(n)}, m | \hat{\boldsymbol{\lambda}}(\hat{\boldsymbol{w}}_n) \right) \tag{2.37}$$

$$\gamma_{m,t_n}^{(n)} = p\Big(m | \boldsymbol{Y}_{t_n}^{(n)}, \boldsymbol{\lambda}(\boldsymbol{w}_n) \Big), \overline{\gamma}_m^{(n)} = \sum_{t_n=1}^{T_n} \gamma_{m,t_n}^{(n)} \tag{2.38}$$

式 (2.37) に基づいて全てのパラメータを一度に更新するのは困難であるため、以下のように逐次的にパラメータを更新する。

1. 現在の基底・バイアスと式 (2.38) を用いて、$\gamma_{m,t_n}^{(n)}$ 及び $\overline{\gamma}_m^{(n)}$ を求める。
2. $\gamma_{m,t_n}^{(n)}$ 及び $\overline{\gamma}_m^{(n)}$ と現在の基底・バイアスを用いて、学習データの各発話の重み行列 $\hat{\boldsymbol{w}}_n$ を更新する。
3. 更新した $\hat{\boldsymbol{w}}_n$ を用いて、基底 $\hat{\boldsymbol{B}}_1, \cdots, \hat{\boldsymbol{B}}_K$ とバイアス $\hat{\boldsymbol{b}}$ を更新する。
4. 上記 1.〜3. の手順を一定の回数繰り返す。

表 2.1: EVC における SAT と i-vector の比較

	i-vector	EVC における SAT
推定基準	MAP	ML
基底の初期値	ランダム	SVD で求めた基底
バイアス	UBM 平均	基底と同時に毎回更新
事後確率計算	UBM	各発話に適応した GMM

手順 2. における重みベクトル $\hat{\boldsymbol{w}}_n$ の更新式は以下のようになる。

$$\hat{\boldsymbol{w}}_n = \left(\sum_{m=1}^{M} \overline{\gamma}_m^{(n)} \boldsymbol{B}_m^\top \boldsymbol{\Sigma}_m^{-1} \boldsymbol{B}_m \right)^{-1} \left[\sum_{m=1}^{M} \boldsymbol{B}_m^\top \boldsymbol{\Sigma}_m^{-1} (\overline{\boldsymbol{Y}}_m^{(n)} - \overline{\gamma}_m^{(n)} \boldsymbol{b}_m) \right]$$
$$\overline{\boldsymbol{Y}}_m^{(n)} = \sum_{t_n=1}^{T_n} \gamma_{m,t_n} \boldsymbol{Y}_{t_n}^{(n)} \tag{2.39}$$

次に、手順 3. における基底 $\hat{\boldsymbol{B}}_1, \cdots, \hat{\boldsymbol{B}}_K$ とバイアス $\hat{\boldsymbol{b}}$ の更新式は以下のようになる。

$$\hat{\boldsymbol{v}}_m = \left(\sum_{n=1}^{N} \overline{\gamma}_m^{(n)} \boldsymbol{W}_n^\top \boldsymbol{\Sigma}_m^{-1} \boldsymbol{W}_n \right)^{-1} \left(\sum_{n=1}^{N} \boldsymbol{W}_n^\top \boldsymbol{\Sigma}_m^{-1} \overline{\boldsymbol{Y}}_m^{(n)} \right) \tag{2.40}$$

$$\hat{\boldsymbol{v}}_m = \left[\hat{\boldsymbol{b}}_m^\top \quad \hat{\boldsymbol{B}}_m^{(1)} \quad \cdots \quad \hat{\boldsymbol{B}}_m^{(K)} \right]^\top \in \mathcal{R}^{(K+1)D \times 1} \tag{2.41}$$

$$\hat{\boldsymbol{W}}_n = \left[1 \quad \hat{\boldsymbol{w}}_n^\top \right] \otimes \boldsymbol{I} \quad \in \mathcal{R}^{D \times (K+1)D} \tag{2.42}$$

2.6.2　Eigenvoice に基づく話者情報表現と SAT, i-vector の関係

2.4, 2.5 節で i-vector と Eigenvoice に基づく話者情報表現が GMM-SV に対する PCA と解釈できると述べたが、厳密にはこれらは異なる方法で実装された PCA となる。Eigenvoice に基づく話者情報表現は SVD を用いた通常の PCA （以下 Deterministic PCA (DPCA) と呼ぶ) であるが、i-vector は DPCA を確率的な生成モデルに拡張した Probabilistic PCA (PPCA) である。DPCA ではデータの分散最大化を基準とし、直交基底によって特徴量空間を構築するため、基底を縮約することにより次元圧縮が可能となる。これに対し PPCA では基底数を事前に設定し、その条件のもとでデータにフィットするように基底を計算し、特徴量空間を構築する。そのため、PPCA によって計算される基底は斜交基底となる。

EVC における SAT も i-vector と同様に PPCA と解釈でき、これを話者単位ではなく発話単位で行なうことで i-vector とほぼ等価になる。発話単位とは「全ての発話が異なる話者によって発声された」とすることであり、話者空間の構築の際に学習データの話者ラベルを用いない教師なしの枠組みとなる。i-vector と EVC における SAT のこの他の違いを表 2.1 に示す。i-vector は事前分布として $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ を仮定し、MAP 基準で推定される。

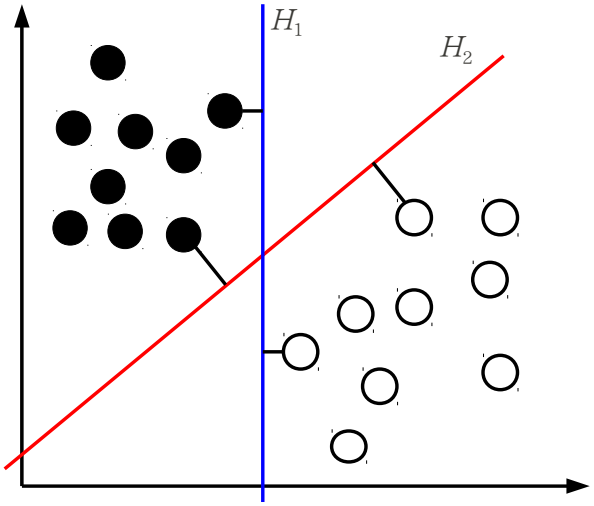

図 2.7: 2 次元データの 2 クラス分類

2.7　Support Vector Machine (SVM)

2.7.1　SVM を用いた 2 クラス分類

GMM-SV を用いた言語識別・話者識別では、識別器として SVM が用いられている [9]。SVM は、以下のように超平面 $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + \boldsymbol{b}$ を境界として入力特徴ベクトル $\boldsymbol{x}$ を 2 つのクラスに分類する線形識別モデルである。

$$y = \boldsymbol{w}^\top \boldsymbol{x} + \boldsymbol{b} = \begin{cases} 1 & (f(\boldsymbol{x}) \geq 0) \\ -1 & (f(\boldsymbol{x}) < 0) \end{cases} \tag{2.43}$$

SVM を学習する際は、学習データからのマージンを最大化する超平面を求める。すなわち、最も近くにあるクラス 1 の学習データからの距離と最も近くにあるクラス 2 のデータからの距離が最大になるように超平面を求める。例えば図 2.7 の場合では、直線 H_1 と直線 H_2 はいずれも 2 つのクラスを分離できているが、SVM によって求まるマージン最大の境界は直線 H_2 となる。

2.7.2　SVM を用いた多クラス分類

SVM は基本的には 2 クラス分類を行なう識別器であるが、複数の SVM を組合せることで多クラス分類も可能となる。複数の SVM の組合せの方法として、one-versus-one 法と one-versus-rest 法の 2 つがある。前者はクラス i と j の分類を行なう SVM を全ての i, j の組合せについて作り、多数決によって最終的なクラスの決定を行なうという方法である。後者はクラス i とそれ以外の分類を行なう SVM を全ての i について作り、識別面から最も遠いクラスを推定結果とする方法である。

2.8 Probabilistic Linear Discriminant Analysis (PLDA)

2.8.1 概要

Linear Discriminant Analysis (LDA) はクラス間分散とクラス内分散に基づきクラスが最も分離するように低次元への射影を行なう手法であるが、これを確率的な生成モデルに拡張した Probabilistic LDA (PLDA) という手法が顔画像認識の分野で提案された [24]。話者識別分野においても i-vector の識別に PLDA が導入され [11, 12] 、現在では話者識別・言語識別分野における標準的な識別手法の一つとなっている。本節では、[11, 12] で検討されている i-vector の識別手法としての PLDA (Gaussian PLDA) について述べる。

2.8.2 モデル

クラス i の発話のうち j 番目の発話を表す i-vector を $\boldsymbol{x}_{ij}$ $(j = 1, \cdots, N_i)$ としたとき、$\boldsymbol{x}_{ij}$ について以下のような分解を仮定する。

$$\boldsymbol{x}_{ij} = \boldsymbol{\mu} + \boldsymbol{F}\boldsymbol{h}_i + \boldsymbol{\epsilon}_{ij} \tag{2.44}$$

但し、$\boldsymbol{\mu}$ はバイアス、$\boldsymbol{F}$ は基底、$\boldsymbol{h}_i$ は $\mathcal{N}(0, \boldsymbol{I})$ に従う潜在変数、$\boldsymbol{\epsilon}_{ij}$ は $\mathcal{N}(0, \boldsymbol{\Sigma})$ に従う残差を表す（$\boldsymbol{\Sigma}$ は全共分散行列）。バイアス $\boldsymbol{\mu}$、基底 $\boldsymbol{F}$、残差の分散共分散行列 $\boldsymbol{\Sigma}$ はクラス非依存、潜在変数 $\boldsymbol{h}_i$、残差 $\boldsymbol{\epsilon}_{ij}$ はクラス依存のパラメータである。式 (2.44) において、$\boldsymbol{\mu} + \boldsymbol{F}\boldsymbol{h}_i$ がクラス間の変動（発話非依存）、残差 $\boldsymbol{\epsilon}_{ij}$ がクラス内の変動（発話依存）を記述している。

2.8.3 パラメータの学習

パラメータ $\theta = \{\boldsymbol{F}, \boldsymbol{\Sigma}\}$ は EM アルゴリズムを用いて最尤基準で推定される。このとき、バイアス $\boldsymbol{\mu}$ は全データ平均となり、E-Step と M-Step の更新式は以下のようになる。

E-Step:

$$\boldsymbol{E}[\boldsymbol{h}_i] = (N_i \boldsymbol{F}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{F} + \boldsymbol{I})^{-1} \left[\sum_{j=1}^{N_i} \boldsymbol{F}^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}_{ij} - \boldsymbol{\mu}) \right] \tag{2.45}$$

$$\boldsymbol{E}[\boldsymbol{h}_i \boldsymbol{h}_i^\top] = (N_i \boldsymbol{F}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{F} + \boldsymbol{I})^{-1} + \boldsymbol{E}[\boldsymbol{h}_i] \boldsymbol{E}[\boldsymbol{h}_i]^\top \tag{2.46}$$

M-Step:

$$\boldsymbol{F} = \left[\sum_{i,j} (\boldsymbol{x}_{ij} - \boldsymbol{\mu}) \boldsymbol{E}[\boldsymbol{h}_i]^\top \right] \left[\sum_{i=1}^{L} N_i \boldsymbol{E}[\boldsymbol{h}_i \boldsymbol{h}_i^\top] \right]^{-1} \tag{2.47}$$

$$\boldsymbol{\Sigma} = \frac{1}{N} \sum_{i,j} \left[(\boldsymbol{x}_{ij} - \boldsymbol{\mu})(\boldsymbol{x}_{ij} - \boldsymbol{\mu})^\top - \boldsymbol{F} \boldsymbol{E}[\boldsymbol{h}_i] (\boldsymbol{x}_{ij} - \boldsymbol{\mu})^\top \right] \tag{2.48}$$

但し、L は言語数、N はパラメータの学習に用いた全発話数 $(= \sum_{i=1}^{L} N_i)$ を表す。パラメータ更新式自体は因子分析と似通っているが、PLDA は潜在変数 $\boldsymbol{h}_i$ を同一クラス内で共有している点と、$\boldsymbol{\Sigma}$ が全共分散行列となっている点で因子分析とは異なっている。

2.8.4　識別

2 つの i-vector $\boldsymbol{x}_1, \boldsymbol{x}_2$ が同一のクラスか否かを以下の対数尤度比に基づくスコアを用いて判定する。

$$score = \log \frac{p(\boldsymbol{x}_1, \boldsymbol{x}_2|\mathcal{H}_s)}{p(\boldsymbol{x}_1|\mathcal{H}_d)p(\boldsymbol{x}_2|\mathcal{H}_d)} \tag{2.49}$$

但し、$\mathcal{H}_s$ は「両者が同一のクラス」という仮説、$\mathcal{H}_d$ は「両者が異なるクラス」という仮説を表す。識別の際は、各クラスの代表値としてクラス平均 i-vector を用意しておき、それらとテスト i-vector のマッチングスコアを求める。

式 (2.49) は以下のように計算される。

$$score = \log \mathcal{N}\left(\begin{bmatrix}\boldsymbol{x}_1 \\ \boldsymbol{x}_2\end{bmatrix}; \begin{bmatrix}\boldsymbol{\mu} \\ \boldsymbol{\mu}\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_{tot} & \boldsymbol{\Sigma}_{ac} \\ \boldsymbol{\Sigma}_{ac} & \boldsymbol{\Sigma}_{tot}\end{bmatrix}\right) - \log \mathcal{N}\left(\begin{bmatrix}\boldsymbol{x}_1 \\ \boldsymbol{x}_2\end{bmatrix}; \begin{bmatrix}\boldsymbol{\mu} \\ \boldsymbol{\mu}\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_{tot} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma}_{tot}\end{bmatrix}\right) \tag{2.50}$$

$$= \boldsymbol{x}_1^\top \boldsymbol{Q} \boldsymbol{x}_1 + \boldsymbol{x}_2^\top \boldsymbol{Q} \boldsymbol{x}_2 + 2\boldsymbol{x}_1^\top \boldsymbol{P} \boldsymbol{x}_2 + const \tag{2.51}$$

但し、

$$\boldsymbol{\Sigma}_{tot} = \boldsymbol{F}\boldsymbol{F}^\top + \boldsymbol{\Sigma} \tag{2.52}$$

$$\boldsymbol{\Sigma}_{ac} = \boldsymbol{F}\boldsymbol{F}^\top \tag{2.53}$$

$$\boldsymbol{Q} = \boldsymbol{\Sigma}_{tot}^{-1} - (\boldsymbol{\Sigma}_{tot} - \boldsymbol{\Sigma}_{ac}\boldsymbol{\Sigma}_{tot}^{-1}\boldsymbol{\Sigma}_{ac})^{-1} \tag{2.54}$$

$$\boldsymbol{P} = \boldsymbol{\Sigma}_{tot}^{-1}\boldsymbol{\Sigma}_{ac}(\boldsymbol{\Sigma}_{tot} - \boldsymbol{\Sigma}_{ac}\boldsymbol{\Sigma}_{tot}^{-1}\boldsymbol{\Sigma}_{ac})^{-1} \tag{2.55}$$

である。

第3章

テンソル分解に基づく話者・言語情報表現と Matrix Variate PLDA を用いた識別

3.1　はじめに

前章では、従来の言語･話者情報表現と識別の手法について説明した。従来手法では発話を GMM によってモデル化し、各分布の平均ベクトルを一列に連結した GMM-SV やそれに対して PCA に相当する次元圧縮を施した特徴量によって言語・話者を表現していた。しかし、複数要因からの音響的な変動を捉えた発話 GMM を GMM-SV というベクトルによる表現に落としこむことでそれらが混在してしまう点が問題となる。そこで、本研究では発話 GMM の各分布の平均ベクトルを行列で表現し、複数発話によって得られる複数行列をテンソルとみなし、テンソル分解に基づく話者・言語情報表現について検討する。また、テンソル分解に基づく手法では特徴量が行列で表現されるため、行列の各行と各列の関係性を考慮することでより適切な識別が可能になることが期待される。そこで、本研究では、PLDA を行列変量に拡張した Matrix Variate PLDA (MV-PLDA) による識別についても検討する。本章では、GMM-SV 表現に内在する問題について述べたのち、テンソル分解に基づく話者・言語情報表現と MV-PLDA を用いた識別について説明する。

3.2　GMM-SV 表現に内在する問題

GMM-SV は GMM の各分布の平均ベクトルが単純に一列に並んだ構成になっており、各分布のインデックスと平均ベクトルの各次元の二つが混在した表現となるため、GMM から GMM-SV を抽出する過程で分布同士の関係性の情報を失う。2.3 節において、GMM を構成する各ガウス分布が凡そ音素や単音のような音響的な特徴を捉えていることが期待できると述べたが、音素や単音には類似したものも大きく異なるものもあり、GMM を構成するガウス分布の中にも相関の高いもの、低いものが存在すると考えられる。従って、それらの関係性を適切に考慮することでより精密な言語・話者情報表現が可能になることが期待できる。

3.3　テンソル分解に基づく話者・言語情報表現

3.3.1　概要

前節で述べた GMM-SV 表現に内在する問題を解決する手法として、テンソル分解に基づく話者情報表現を用いた話者識別 [10] と声質変換 [25] が検討されている。[10, 25] では、GMM の各分布の平均ベクトル群を GMM-SV ではなく行および列がそれぞれ GMM の各分布と平均ベクトルに対応するような行列によって表し、複数名分の行列を重ねて一つのテンソルとして表現する。このテンソルに対して、PCA の射影行列の一つの計算法である SVD を拡張した Tucker 分解と呼ばれるテンソル分解を用いて話者情報を表現している。本研究では、この手法を言語情報表現にも適用する。

3.3.2　多重線形解析

本節では本研究に関連する多重線形解析について述べる。テンソルは、行列表現を一般化した多次元配列表現である。テンソルにおける個々のインデックスはモードと呼ばれ、特定のモードに沿ってスライスする平坦化操作によってテンソルを行列の形で表現できる。$\mathcal{A} \in \mathcal{R}^{I_1 \times I_2 \times I_3}$ を 3 階のテンソルとすると、これをそれぞれモード 1, 2, 3 に沿って平坦化した行列 $\mathcal{A}_{(1)}, \mathcal{A}_{(2)}, \mathcal{A}_{(3)}$

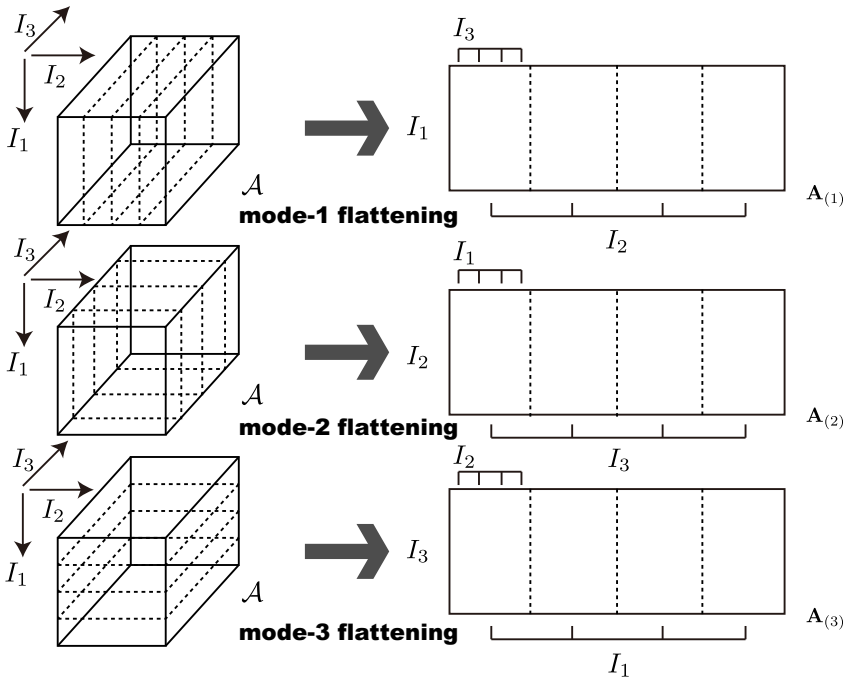

図 3.1: $(I_1 \times I_2 \times I_3)$-テンソル $\mathcal{A}$ の 平坦化行列 $\mathcal{A}_{(1)}, \mathcal{A}_{(2)}, \mathcal{A}_{(3)}$ [25]

は図 3.1 のようになる。このような平坦化行列を用いて、テンソルと行列間の積が定義できる。テンソル $\mathcal{G} \in \mathcal{R}^{I_1 \times \cdots \times I_N}$ と行列 $\boldsymbol{B} \in \mathcal{R}^{J_n \times I_n}$ のモード n 積 $\mathcal{A} = \mathcal{G} \times_n \boldsymbol{B}$ はモード n の平坦化行列による演算 $\boldsymbol{A}_{(n)} = \boldsymbol{B} \cdot \boldsymbol{G}_{(n)}$ によって定義される。

3.3.3　Tucker 分解

Eigenvoice に基づく手法では SVD を用いて話者空間の構築を行なっていたが、行列 $\boldsymbol{A}$ に対する SVD を 2 階のテンソルの分解として書き直すと以下のようになる。

$$\boldsymbol{A} = \boldsymbol{S} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \tag{3.1}$$

これを以下のように高階のテンソルに拡張したものを Tucker 分解と呼ぶ（図 3.2 参照）[26]。

$$\mathcal{A} = \mathcal{S} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3 \tag{3.2}$$

但し、$\boldsymbol{U}_1, \boldsymbol{U}_2, \boldsymbol{U}_3$ が直交行列の場合はコアテンソル $\mathcal{S}$ は密なテンソルとなる。

Tucker 分解は、各モードの平坦化行列に対して SVD を行なうことで計算できる。各モードの平坦化行列を $\mathcal{A}_{(1)}, \mathcal{A}_{(2)}, \mathcal{A}_{(3)}$ とし、これらに対して SVD を行なうと以下のように $\boldsymbol{U}_1, \boldsymbol{U}_2, \boldsymbol{U}_3$ が求まる。

$$\mathcal{A}_{(1)} = \boldsymbol{U}_1 \boldsymbol{S}_1 \boldsymbol{V}_1^\top \tag{3.3}$$
$$\mathcal{A}_{(2)} = \boldsymbol{U}_2 \boldsymbol{S}_2 \boldsymbol{V}_2^\top \tag{3.4}$$
$$\mathcal{A}_{(3)} = \boldsymbol{U}_3 \boldsymbol{S}_3 \boldsymbol{V}_3^\top \tag{3.5}$$

このように、各モードの平坦化行列の左特異行列として式 (3.2) の基底が求まる。これらを用いて、以下のようにコアテンソル $\mathcal{S}$ を導出する。

$$\mathcal{S} = \mathcal{A} \times_1 \boldsymbol{U}_1^\top \times_2 \boldsymbol{U}_2^\top \times_3 \boldsymbol{U}_3^\top \tag{3.6}$$

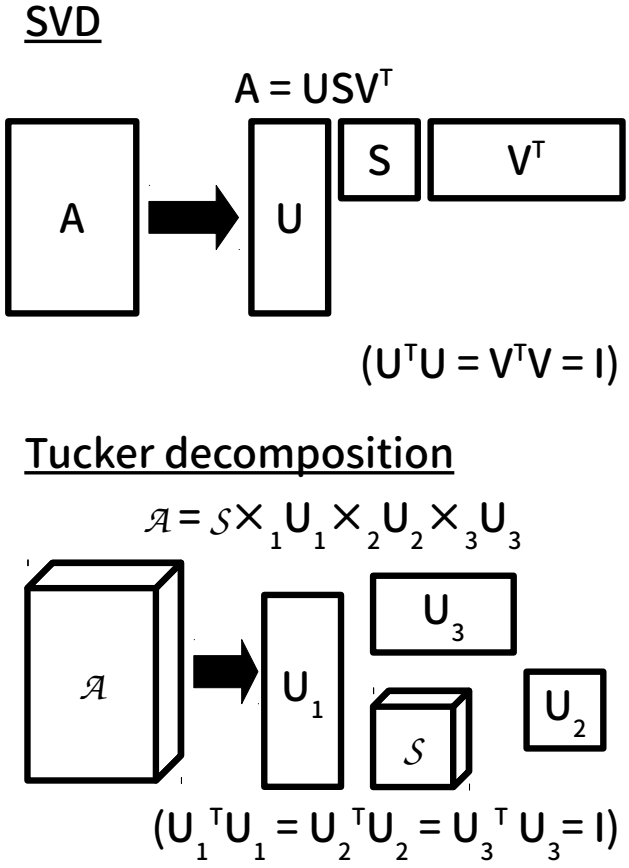

図 3.2: SVD と Tucker 分解の比較

3.3.4　Tucker 分解による言語・話者情報表現

i)　Tucker 分解による言語・話者空間の基底の計算

学習データの各発話について、一発話の GMM の各分布の平均ベクトルを並べて「混合数 $(M)\times$ 平均ベクトルの次元数 (D)」の行列で表現する。始めに全発話の平均を求め、これをバイアス行列 $\boldsymbol{b}$ として各発話の行列から減算しておく。学習データの発話数を N としたとき、N 個の $M \times D$ 行列を 3 階のテンソル $\mathcal{M} \in \mathcal{R}^{M\times D\times N}$ として扱い、Tucker 分解を行なうと以下のようになる（図 3.3 参照）。

$$\mathcal{M} = \mathcal{G}^{M\times D\times N} \times_1 \boldsymbol{U}^{(M)} \times_2 \boldsymbol{U}^{(D)} \times_3 \boldsymbol{U}^{(N)} \tag{3.7}$$

但し $\boldsymbol{U}^{(M)} \in \mathcal{R}^{M\times M}, \boldsymbol{U}^{(D)} \in \mathcal{R}^{D\times D}, \boldsymbol{U}^{(N)} \in \mathcal{R}^{N\times N}$ であり、それぞれ GMM 混合数、平均ベクトルの次元、発話インデックスの効果を捉えている。ここで、以下のように $\mathcal{M}$ の第 3 モードを固定することで、特定の発話を表す行列が得られると考えられる。

$$\hat{\boldsymbol{\mu}}^{(n)} = \mathcal{G} \times_1 \boldsymbol{U}^{(M)} \times_2 \boldsymbol{U}^{(D)} \times_3 \boldsymbol{U}^{(N)}(n,:) \tag{3.8}$$

$\boldsymbol{U}^{(N)}(n,:) \in \mathcal{R}^{1\times N}$ を重みパラメータ、その他のモード積の項を言語空間の基底と捉えると SVD と等価になる。一方、本稿では GMM の各分布の関係性を考慮するため以下のようなグルーピン

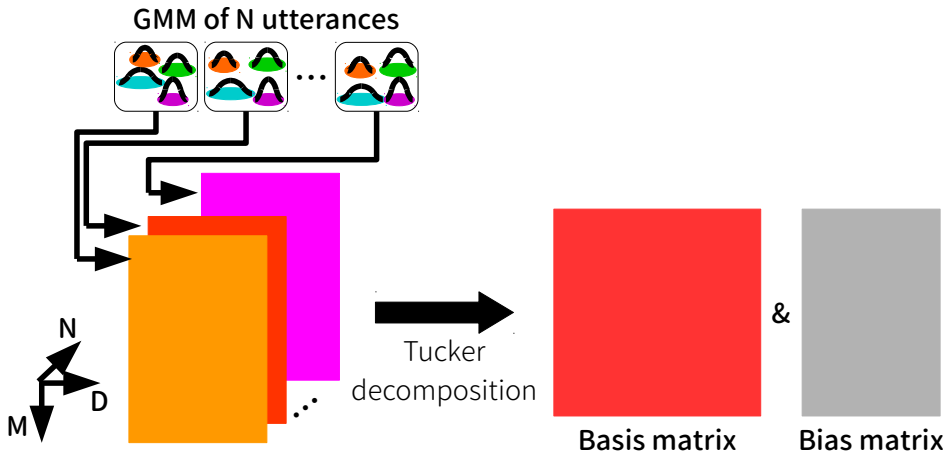

図 3.3: テンソル分解に基づく言語・話者空間構築

グを考える。

$$\hat{\boldsymbol{\mu}}^{(n)} = \boldsymbol{U}^{(M)} \left\{ \mathcal{G} \times_2 \boldsymbol{U}^{(D)} \times_3 \boldsymbol{U}^{(N)}(n,:) \right\} \tag{3.9}$$

$$= \boldsymbol{U}^{(M)} \boldsymbol{W}_n^{\top} \tag{3.10}$$

$\boldsymbol{U}^{(M)}$ を基底、$\boldsymbol{W}_n \in \mathcal{R}^{D \times M}$ を重み行列とする。次元圧縮の観点から、基底を縮約することで任意の発話は以下のような行列で表される。

$$\boldsymbol{\mu}^{(new)} = \boldsymbol{U}^{(M)} \boldsymbol{W}_{(new)}^{\top} + \boldsymbol{b} \tag{3.11}$$

$\boldsymbol{U}^{(M)} \in \mathcal{R}^{M \times K_M} (K_M \leqq N)$ が表現行列、$\boldsymbol{W}_{(new)} \in \mathcal{R}^{D \times K_M}$ が重み行列となり、$D \times K_M$ の重み行列によって発話の言語情報を表現する。$\boldsymbol{U}^{(M)}$ は GMM の各分布の関係性を記述していると考えられ、重み行列 $\boldsymbol{W}$ を推定することで M 混合の GMM を効率的に推定していると解釈できる。

ii)　最尤基準に基づく重み行列の計算

[10] では式 (3.11) の最小二乗解として重み行列 $\boldsymbol{W}$ を算出しているが、これも EVC の重みベクトルと同様に、以下のような最尤基準に基づいて $\boldsymbol{W}$ の推定を行なうことができる [25]。

$$\hat{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\operatorname{argmax}} \int p(\boldsymbol{X}, \boldsymbol{Y}^{(tar)} | \boldsymbol{\lambda}^{(EV)}, \boldsymbol{W}) \mathrm{d}\boldsymbol{X} \tag{3.12}$$

$$= \underset{\boldsymbol{W}}{\operatorname{argmax}}\, p(\boldsymbol{Y}^{(tar)} | \boldsymbol{\lambda}^{(EV)}, \boldsymbol{W}) \tag{3.13}$$

上記の確率密度分布も GMM で表されるため、EVC における重みベクトルの推定と同様に、式 (2.28) と同形の $\boldsymbol{W}$ に関する補助関数を導入して以下のように重み行列を繰り返し更新する。

$$\mathrm{vec}(\hat{\boldsymbol{W}}) = \left[\sum_{m=1}^{M} \overline{\gamma}_m^{(tar)} \boldsymbol{U}_m^\top \boldsymbol{U}_m \otimes \boldsymbol{\Sigma}_m^{(YY)^{-1}}\right]^{-1} \mathrm{vec}(\boldsymbol{C}) \tag{3.14}$$

$$\boldsymbol{C} = \sum_{m=1}^{M} \boldsymbol{\Sigma}_m^{(YY)^{-1}} (\overline{\boldsymbol{Y}}_m^{(tar)} - \overline{\gamma}_m^{(tar)} \boldsymbol{b}_m^{(0)}) \boldsymbol{U}_m \tag{3.15}$$

$$\boldsymbol{U}_m = \boldsymbol{U}^{(M)}(m,:) \in \mathcal{R}^{1\times K_M} \tag{3.16}$$

$$\boldsymbol{b}_m^{(0)} = \boldsymbol{b}(m,:)^\top \in \mathcal{R}^{D\times 1} \tag{3.17}$$

$$\overline{\gamma}_m^{(tar)} = \sum_{t=1}^{T} \gamma_{m,t} \tag{3.18}$$

$$\overline{\boldsymbol{Y}}_m^{(tar)} = \sum_{t=1}^{T} \gamma_{m,t} (\boldsymbol{Y}_t^{(tar)} - \boldsymbol{b}_m^{(0)}) \tag{3.19}$$

$$\gamma_{m,t} = P(m|\boldsymbol{Y}_t^{(tar)}, \boldsymbol{\lambda}^{(EV)}, \boldsymbol{w}) \tag{3.20}$$

ここで、vec() は行列を列ベクトルに展開する演算子である。

iii)　MAP 基準に基づく重み行列の計算

EVC の重みベクトルと同様に、以下のような MAP 基準に基づいて $\boldsymbol{W}$ の推定を行なうことができる。

$$\hat{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\mathrm{argmax}}\, p(\boldsymbol{\lambda}^{(EV)}, \boldsymbol{W}|\boldsymbol{Y}^{(tar)}) \tag{3.21}$$

$$= \underset{\boldsymbol{W}}{\mathrm{argmax}}\, p(\boldsymbol{Y}^{(tar)}|\boldsymbol{\lambda}^{(EV)}, \boldsymbol{W}) p(\boldsymbol{W}) \tag{3.22}$$

事前分布を $p(\mathrm{vec}(\boldsymbol{W}_n)) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ と仮定した場合、重み行列 $\hat{\boldsymbol{W}}_n$ の更新式は以下のようになる。

$$\mathrm{vec}(\hat{\boldsymbol{W}}) = \left[\left(\sum_{m=1}^{M} \overline{\gamma}_m^{(tar)} \boldsymbol{U}_m^\top \boldsymbol{U}_m \otimes \boldsymbol{\Sigma}_m^{(YY)^{-1}}\right) + \boldsymbol{I}\right]^{-1} \mathrm{vec}(\boldsymbol{C}) \tag{3.23}$$

$$\boldsymbol{C} = \sum_{m=1}^{M} \boldsymbol{\Sigma}_m^{(YY)^{-1}} \left[\overline{\boldsymbol{Y}}_t^{(tar)} - (\boldsymbol{W}_n \boldsymbol{U}_m^\top + \boldsymbol{b}_m)\right] \tag{3.24}$$

3.4　テンソル分解に基づく言語・話者情報表現への SAT の導入

テンソル分解に基づく声質変換 [25] に関しても SAT の導入が検討されており、それによる変換精度の向上が示されている [27]。これも Eigenvoice における発話単位の SAT と同様に、以下のように全ての学習データに対するモデルの尤度を最大化する基準で基底とバイアス、学習データの各発話の重み行列を推定する。

$$\hat{\boldsymbol{\lambda}}(\hat{\mathcal{W}}_1^N) = \underset{\boldsymbol{\lambda}, \boldsymbol{\mathcal{W}}_1^N}{\mathrm{argmax}} \prod_{n=1}^{N} \prod_{t_n=1}^{Tn} p(\boldsymbol{Y}_{t_n}^{(n)}|\boldsymbol{\lambda}(\boldsymbol{W}_n)) \tag{3.25}$$

ここで、$\boldsymbol{\lambda}(\boldsymbol{W}_n)$ は重み行列 $\boldsymbol{W}_n$ で表される n 番目の学習データ（発話）に適応した GMM である。$\mathcal{W}_1^N$ は全ての学習データの重み行列 $(\boldsymbol{W}_1, \boldsymbol{W}_2, \cdots, \boldsymbol{W}_N)$ を集めたテンソル表現である。式 (3.25) を計算するために以下の補助関数を導入する。

$$Q(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}}) = \sum_{n=1}^{N} \sum_{m=1}^{M} \overline{\gamma}_m^{(n)} \log p\left(\boldsymbol{Y}^{(n)}, m | \hat{\boldsymbol{\lambda}}(\hat{\boldsymbol{W}}_n)\right) \tag{3.26}$$

$$\gamma_{m,t_n}^{(n)} = p\left(m | \boldsymbol{Y}_{t_n}^{(n)}, \boldsymbol{\lambda}(\boldsymbol{W}_n)\right) \tag{3.27}$$

$$\overline{\gamma}_m^{(n)} = \sum_{t_n=1}^{T_n} \gamma_{m,t_n}^{(n)} \tag{3.28}$$

式 (3.26) に基づいて全てのパラメータを一度に更新するのは困難であるため、Eigenvoice における発話単位の SAT と同様の手順で逐次的にパラメータを更新する。

重み行列 $\hat{\boldsymbol{W}}_n$ の更新式は以下のようになる。

$$\mathrm{vec}(\hat{\boldsymbol{W}}_n) = \left[\sum_{m=1}^{M} \overline{\gamma}_m^{(n)} \boldsymbol{U}_m^\top \boldsymbol{U}_m \otimes \boldsymbol{\Sigma}_m^{-1}\right]^{-1} \mathrm{vec}(\boldsymbol{C}) \tag{3.29}$$

$$\boldsymbol{C} = \sum_{m=1}^{M} \boldsymbol{\Sigma}_m^{-1} \left[\overline{\boldsymbol{Y}}_{t_n}^{(n)} - (\boldsymbol{W}_n \boldsymbol{U}_m^\top + \boldsymbol{b}_m)\right] \tag{3.30}$$

基底 $\hat{\boldsymbol{U}}^{(M)}$ とバイアス $\hat{\boldsymbol{b}}$ の更新式は以下のようになる。

$$\hat{\boldsymbol{u}}_m = \left(\sum_{n=1}^{N} \overline{\gamma}_m^{(n)} \hat{\boldsymbol{E}}_n^\top \boldsymbol{\Sigma}_m^{-1} \hat{\boldsymbol{E}}_n\right)^{-1} \left(\sum_{n=1}^{N} \hat{\boldsymbol{E}}_n^\top \boldsymbol{\Sigma}_m^{-1} \overline{\boldsymbol{Y}}_m^{(n)}\right) \tag{3.31}$$

$$\hat{\boldsymbol{u}}_m = \left[\hat{\boldsymbol{b}}_m^\top \quad \hat{\boldsymbol{U}}_m\right]^\top \in \mathcal{R}^{(D+K_M)\times 1} \tag{3.32}$$

$$\hat{\boldsymbol{E}}_n = \left[\boldsymbol{I} \quad \hat{\boldsymbol{W}}_n\right] \in \mathcal{R}^{D\times(D+K_M)} \tag{3.33}$$

3.5　テンソル分解に基づく言語・話者情報表現における Bilinear 基底の検討

前節で述べたように、[10, 25, 27] では式 (3.8) における $U^{(M)}$ のみを基底として用いており、これにより GMM の各分布の関係性を考慮した言語・話者空間の構築を行なっている。一方、平均ベクトルの各次元についても相関の高いもの、低いものが存在すると考えられるため、それらの関係性についても同時に考慮することができれば、より効率的に言語・話者情報を表現できることが期待される。そこで、本研究では $U^{(M)}$ と平均ベクトルの各次元の関係性を捉えた $U^{(D)}$ の二つを基底として扱う Bilinear 基底についても新たに検討する。すなわち、式 (3.8) について以下のようなグルーピングを考える。

$$\hat{\boldsymbol{\mu}}^{(n)} = \boldsymbol{U}^{(M)} \left\{\mathcal{G} \times_3 \boldsymbol{U}^{(N)}(n,:)\right\} \boldsymbol{U}^{(D)^\top} \tag{3.34}$$

$$= \boldsymbol{U}^{(M)} \boldsymbol{W}_n'^\top \boldsymbol{U}^{(D)^\top} \tag{3.35}$$

基底を縮約することで任意の発話は以下のような行列で表される。

$$\boldsymbol{\mu}^{(new)} = \boldsymbol{U}^{(M)} \boldsymbol{W}_{(new)}'^\top \boldsymbol{U}^{(D)^\top} + \boldsymbol{b} \tag{3.36}$$

このとき、$\boldsymbol{U}^{(M)} \in \mathcal{R}^{M\times K_M}, \boldsymbol{U}^{(D)} \in \mathcal{R}^{D\times K_D}$ が基底、$\boldsymbol{W}'_{(new)} \in \mathcal{R}^{K_D\times K_M}$ が重み行列となる ($K_M \leqq M, K_D \leqq D$)。混合数 M の方向に加えて次元 D の方向の圧縮も行なうことができるため、重み行列をよりコンパクトに表現できるようになる。i-vector や Eigenvoice に基づく話者情報表現では GMM-SV の各次元の関係性を同等に考慮して話者空間の構築を行なうが、Bilinear 基底ではこれを GMM の各分布の関係性と平均ベクトルの各次元の関係性の二つの要素に明示的に切り分けたものと解釈できる。

$\boldsymbol{U}^{(M)}$ のみを基底とした場合と同様に、最尤基準に基づいて $\boldsymbol{W}'$ の推定を行なうことができる。EM アルゴリズムに基づき、以下の更新式を得る。

$$\mathrm{vec}(\hat{\boldsymbol{W}}') = \left[\sum_{m=1}^{M} \overline{\gamma}_m^{(tar)} \boldsymbol{U}_m^\top \boldsymbol{U}_m \otimes \boldsymbol{U}^{(D)\top} \boldsymbol{\Sigma}_m^{-1} \boldsymbol{U}^{(D)}\right]^{-1} \mathrm{vec}(C) \tag{3.37}$$

$$\boldsymbol{C} = \sum_{t=1}^{T}\sum_{m=1}^{M} \gamma_{m,t} \boldsymbol{U}^{(D)\top} \boldsymbol{\Sigma}_m^{-1} (\boldsymbol{Y}_t^{(tar)} - \boldsymbol{b}_m^{(0)}) \boldsymbol{U}_m \tag{3.38}$$

$$\gamma_{m,t} = p(m|\boldsymbol{Y}_t^{(tar)}, \boldsymbol{\lambda}^{(EV)}) \tag{3.39}$$

$$\overline{\gamma}_m^{(tar)} = \sum_{t=1}^{T} \gamma_{m,t} \tag{3.40}$$

$$\boldsymbol{U}_m = \boldsymbol{U}^{(M)}(m,:) \in \mathcal{R}^{1\times K} \tag{3.41}$$

$$\boldsymbol{b}_m^{(0)} = \boldsymbol{b}(m,:)^\top \in \mathcal{R}^{D\times 1} \tag{3.42}$$

但し、$\boldsymbol{Y}_t^{(tar)}$ は t フレーム目の入力特徴量を表す。

3.6　Matrix Variate PLDA (MV-PLDA)

3.6.1　概要

2.8 節では従来の識別手法として PLDA について述べたが、これは入力がベクトル変量の特徴量に限られるため、テンソル分解に基づく重み行列 $\boldsymbol{W}_{ij}$ を入力する場合はベクトル化を行なって $\mathrm{vec}(\boldsymbol{W}_{ij})$ として扱う必要がある。しかし、テンソル分解に基づく重み行列の各行は平均ベクトルの各次元、各列は GMM の各分布という明確に異なる意味を持っており、各行と各列の二つの関係性を考慮することでより適切な識別が可能になることが期待される。そこで本研究では PLDA を行列変量に拡張し、行列の形で表される特徴量をそのままの形状で入力できる Matrix Variate PLDA (MV-PLDA) を提案する。本節では、MV-PLDA の基礎となる行列変量ガウス分布について述べたのち、MV-PLDA のモデルと定式化について述べる。

3.6.2　行列変量ガウス分布

行列 $\boldsymbol{X} \in \mathcal{R}^{n\times p}$ を確率変数とする行列変量ガウス分布の確率密度関数 $p(\boldsymbol{X})$ は以下の式で表される。

$$p(\boldsymbol{X}) = c^{-1} \exp\left[-\frac{1}{2}\mathrm{Tr}\{\boldsymbol{U}^{-1}(\boldsymbol{X}-\boldsymbol{M})\boldsymbol{V}^{-1}(\boldsymbol{X}-\boldsymbol{M})^\top\}\right]$$
$$\text{where } c = (2\pi)^{\frac{1}{2}np} |\boldsymbol{U}|^{\frac{1}{2}n} |\boldsymbol{V}|^{\frac{1}{2}p} \tag{3.43}$$

行列変量ガウス分布の確率密度関数は平均 $\boldsymbol{M} \in \mathcal{R}^{n \times p}$、行方向の分散共分散行列 $\boldsymbol{U} \in \mathcal{R}^{n \times n}$、列方向の分散共分散行列 $\boldsymbol{V} \in \mathcal{R}^{p \times p}$ の 3 つのパラメータを持ち、以下これを $\mathcal{N}_{\mathrm{mv}}(\boldsymbol{X}; \boldsymbol{M}, \boldsymbol{U}, \boldsymbol{V})$ と表記する。式 (3.43) は、ベクトル $\mathrm{vec}(\boldsymbol{X})$ に対する以下の確率密度関数 $p(\mathrm{vec}(\boldsymbol{X}))$ と等価である。

$$p(\mathrm{vec}(\boldsymbol{X})) = \mathcal{N}(\mathrm{vec}(\boldsymbol{X}); \mathrm{vec}(\boldsymbol{M}), \boldsymbol{V} \otimes \boldsymbol{U}) \tag{3.44}$$

このように、vec 演算子を導入することでベクトル変量ガウス分布と対応付けることができる。両者を比較すると、行列変量ガウス分布は分散のパラメータを行方向 ($\boldsymbol{U}$)、列方向 ($\boldsymbol{V}$) で別々に扱うことで分散に制約を課していると解釈できる。すなわち、テンソル分解に基づく重み行列をモデル化した行列変量ガウス分布は、GMM が捉える複数要因（音素に相当する）の間でのばらつきと各要因そのものが持つばらつきを切り分けて考えたモデルと解釈できる。ベクトル変量の PLDA では観測変数と潜在変数、残差の従う分布としてベクトル変量ガウス分布が仮定されるが、MV-PLDA では行列変量ガウス分布が仮定される。

3.6.3　MV-PLDA

i)　モデル

MV-PLDA では、クラス i の j 番目の発話を表す行列 $\boldsymbol{W}_{ij} \in \mathcal{R}^{d \times m}$ について以下のような分解を仮定する。

$$\boldsymbol{W}_{ij} = \boldsymbol{\mu} + \boldsymbol{F}_d \boldsymbol{H}_i \boldsymbol{F}_m^{\top} + \boldsymbol{\epsilon}_{ij} \tag{3.45}$$

但し $\boldsymbol{\mu}$ はバイアス、$\boldsymbol{F}_d \in \mathcal{R}^{d \times d'}$, $\boldsymbol{F}_m \in \mathcal{R}^{m \times m'}$ は基底、$\boldsymbol{H}_i \in \mathcal{R}^{d' \times m'}$ は $\mathcal{N}_{\mathrm{mv}}(\boldsymbol{0}, \boldsymbol{I}, \boldsymbol{I})$ に従う潜在変数、$\boldsymbol{\epsilon}_{ij}$ は $\mathcal{N}_{\mathrm{mv}}(\boldsymbol{0}, \boldsymbol{\Sigma}, \boldsymbol{\Gamma})$ に従う残差を表す（$\boldsymbol{\Sigma}, \boldsymbol{\Gamma}$ は全共分散行列）。バイアス $\boldsymbol{\mu}$、基底 $\boldsymbol{F}_d, \boldsymbol{F}_m$、残差の分散共分散行列 $\boldsymbol{\Sigma}, \boldsymbol{\Gamma}$ はクラス非依存、潜在変数 $\boldsymbol{H}_i$、残差 $\boldsymbol{\epsilon}_{ij}$ はクラス依存のパラメータである。

式 (3.45) に対し vec 演算子を導入し、両辺をベクトル化すると以下のようになる。

$$\mathrm{vec}(\boldsymbol{W}_{ij}) = \mathrm{vec}(\boldsymbol{\mu}) + (\boldsymbol{F}_m \otimes \boldsymbol{F}_d)\mathrm{vec}(\boldsymbol{H}_i) + \mathrm{vec}(\boldsymbol{\epsilon}_{ij}) \tag{3.46}$$

式 (3.46) と式 (2.44) を比較すると、MV-PLDA は、PLDA における基底に対し、$\boldsymbol{W}_{ij}$ の行方向の成分と列方向の成分を別々に扱うように制約を課したモデルと解釈できる。

ii)　パラメータの学習

ベクトル変量の PLDA と同様に、パラメータ $\theta = \{\boldsymbol{F}_d, \boldsymbol{F}_m, \boldsymbol{\Sigma}, \boldsymbol{\Gamma}\}$ は EM アルゴリズムを用いて最尤基準で推定される。このとき、バイアス $\boldsymbol{\mu}$ は全データ平均となり、E-Step と M-Step の更新式は以下のようになる。

E-Step:

$$\boldsymbol{E}[\boldsymbol{H}_i] = \mathrm{Cov}^{(\mathrm{row})}[\boldsymbol{H}_i] \boldsymbol{F}_d^{\top} \boldsymbol{\Sigma}^{-1} \sum_{j=1}^{N_i} (\boldsymbol{W}_{ij} - \boldsymbol{\mu}) \boldsymbol{\Gamma}^{-1} \boldsymbol{F}_m \mathrm{Cov}^{(\mathrm{col})}[\boldsymbol{H}_i] \tag{3.47}$$

$$\mathrm{Cov}^{(\mathrm{row})}[\boldsymbol{H}_i] = (\boldsymbol{I} + N_i \boldsymbol{F}_d^{\top} \boldsymbol{\Sigma}^{-1} \boldsymbol{F}_d)^{-1} \tag{3.48}$$

$$\mathrm{Cov}^{(\mathrm{col})}[\boldsymbol{H}_i] = (\boldsymbol{I} + \boldsymbol{F}_m^{\top} \boldsymbol{\Gamma}^{-1} \boldsymbol{F}_m)^{-1} \tag{3.49}$$

M-Step:

$$\boldsymbol{F}_d = \left[\sum_{i,j} \frac{1}{N_i}(\boldsymbol{W}_{ij} - \boldsymbol{\mu})\boldsymbol{\Gamma}^{-1}\boldsymbol{F}_m\boldsymbol{E}[\boldsymbol{H}_i]^\top\right]\boldsymbol{C}_d^{-1} \tag{3.50}$$

$$\boldsymbol{C}_d = \sum_{i=1}^{L}\left[\mathrm{Tr}(\boldsymbol{F}_m^\top\boldsymbol{\Gamma}^{-1}\boldsymbol{F}_m)\mathrm{Cov}^{(\mathrm{col})}[\boldsymbol{H}_i] + \boldsymbol{E}[\boldsymbol{W}_i]\boldsymbol{F}_m^\top\boldsymbol{\Gamma}^{-1}\boldsymbol{F}_m\boldsymbol{E}[\boldsymbol{H}_i]^\top\right] \tag{3.51}$$

$$\boldsymbol{F}_m = \left[\sum_{i,j} \frac{1}{N_i}(\boldsymbol{W}_{ij} - \boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{F}_m\boldsymbol{E}[\boldsymbol{H}_i]\right]\boldsymbol{C}_m^{-1} \tag{3.52}$$

$$\boldsymbol{C}_m = \sum_{i=1}^{L}\left[\mathrm{Tr}(\boldsymbol{F}_d^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{F}_d)\mathrm{Cov}^{(\mathrm{row})}[\boldsymbol{H}_i] + \boldsymbol{E}[\boldsymbol{W}_i]^\top\boldsymbol{F}_d^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{F}_d\boldsymbol{E}[\boldsymbol{H}_i]\right] \tag{3.53}$$

$$\boldsymbol{\Sigma} = \frac{1}{dN}\sum_{i,j}\left[(\boldsymbol{W}_{ij} - \boldsymbol{\mu})\boldsymbol{\Gamma}^{-1}(\boldsymbol{W}_{ij} - \boldsymbol{\mu})^\top - (\boldsymbol{W}_{ij} - \boldsymbol{\mu})\boldsymbol{\Gamma}^{-1}\boldsymbol{F}_m\boldsymbol{E}[\boldsymbol{H}_i]^\top\boldsymbol{F}_d^\top\right] \tag{3.54}$$

$$\boldsymbol{\Gamma} = \frac{1}{mN}\sum_{i,j}\left[(\boldsymbol{W}_{ij} - \boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{W}_{ij} - \boldsymbol{\mu}) - (\boldsymbol{W}_{ij} - \boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{F}_d\boldsymbol{E}[\boldsymbol{H}_i]\boldsymbol{F}_m^\top\right] \tag{3.55}$$

但し、L は言語数、N はパラメータの学習に用いた全発話数 $(= \sum_{i=1}^{L} N_i)$ を表す。

第4章

実験

4.1　はじめに

前章では、テンソル分解に基づく言語・話者情報表現と MV-PLDA を用いた識別について説明した。本章ではこれらの手法の有効性を検証するため、言語識別実験と話者識別実験の二種類の実験を行なう。

4.2　言語識別実験

4.2.1　コーパス

言語識別用の多言語音声コーパスとして The National Institute of Standards and Technology Language Recognition Evaluation (NIST LRE) の 2003・2005・2007 年版を用いて実験を行なった。このコーパスには、多数の言語・方言による電話での会話が継続長 3 秒・10 秒・30 秒の三つのカテゴリーに分かれて収録されている。本実験では、識別対象の言語のセット (target languages) は The 2007 NIST Language Recognition Evaluation Plan [1] に従いアラビア語・ベンガル語・ペルシア語・ドイツ語・日本語・韓国語・ロシア語・スペイン語・タミル語・タイ語・ベトナム語・中国語・英語・ヒンドゥスタン語の 14 言語に設定し、方言については区別しないものとした。

4.2.2　実験条件

各音声データについて表 4.2 の条件で音響分析を行ない、Voice Activity Detection (VAD) と CMN, VTLN を施した音響特徴量系列を計算した。VAD は無音区間の除去を行なう処理である。学習・テストに用いた言語毎のデータ数を表 4.1 に示す。UBM は 24,577 発話を用いて表 4.3 に示した条件で推定した。UBM の学習に用いたデータからフランス語のデータを除いた 23,665 発話を用いて言語空間の基底を求め、テンソル分解に基づく重み行列 (Tensor-based)、Tensor-based における SAT に基づく重み行列 (Tensor-based SAT)、Bilinear 基底に基づく重み行列 (Tensor-based bilinear) を計算した。ベースラインとして i-vector、Eigenvoice に基づく重みベクトル (EV-based)、EVC における SAT に基づく重みベクトル (EV-based SAT) を計算した。発話 GMM を推定する際の MAP 推定のパラメータ (τ) は継続長カテゴリー 3 秒、10 秒、30 秒の発話についてそれぞれ 0.25, 0.83, 2.5 とした。式 (2.20) における各項の係数の値を凡そ揃えるため、継続長カテゴリー毎に別々に設定した。EV-based, Tensor-based における基底の個数 K, K_M はそれぞれ 600, 32 とした。Tensor-based bilinear における基底の個数は $(K_M, K_D) = (32, 49)$ とした。i-vector の次元数は 600 とした。i-vector と EV-based の重み行列の識別には PLDA を用い、Tensor-based の重み行列の識別には PLDA と MV-PLDA を用いた。各特徴量について、LDA と白色化を施してから PLDA, MV-PLDA に入力した。行列の LDA は DATER [28] に基づいて行なった。言語空間の基底計算に用いた 23,665 発話で学習、NIST LRE 2007 Evaluation Set 6,474 発話（3 秒、10 秒、30 秒各 2,158 発話）でテストを行なった。音響特徴量系列と i-vector の計算には Kaldi Toolkit[2]、GMM の学習には Hidden Markov Model Toolkit (HTK)[3]、PLDA の実装には MSR Identity Toolbox[4] を用いた。評価として、Equal Error Rate (EER) をテストデータの継続長カテゴリ (3 秒、10 秒、30 秒) 毎に計算した。EER は False Rejection Rate (FRR)

[1] http://www.itl.nist.gov/iad/mig/tests/lre/2007/LRE07EvalPlan-v8b.pdf

表 4.1: 実験に用いた言語毎の音声データ数

Language	UBM	Basis/Train	Test
Arabic	906	906	240
Bengali	200	200	240
Chinese	4502	4502	1194
English	5739	5739	720
Farsi	717	717	240
German	970	970	240
Hindustani	1335	1335	720
Japanese	2050	2050	240
Korean	1655	1655	240
Russian	440	440	480
Spanish	3007	3007	720
Tamil	1237	1237	480
Thai	200	200	240
Vietnamense	707	707	480
French	912	-	-
Total	24577	23665	6474

と False Acceptance Rate (FAR) が等しくなるときのエラー率である。

4.2.3　実験結果 (1): PLDA を用いた識別

表 4.4 に PLDA を用いた場合の実験結果を示す。3s, 10s, 30s はそれぞれ 3 秒、10 秒、30 秒のテストデータに関する結果を表す。EV-based, Tensor-based で重みをそれぞれ式 (2.25), (3.11) の最小二乗解として求めた場合を MMSE、重みをそれぞれ式 (2.27), (3.13) の最尤基準で求めた場合を ML、重みをそれぞれ式 (2.34), (3.21) の MAP 基準で求めた場合を MAP と表記している。

i)　重みの計算方法による比較

EV-based に関しては、MMSE に比べて ML と MAP で良い結果となったが、Tensor-based に関しては MMSE が最も良い結果となった。Tensor-based では基底が EV-based よりもコンパクトであるが、代わりに重みのパラメータ数（次元数）を EV-based よりも大きく取る必要があり、一発話という少量のサンプルから重み行列を計算する場合は MMSE が適していると考えられる。

ii)　SAT の有無による比較

EV-based に関しては、SAT を導入することで 3s, 10s の EER が大きく改善し、SAT が短い発話の重み推定に適していることが分かった。一方 Tensor-based に関しては、SAT を導入する

[2]http://kaldi-asr.org/
[3]http://htk.eng.cam.ac.uk/
[4]http://research.microsoft.com/apps/pubs/default.aspx?id=211317

表 4.2: 音響分析条件

サンプリング	8 bit / 8 kHz
窓	25 ms length / 10 ms shift
音響的特徴	MFCC (7 次元) + SDC ($(N, d, P, k) = (7, 1, 3, 7)$) (49 次元)

表 4.3: UBM 推定条件

音響的特徴	MFCC (7 次元) + SDC ($(N, d, P, k) = (7, 1, 3, 7)$) (49 次元)
パラメータ推定	最尤推定
GMM	2,048 混合ガウス分布 （対角共分散行列）

ことで全体的に EER が高くなる結果となった。SAT では重み更新→基底更新を繰り返すことによって基底を計算するため、重みの次元数が大きい Tensor-based に関しては、SAT を導入することで基底計算に用いたデータに過剰に適応した基底が計算されたことが考えられる。

iii)　Tensor-based と Tensor-based bilinear の比較

Tensor-based では 式 (3.8) における $U^{(M)}$ のみを基底として用いる場合と、$U^{(M)}$ と $U^{(D)}$ の二つを基底として用いる場合の二種類の基底について検討したが、前者の基底を用いたほうが良い結果となった。音響特徴量として MFCC に加えて SDC を用いたが、SDC によって捉えた複数の時間幅における MFCC の時間変化が独立した情報を持っており、$U^{(D)}$ の方向に圧縮することが逆効果となったことが考えられる。

iv)　手法間の比較

全体的には Tensor-based, i-vector に比べて EV-based が良い結果となった。Tensor-based (MMSE) は 30s では i-vector に比べて EER が低いが、SAT を導入した場合も含めれば i-vector よりも EV-based が高い性能を発揮した。i-vector と EV-based SAT は基本的には等価であるが、SVD によって求めた基底を初期値として用いたことがより効果的だったと考えられる。

4.2.4　実験結果 (2): MV-PLDA を用いた識別

表 4.5 に MV-PLDA を用いた場合の実験結果を示す。MV-PLDA を用いて識別した場合、全体的に PLDA を用いた場合と比べて良い結果は得られなかった。特に、PLDA と比べて 10s, 30s での EER の悪化が大きい。Tensor-based bilinear (MMSE) について MV-PLDA の学習データをそのままテストしたところ、EER が 26.09 % となった。3.6 節において MV-PLDA が PLDA に対して制約を課したモデルであると述べたが、この制約が強すぎたためにモデルが正しく学習できていないことが考えられる。

表 4.4: EER (using PLDA) [%]

	3s	10s	30s
i-vector	24.14	16.03	11.68
EV-based (MMSE)	27.25	18.12	12.56
EV-based (ML)	24.51	15.71	10.91
EV-based (MAP)	25.63	15.57	**9.22**
EV-based SAT (ML)	**21.96**	14.29	10.16
EV-based SAT (MAP)	22.39	**13.52**	11.05
Tensor-based (MMSE)	27.53	17.56	10.31
Tensor-based (ML)	28.38	18.59	13.86
Tensor-based (MAP)	30.58	19.79	13.28
Tensor-based SAT (ML)	33.92	23.59	20.59
Tensor-based SAT (MAP)	37.39	29.84	27.90
Tensor-based bilinear (MMSE)	28.41	18.35	11.47
Tensor-based bilinear (ML)	28.63	19.22	14.03

表 4.5: EER (using MV-PLDA) [%]

	3s	10s	30s
Tensor-based (MMSE)	**35.87**	**31.19**	**28.54**
Tensor-based (ML)	36.01	30.86	29.23
Tensor-based (MAP)	50.09	41.20	41.98
Tensor-based SAT (ML)	43.52	42.54	41.77
Tensor-based SAT (MAP)	47.69	46.34	45.23
Tensor-based bilinear (MMSE)	39.20	32.85	29.46
Tensor-based bilinear (ML)	38.46	32.77	29.43

4.3　話者識別実験

4.3.1　コーパス

話者識別用の音声コーパスとして、日本音響学会新聞記事読み上げ音声コーパス (Japanese Newspaper Article Sentences; JNAS) [1] を用いて実験を行なった。コーパスの概要を表 4.6 に示す。このコーパスには、男女各 153 名、計 306 名の日本語読み上げ音声がヘッドセットマイク (HS) と卓上型マイク (DT) の二本のマイクで同時録音されたものが収録されている。収録文は、各話者について新聞記事の読み上げが 100 文程度と音素バランス文の読み上げが 50 文程度となっている。本実験では、識別対象の話者をラベル m001 - m060 の男性 60 名、ラベル f001 - f060 の女性 60 名、計 120 名に設定した。

[1] http://www.mibel.cs.tsukuba.ac.jp/_090624/jnas/

表 4.6: JNAS コーパスの概要

話者		男女各 129 名（計 258 名）
読み上げテキスト	新聞記事文	155 セット（約 100 文/セット）
	音素バランス文	10 セット（約 50 文/セット）
文数/話者	新聞記事文	1 セット
	音素バランス文	1 セット
収録マイク		headset (HS) と desktop (DT) の二本のマイク
音声データ		16 bit 量子化、16 kHz サンプリング

表 4.7: 実験に用いた一話者あたりの音声データ数

	UBM/Basis	Set 1		Set 2		Set 3	
		Train	Test	Train	Test	Train	Test
新聞記事	50	10	約 90	20	約 80	50	約 50
音素バランス文	50	20	約 30	40	約 10	40	約 10
計	100	30	約 120	60	約 90	90	約 60

4.3.2　実験条件

各音声データについて表 4.8 の条件で音響分析を行ない、VAD と CMN を施した音響特徴量系列を計算した。学習・テストに用いた一話者あたりのデータ数を表 4.7 に示す。UBM は表 4.9 に示した条件で、識別対象話者でない男女各 69 名、計 138 名について各話者・各収録環境の新聞記事 50 文、音素バランス文 50 文、計 37,200 発話を用いて推定した。UBM の推定に用いたのと同様のデータを用いて話者空間の基底を求め、テンソル分解に基づく重み行列 (Tensor-based)、Tensor-based における SAT に基づく重み行列 (Tensor-based SAT)、Bilinear 基底に基づく重み行列 (Tensor-based bilinear) を計算した。ベースラインとして i-vector、Eigenvoice に基づく重みベクトル (EV-based)、EVC における SAT に基づく重みベクトル (EV-based SAT) を計算した。発話 GMM を推定する際の MAP 推定のパラメータ (τ) は 0.83 とした。EV-based, Tensor-based における基底の個数 K, K_M はそれぞれ 400, 32 とした。Tensor-based bilinear における基底の個数は $(K_M, K_D) = (32, 45)$ とした。i-vector の次元数は 400 とした。i-vector と EV-based の重み行列の識別には PLDA を用い、Tensor-based の重み行列の識別には PLDA と MV-PLDA を用いた。行列の LDA は DATER [28] に基づいて行なった。各特徴量について、LDA と白色化を施してから PLDA, MV-PLDA に入力した。学習・テストは収録環境 (HS/DT) のマッチ条件 (HS-HS, DT-DT)、ミスマッチ条件 (HS-DT, DT-HS) の計 4 条件で行ない、各条件について学習データ数を変えて三通り実験した。Set 1 では一話者あたり 30 発話、計 3,600 発話で学習、それ以外の計 14,894 発話でテスト、Set 2 では一話者あたり 60 発話、計 7,200 発話で学習、それ以外の計 11,294 発話でテスト、Set 3 では一話者あたり 30 発話、計 10,800 発話で学習、それ以外の計 7,694 発話でテストとした。実装には言語識別実験と同様のものを用いた。評価として、言語識別実験と同様に EER を用いた。

表 4.8: 音響分析条件

サンプリング	16 bit / 16 kHz
窓	25 ms length / 10 ms shift
音響的特徴	MFCC (20 次元) + ΔMFCC (20 次元) + ΔΔMFCC (20 次元)

表 4.9: UBM 推定条件

音響的特徴	MFCC (20 次元) + ΔMFCC (20 次元) + ΔΔMFCC (20 次元)
パラメータ推定	最尤推定
GMM	2,048 混合ガウス分布 （対角共分散行列）

表 4.10: EV-based, Tensor-based, Tensor-based bilinear の重みの各計算方法による Set 1 での EER (PLDA) [%]

	HS-HS	HS-DT	DT-HS	DT-DT
EV-based (MMSE)	**0.35**	**2.67**	**3.18**	0.37
EV-based (ML)	0.84	3.58	4.86	0.61
EV-based (MAP)	**0.35**	4.16	5.07	**0.35**
Tensor-based (MMSE)	0.43	**3.21**	**4.04**	**0.35**
Tensor-based (ML)	2.72	5.72	7.17	2.04
Tensor-based (MAP)	**0.40**	4.99	5.97	0.39
Tensor-based bilinear (MMSE)	**0.33**	**2.66**	**3.27**	**0.30**
Tensor-based bilinear (ML)	2.21	5.07	6.32	1.63

4.3.3　実験結果 (1): PLDA を用いた識別

i)　重みの計算方法による比較

表 4.10 に重みの各計算方法についての Set 1 での実験結果を示す。EV-based に関しては、ML や MAP と比べて MMSE で良い結果となった。ML と MAP を比較すると ML の方が EER が高いことから、EV-based においても重みの過学習が起こっていると考えられ、言語識別実験の結果と異なる傾向を示した。また Tensor-based に関しても、言語識別実験の結果と同様に ML や MAP と比べて MMSE で良い結果となった。これらの結果は Set 2, 3 においても同様であった（付録 A 参照）。

ii)　SAT の有無による比較

表 4.11 に SAT を導入した場合、導入しなかった場合についての Set 1 での実験結果を示す。EV-based に関しては、SAT を導入することによる EER の改善は見られず、言語識別実験の結果と異なる傾向を示した。言語識別実験では GMM を学習する際に VTLN を施した音響特徴量を用いたことで音響空間が縮小したため、SAT を用いて特徴量空間を斜交基底でよりコンパクトに構築した効果によって識別性能が向上したと考えられる。また、Tenor-based に関しても言語識別実験の結果と同様に SAT を導入することによる EER の改善は見られなかった。これらの結

表 4.11: EV-based, Tensor-based に対する SAT の導入前後における Set 1 での EER (PLDA) [%]

	HS-HS	HS-DT	DT-HS	DT-DT
EV-based (MMSE)	**0.35**	**2.67**	**3.18**	**0.37**
EV-based SAT (ML)	0.61	4.39	7.02	0.58
EV-based SAT (MAP)	0.56	4.29	6.19	0.56
Tensor-based (MMSE)	**0.43**	**3.21**	**4.04**	**0.35**
Tensor-based SAT (ML)	9.42	14.09	16.13	8.09
Tensor-based SAT (MAP)	9.94	14.80	16.48	8.03

表 4.12: Tensor-based, Tensor-based bilinear における各 Set での EER (PLDA) [%]

Set 1

	HS-HS	HS-DT	DT-HS	DT-DT
Tensor-based (MMSE)	0.43	3.21	4.04	0.35
Tensor-based bilinear (MMSE)	**0.33**	**2.66**	**3.27**	**0.30**

Set 2

	HS-HS	HS-DT	DT-HS	DT-DT
Tensor-based (MMSE)	0.15	1.77	2.25	**0.17**
Tensor-based bilinear (MMSE)	**0.14**	**1.75**	**2.22**	0.18

Set 3

	HS-HS	HS-DT	DT-HS	DT-DT
Tensor-based (MMSE)	0.09	**1.35**	1.72	**0.10**
Tensor-based bilinear (MMSE)	**0.08**	1.42	**1.66**	0.12

果は Set 2, 3 においても同様であった（付録 A 参照）。

iii)　Tensor-based と Tensor-based bilinear の比較

表 4.12 に Tensor-based, Tensor-based bilinear についての各 Set での実験結果を示す。全体的に Bilinear 基底を用いた場合の方が良い結果となり、特に Set 1 に関しては性能向上が最も大きかった。重み行列を GMM の各分布の平均ベクトル (D) の方向からも圧縮して次元数を削減する効果が学習データが少量の場合に特に大きいと考えられる。

iv)　手法間の比較

表 4.13 に i-vector, EV-based, Tensor-based bilinear について各 Set での実験結果を示す。全体的に Tensor-based bilinear が EV-based, i-vector と比べて良い結果となり、さらに Set 1 → 2 → 3 と学習データが増えるほど性能向上が大きくなった。これによりテンソル分解に基づく話者情報表現の有効性が示され、学習データが多い場合に特に有効であることが分かった。また、

表 4.13: i-vector, EV-based, Tensor-based bilinear における各 Set での EER (PLDA) [%]

Set 1

	HS-HS	HS-DT	DT-HS	DT-DT
i-vector	0.38	3.18	4.46	0.43
EV-based (MMSE)	0.35	2.67	**3.18**	0.37
Tensor-based bilinear (MMSE)	**0.33**	**2.66**	3.27	**0.30**

Set 2

	HS-HS	HS-DT	DT-HS	DT-DT
i-vector	0.36	2.97	4.24	0.36
EV-based (MMSE)	0.31	2.28	3.07	0.33
Tensor-based bilinear (MMSE)	**0.14**	**1.75**	**2.22**	**0.18**

Set 3

	HS-HS	HS-DT	DT-HS	DT-DT
i-vector	0.31	2.51	3.39	0.35
EV-based (MMSE)	0.23	1.99	2.59	0.30
Tensor-based bilinear (MMSE)	**0.08**	**1.42**	**1.66**	**0.12**

EV-based が EV-based SAT と i-vector と比べて良い結果となっているが、基底を計算するのに用いたデータ数が少ない場合には PPCA よりも DPCA が有効であると考えられる。

4.3.4　実験結果 (2): MV-PLDA を用いた識別

表 4.14 に MV-PLDA を用いた場合の Set 1 での実験結果を示す。言語識別実験の結果と同様に、MV-PLDA を用いて識別した場合、全体的に PLDA を用いた場合と比べて良い結果は得られなかった。表 4.15 に Tensor-based bilinear (MMSE) における各 Set での実験結果を示す。Set 1 → 2 → 3 と学習データが増やしていくことによる EER の改善は見られず、むしろ悪化する場合があった。Tensor-based bilinear (MMSE) について MV-PLDA の学習データ (Set 1) をそのままテストしたところ、EER が HS-HS 条件では 0.90 %、DT-DT 条件では 0.75 % となりいずれも 0 % にはならなかった。言語識別実験と同様に、MV-PLDA における制約が強すぎるためにモデルが正しく学習できていないことが考えられる。

4.4　二つの実験結果に対する言語識別・話者識別のタスクの違いに基づく考察

i-vector、Eigenvoice に基づく手法、テンソル分解に基づく手法ではいずれも発話を GMM でモデル化し、その各分布の平均を基に特徴量空間を構築する。GMM の各分布が凡そ各音素・単音等に対応していると仮定した場合、発話 GMM の各分布の平均は「その発話において」各音素・単音がどのような音響的特徴で実現されたかを捉えていることになり、言語情報と非言語情報（話者情報）の両方を含んでいることが期待される。GMM-SV では発話 GMM の各分布の平均をそのまま特徴量として用い、言語情報と話者情報の分離は識別器に頼る手法となっている。

表 4.14: Tensor-based, Tensor-based bilinear における Set 1 での EER (MV-PLDA) [%]

	HS-HS	HS-DT	DT-HS	DT-DT
Tensor-based (MMSE)	**1.23**	5.20	7.07	1.51
Tensor-based (ML)	6.60	11.24	13.27	6.62
Tensor-based (MAP)	6.15	13.62	11.65	4.60
Tensor-based SAT (ML)	14.38	19.04	20.06	14.04
Tensor-based SAT (MAP)	18.40	22.20	21.29	14.89
Tensor-based bilinear (MMSE)	1.31	**4.51**	**4.39**	**1.33**
Tensor-based bilinear (ML)	9.86	14.21	14.47	8.45

表 4.15: Tensor-based bilinear (MMSE) における各 Set での EER (MV-PLDA) [%]

	HS-HS	HS-DT	DT-HS	DT-DT
Set 1	**1.31**	4.51	4.39	1.33
Set 2	1.44	4.38	**3.85**	**1.13**
Set 3	1.34	**3.83**	3.91	1.16

これに対し、i-vector、Eigenvoice に基づく手法では、特徴量空間の構築を行なう過程でこの二つの情報を分離し、不要な情報を除去することによって識別性能の向上をねらっている。本研究で提案したテンソル分解に基づく手法では、これらの手法をベースとして更に GMM の各分布の関係性を明示的に捉えることで識別性能の向上をねらっていたが、実験結果から話者識別においては有効であるが言語識別においては有効でないことが分かった。発話 GMM の各分布の平均は話者の特徴をより表現していると考えられる。

言語識別分野では、GMM-SV が提案される以前は音素の連鎖の情報、すなわち音素 N-gram に基づく手法が主流であった [13]。例えば、/p/ という音素はペルシャ語には存在するがアラビア語には存在しない。また、ドイツ語の ”spiel” という単語の音素表記 /sh p iy l/ の /sh p/ に着目すると、ドイツ語にも英語にも /sh/, /p/ という音素は存在するが、/sh p/ という音素の連鎖は英語には存在しない。このような各言語の音素のセットや連鎖の傾向の違いを捉えて識別を行なうことが有効だと考えられきた。[13] では、英語の音素認識器を用いて入力音声を英語音素列にデコードし、英語音素列を用いて学習された N-gram 言語モデルの尤度に基づいて識別を行なう Phone Recognition Followed by Language Modeling (PRLM) について検討されている。その後、話者識別分野で提案された GMM-SV や i-vector が言語識別分野にも導入され、主流がこれらに移り変わったが、近年では再び、このように音素の連鎖や出現頻度に着目した手法が検討され始めている [29, 30, 31]。[29] では、発話 GMM の各分布の重みを一列に連結した GMM weight supervector を基に低次元の特徴量 r-vector を i-vector のアプローチと同様の手順で計算し、r-vector による識別結果と i-vector による識別結果を統合することにより、i-vector 単体で識別した場合と比べて識別性能が向上している。発話 GMM の各分布の重みはおよそ「その発話における」音素の出現頻度を捉えていることが期待されるが、これは言語らしさと発話内容の二つの情報を含んでいる。r-vector は GMM weight supervector の次元圧縮によりこれらを分離し、凡そ言語らしさのみを抽出した特徴量と解釈でき、発話 GMM の各分布の平均に対して付加的な

情報を持っていることが実験的に示されている。[30] では Deep Neural Network (DNN) を用いて音声の各フレームについて言語クラス事後確率を推定し、それらのフレーム平均に基づいて識別を行なう手法が検討されており、これにより i-vector と比べて大きく識別性能が向上している。DNN によって計算された一発話の言語クラス事後確率系列は音素・単音の系列に類似したものと解釈できるが、それらの時間平均を取るだけでは出現頻度しか捉えることができない。この点に着目して [31] では一発話の言語クラス事後確率系列をベクトル量子化に基づき離散系列化し、その離散系列を言語モデルによってモデル化する手法が検討されている。これは [30] の DNN-based のアプローチと PRLM のアプローチを統合した手法と解釈でき、[30] の手法を上回る識別性能を実現している。

このように、言語識別と話者識別はいずれもクラス分類のタスクであるが、前者は言語的な情報、後者は非言語的な情報を扱うという点で異なり、それぞれに最適な手法も異なってくると考えられる。言語識別実験と話者識別実験の二つの実験結果から、本研究で提案したテンソル分解に基づく手法が話者識別により適した手法であることが分かった。

第5章

結論

5.1　まとめ

本論文では、より高度な音声インターフェースを実現するための技術として音声からの言語識別・話者識別の二つに着目し、その精度向上を目指した。従来言語識別・話者識別において標準的に用いられてきた i-vector では考慮されていなかった GMM の各分布の関係性を捉えた新しい特徴量表現としてテンソル分解に基づく手法について検討し、これに合わせた新たな識別手法として従来の PLDA を行列変量に拡張した MV-PLDA を提案した。テンソル分解に基づく音声の特徴量表現の手法に関しては、実験により話者識別では i-vector と比べて一定の精度向上が確認でき、さらに Bilinear 基底を用いて GMM の各分布の関係性と平均ベクトルの次元間の関係性の両方を考慮した話者情報表現が特に有効であることが確認できた。言語識別では i-vector と比べて精度向上が確認できなかったものの、実験結果に基づき、話者識別・言語識別のタスクとしての違いから両者において識別に有効な特徴量表現が異なる可能性について指摘した。また、MV-PLDA に関しては言語識別・話者識別タスクでは従来の PLDA からの識別精度の向上が確認できなかった。MV-PLDA は PLDA に制約を課したモデルと解釈できるが、この制約が強すぎたことが識別性能の低下につながった可能性を示唆する実験結果となった。

5.2　今後の課題

言語識別タスクでは音素の連鎖や出現頻度の情報が識別に有効である可能性について述べたが、テンソル分解に基づく特徴量に加えて、発話 GMM の各分布の重みに基づく特徴量を用いることで識別精度の向上が期待できる。また、本論文では i-vector と条件を揃えるために Eigenvoice に基づく手法、テンソル分解に基づく手法における SAT として発話単位の SAT について検討したが、特徴量空間の構築に用いる学習データが少量の場合には基底が学習データに過剰に適応してしまう可能性があることが実験結果から分かった。SAT を言語（話者）単位にすることでこの問題を軽減できると考えられるため、これについても検討する必要がある。

謝辞

二年間、本研究を進めるにあたって日頃よりご指導を頂いた峯松信明教授に深く感謝いたします。峯松信明教授からは、ご多忙にもかかわらず様々なアドバイスをいただきました。また、日頃の研究生活を支えてくださった高橋登技官、秘書の池上恵氏、折茂結実子氏にも厚く御礼申し上げます。

また、毎日のように相談に乗ってくださった齋藤大輔助教に感謝いたします。研究に関する小さなことや実装に関する知識から、研究指針や理論的な知識、研究に対する心構えまで、様々なことを教えていただき、大変お世話になりました。諸先輩方には、研究だけでなく生活のことまでいろいろとお世話になりました。卒論のときから一緒に頑張ってきた同期の百武恭汰氏と大学院・研究室での生活を楽しく充実したものにしてくださった後輩の皆様にも感謝いたします。

最後に、これまで私を支えてくださった家族、友人に感謝いたします。本当に、ありがとうございました。

2016 年 2 月 4 日

鈴木 颯

参考文献

[1] T. Hori, S. Araki, T. Yoshioka, M. Fujimoto, S. Watanabe, T. Oba, A. Ogawa, K. Otsuka, D. Mikami, K. Kinoshita, T.Nakatani, A. Nakamura, and J. Yamato, "Low-latency Real-time Meeting Recognition and Understanding Using Distant Microphones and Omni-directional Camera," IEEE Transactions on Audio, Speech, and Language Processing, vol. 20, no. 2, pp. 499–513, 2012.

[2] 辻川剛範, 岡部浩司, 花沢健, "雑音下でも頑健に動作する音声 UI 技術とその応用," NEC 技報 65 巻 3 号, pp. 114–118, 2013.

[3] Y. K. Muthusamy, E. Barnard, and R. A. Cole, "Reviewing automatic language identification," Signal Processing Magazine, IEEE, vol. 11, no. 4, pp. 33–41, 1994.

[4] E. Ambikairajah, Haizhou Li, Liang Wang, Bo Yin, and V. Sethu, "Language identification: A tutorial," Circuits and Systems Magazine, IEEE, vol. 11, no. 2, pp. 82–108, 2011.

[5] R. G. Leonard and G. R. Doddington, "Automatic Language Identication," Technical Report RADC-TR-74-200, Air Force Rome Air Development Center, 1974.

[6] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," IEEE Transactions Speech Audio Processing, no. 3, pp. 72–83, 1995.

[7] N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language Recognition via Ivectors and Dimensionality Reduction," Proc. INTERSPEECH, pp. 857–860, 2011.

[8] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End Factor Analysis for Speaker Verication," IEEE Transactions on Audio, Speech, and Language Processing, vol. 19, no. 4, pp. 788–798, 2011.

[9] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support Vector Machines using GMM Supervectors for Speaker Verification," IEEE Signal Processing Letters, vol. 13, pp. 308–311, 2006.

[10] チン・トゥアン・トゥー, 齋藤大輔, 峯松信明, 広瀬啓吉, "テンソル分解に基づく話者情報表現を用いた話者識別の検討," 日本音響学会春季講演論文集, pp. 217–220, 2015.

[11] P. Kenny, "Bayesian Speaker Verification with Heavy-Tailed Priors," Proc. Odyssey, pp. 1–8, 2010.

[12] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of I-vector Length Normalization in Speaker Recognition Systems," Proc. INTERSPEECH, pp. 249–252, 2011.

[13] M. A. Zissman and E. Singer, “Automatic language identification of telephone speech messages using phoneme recognition and N-gram modeling,” Proc. ICASSP, vol. 1, pp. 305–308, 1994.

[14] X. Ma, A. Nemoto, N. Minematsu, Y. Qiao, and K. Hirose, “Structural analysis of dialects, sub-dialects, and sub-sub dialects of Chinese,” Proc. INTERSPEECH, pp. 2219–2222, 2009.

[15] W. M. Campbell, E. Singer, P. A. Torres-Carrasquillo, and D. A. Reynolds, “Language Recognition with Support Vector Machines,” Proc. Odyssey, pp. 41–44, 2004.

[16] 中川聖一, 小林聡, 峯松信明, 宇津呂武仁, 秋葉友良, 北岡教英, 山本幹雄, 甲斐充彦, 山本一公, 土屋雅稔, “音声言語処理と自然言語処理,” コロナ社, 2013.

[17] B. Bielefeld, “Language identification using shifted delta cepstrum,” In Fourteenth Annual Speech Research Symposium, 1994.

[18] B. Atal, “Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification,” Journal of the Acoustical Society of America, vol. 55, no. 6, pp. 1304–1312, 1974.

[19] E. Eide and H. Gish, “A parametric approach to vocal tract length normalization,” Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol.1, pp.346348, 1996.

[20] M. Pitz and H. Ney, “Vocal tract normalization equals linear transformation in cepstral space,” IEEE Transanctions on Speech and Audio Processing, vol. 13, pp. 930–944, 2005.

[21] R. Kuhn, J. Junqua, P. Nguyen, and N. Niedzielski, “Rapid Speaker Adaptation in Eigenvoice Space,” IEEE Transactions on Speech and Audio Processing, vol. 8, no. 6, pp. 695–707, 2000.

[22] T. Toda, Y. Ohtani, and K. Shikano, “Eigenvoice Conversion Based on Gaussian Mixture Model,” Proc. INTERSPEECH, pp. 2446–2449, 2006.

[23] Y. Ohtani, T. Toda, H. Saruwatari, and K. Shikano, “Speaker Adaptive Training for One-to-Many Eigenvoice Conversion Based on Gaussian Mixture Model,” Proc. INTERSPEECH, pp. 1981–1984, 2007.

[24] S. J. D. Prince and J. H. Elder, “Probabilistic Linear Discriminant Analysis for Inferences About Identity,” Proc. IEEE International Conference on Computer Vision (ICCV) , pp. 1–8, 2007.

[25] D. Saito, K. Yamamoto, N. Minematsu, and K. Hirose, “One-to-Many Voice Conversion Based on Tensor Representation of Speaker Space,” Proc. INTERSPEECH, pp. 653–656, 2011.

[26] L. R. Tucker, “Some mathematical notes on three-mode factor analysis,” Psychometria, vol. 31, no. 3, pp. 279–311, 1966.

[27] D. Saito, N. Minematsu, and K. Hirose, “Effects of Speaker Adaptive Training on Tensor-based Arbitrary Speaker Conversion,” Proc. INTERSPEECH, pp. 98–101, 2012

[28] S. Yan, D. Xu, Q Yang, L. Zhang, X. Tang, and H. Zhang, “Discriminant Analysis with Tensor Representation,” Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 526–532, 2005

[29] M. H. Bahari, N. Dehak, and H. Van hamme, “Gaussian mixture model weight supervector decomposition and adaptation,” Tech. Rep., 2013.

[30] I. Lopez-Moreno, J. Gonzalez-Dominguez and O. Plchot, “Automatic Language Identification Using Deep Neural Networks,” Proc. ICASSP, pp. 5337–5341, 2014.

[31] 増村 亮, Sheri Sever, 浅見 太一, 政瀧 浩和, 阪内 澄宇, “DNN 事後確率系列の言語モデル化に基づく言語識別,” 言語処理学会第 21 回年次大会発表論文集, pp. 916–919, 2015.

発表文献

国内研究会・全国大会

[1] 鈴木颯, 齋藤大輔, 峯松信明, 広瀬啓吉, “構造的表象と GMM スーパーベクトルを用いた言語識別に関する検討,” 日本音響学会春季講演論文集, pp. 113–116, 2014.

[2] 鈴木颯, 齋藤大輔, 峯松信明, “テンソル分解に基づく言語情報表現を用いた言語識別に関する検討,” 日本音響学会秋季講演論文集, pp. 181–184, 2015.

[3] 鈴木颯, 齋藤大輔, 峯松信明, “テンソル分解に基づく音声表現とその言語識別・話者識別への応用,” 電子情報通信学会音声研究会資料, 2016（発表予定）.

学位論文

[4] 鈴木颯, “構造的表象と GMM スーパーベクトルを用いた言語識別に関する検討,” 東京大学工学部電子情報工学科卒業論文, 2014.

付録 A

話者識別実験の結果一覧

表 A.1: EER (using PLDA) [%]: Set 1

	HS-HS	HS-DT	DT-HS	DT-DT
i-vector	0.38	3.18	4.46	0.43
EV-based (MMSE)	0.35	2.67	**3.18**	0.37
EV-based (ML)	0.84	3.58	4.86	0.61
EV-based (MAP)	0.35	4.16	5.07	0.35
EV-based SAT (ML)	0.61	4.39	7.02	0.58
EV-based SAT (MAP)	0.56	4.29	6.19	0.56
Tensor-based (MMSE)	0.43	3.21	4.04	0.35
Tensor-based (ML)	2.72	5.72	7.17	2.04
Tensor-based (MAP)	0.40	4.99	5.97	0.39
Tensor-based SAT (ML)	9.42	14.09	16.13	8.09
Tensor-based SAT (MAP)	9.94	14.80	16.48	8.03
Tensor-based bilinear (MMSE)	**0.33**	**2.66**	3.27	**0.30**
Tensor-based bilinear (ML)	2.21	5.07	6.32	1.63

表 A.2: EER (using MV-PLDA) [%]: Set 1

	HS-HS	HS-DT	DT-HS	DT-DT
Tensor-based (MMSE)	**1.23**	5.20	7.07	1.51
Tensor-based (ML)	6.60	11.24	13.27	6.62
Tensor-based (MAP)	6.15	13.62	11.65	4.60
Tensor-based SAT (ML)	14.38	19.04	20.06	14.04
Tensor-based SAT (MAP)	18.40	22.20	21.29	14.89
Tensor-based bilinear (MMSE)	1.31	**4.51**	**4.39**	**1.33**
Tensor-based bilinear (ML)	9.86	14.21	14.47	8.45

表 A.3: EER (using PLDA) [%]: Set 2

	HS-HS	HS-DT	DT-HS	DT-DT
i-vector	0.36	2.97	4.24	0.36
EV-based (MMSE)	0.31	2.28	3.07	0.33
EV-based (ML)	0.68	3.00	4.67	0.58
EV-based (MAP)	0.32	3.97	5.64	0.3
EV-based SAT (ML)	0.52	3.96	6.06	0.49
EV-based SAT (MAP)	0.51	3.93	5.89	0.48
Tensor-based (MMSE)	0.15	1.77	2.25	**0.17**
Tensor-based (ML)	1.27	3.48	5.52	0.96
Tensor-based (MAP)	0.15	3.55	5.10	0.18
Tensor-based SAT (ML)	1.27	3.48	5.52	0.96
Tensor-based SAT (MAP)	5.78	11.56	12.67	4.14
Tensor-based bilinear (MMSE)	**0.14**	**1.75**	**2.22**	0.18
Tensor-based bilinear (ML)	1.15	3.32	4.57	0.86

表 A.4: EER (using MV-PLDA) [%]: Set 2

	HS-HS	HS-DT	DT-HS	DT-DT
Tensor-based (MMSE)	1.88	5.93	6.32	1.92
Tensor-based (ML)	8.49	13.33	14.19	7.46
Tensor-based (MAP)	6.11	13.42	11.30	4.89
Tensor-based SAT (ML)	15.60	20.07	20.36	14.83
Tensor-based SAT (MAP)	22.63	26.37	29.63	23.64
Tensor-based bilinear (MMSE)	**1.44**	**4.38**	**3.85**	**1.13**
Tensor-based bilinear (ML)	9.49	13.89	13.87	7.77

表 A.5: EER (using PLDA) [%]: Set 3

	HS-HS	HS-DT	DT-HS	DT-DT
i-vector	0.31	2.51	3.39	0.35
EV-based (MMSE)	0.23	1.99	2.59	0.30
EV-based (ML)	0.60	2.76	4.01	0.45
EV-based (MAP)	0.29	3.40	4.96	0.26
EV-based SAT (ML)	0.43	3.50	5.29	0.32
EV-based SAT (MAP)	0.38	3.33	5.27	0.36
Tensor-based (MMSE)	0.09	**1.35**	1.72	**0.10**
Tensor-based (ML)	0.81	2.76	3.95	0.60
Tensor-based (MAP)	0.16	2.99	3.95	0.15
Tensor-based SAT (ML)	4.24	10.54	13.48	3.16
Tensor-based SAT (MAP)	4.17	10.07	11.69	3.07
Tensor-based bilinear (MMSE)	**0.08**	1.42	**1.66**	0.12
Tensor-based bilinear (ML)	0.74	2.64	3.83	0.60

表 A.6: EER (using MV-PLDA) [%]: Set 3

	HS-HS	HS-DT	DT-HS	DT-DT
Tensor-based (MMSE)	1.79	5.15	5.66	1.33
Tensor-based (ML)	9.84	15.22	15.13	8.61
Tensor-based (MAP)	5.63	12.83	10.24	3.98
Tensor-based SAT (ML)	15.12	19.27	19.33	12.88
Tensor-based SAT (MAP)	17.46	21.90	33.21	28.50
Tensor-based bilinear (MMSE)	**1.34**	**3.83**	**3.91**	**1.16**
Tensor-based bilinear (ML)	7.86	12.49	12.91	6.99