博士論文

話者空間基底を用いた特徴量分解とそれに基づくパラレルデータフリーDNN型声質変換

指導教員　峯松 信明 教授

東京大学大学院 工学系研究科
電気系工学専攻

37-157078　橋本 哲弥

概要

本研究では，話者性の柔軟な制御に向けての Deep Neural Network(DNN) による声質変換手法を提案する．DNN では，既存手法である Gaussian Mixture Models(GMM) よりも高精度の変換が可能であるという報告もされているが，DNN は各ノード・各レイヤーがどういった情報を扱い，どのような変換を行っているかが不明瞭であるため，GMM のようなパラメータ適応が難しく，柔軟な変換を実現することが難しいという問題がある．この問題を改善する足がかりとして，DNN による声質変換の処理を，浅い層 (入力層に近い層) における話者非依存な特徴量抽出と，深い層 (出力層に近い層) における話者依存の話者性の変換・生成に分離するために，1 つの話者非依存サブネットワークと複数の話者依存サブネットワークを持つ DNN を，複数の話者からなるパラレルコーパスによって学習することで，単一の入力層と複数の出力層を持つ DNN を構築した．提案手法によって構築した DNN に対して，学習に使用した話者と学習に使用していない未知話者に対しての変換精度を客観評価指標によって評価した結果，既存手法である GMM と通常の DNN を上回る変換精度が得られた．また，多対多声質変換において学習に用いるパラレルデータの量の削減は重要な課題であり，現在も多くの研究が行われている．そこで，より少量のデータでモデルの適応が可能かつ柔軟な多対多声質変換の枠組みとして，EVGMM と DNN を組み合わせた変換手法を提案した．さらに，より大規模なデータを用いた学習を可能とするため，全学習過程においてパラレルデータを必要としない枠組みを提案した．実験結果から，パラレルデータによる事前学習を行った際には既存手法を上回る変換精度が得られ，パラレルデータを一切使用しない場合も，話者性の変換に関してはパラレルデータを使用した手法とほぼ同等の変換結果であるという評価が得られた．加えて，DNN の各層に対して適応を行う Factorized Hidden Layers を用いた多対多声質変換を提案し，各層の適応パラメータを分析することで，DNN による実際の変換が上述した DNN 声質変換における仮定に沿っているかの確認を行った．

目次

図目次

表目次

第1章

序論

1.1　背景・目的

声質変換 (Voice Conversion) は入力されたある話者の発話を，言語情報を損なうことなく他の話者の音声に変換する技術のことを言う．変換対象としては音声を発する話者そのものや，音声に付加されている感情などが挙げられ，広義的には雑音除去も声質変換の一種と捉えることができる．応用としては，文章から音声を生成する Text-To-Speech (TTS) システムに対してより自由な音声出力を可能とするようなエンターテイメント的なものから，人工声帯から発せられる音声をより自然なものに変換するような医療的なものなど多くのアプリケーションに用いられている [1, 2]．声質変換は，入力音声と出力音声の特徴量空間上でのマッピングを構築するというタスクと考えられ，統計的な変換モデルによる実装がよく用いられている．コードブックマッピング [3] のような事例ベースの手法も提案されているものの，近年では，統計的手法の中でも Gaussian Mixture Models (GMM)，Artificial Neural Networks (ANN) の 2 手法が広く研究されている [4][5]．これらの手法で変換モデルを学習する際には，パラレルコーパスと呼ばれる，変換元の話者と変換先の話者による同一文の読み上げ音声データが必要となる．

しかし，この方法で構成した変換モデルは，学習に用いていない話者に対しては変換精度が低く，学習に使用した特定の話者間に対してしか用いることができない．実用的な場面を考えた場合，新たな話者を入力及び出力としようとする度に，特定の文章の読み上げ音声を収録するのはコストが大きい．また，医療的な応用を考えた際には，新たに大量の音声を収録するのが難しい場合も考えられる．そのため，入力・出力話者に対して柔軟な声質変換は非常に重要なタスクであり，現在も多く研究されている．このような入出力話者に関してより少量の音声データで変換を可能とする枠組みを多対多声質変換と呼ぶ．

GMM においてはパラメータ適応と呼ばれる，一部のパラメータのみを話者毎に更新する手法による一対多および多対一声質変換が提案されている [6, 7, 8]．これらの手法においては，予めパラレルデータが存在している大量の話者によって学習を行うことによって，話者に依存しない一般的な事前知識を獲得し，それを基に少量のパラメータの更新によって変換の対象とする話者を変更するという枠組みが用いられている．一方で，ANN の応用手法であり，近年多分野において研究が行われている Deep Neural Netwotk (DNN) では，GMM よりも高精度の変換が可能であるという報告もされている [9, 10]．しかし，DNN は各ノード・各レイヤーがどういった情報を扱い，どのような変換を行っているかが不明瞭であるため，GMM のようなパラメータ適応が難しく，柔軟な変換を実現することが難しいという問題がある．GMM における Eigenvoice GMM (EVGMM)[11] やテンソル空間を用いた声質変換 [12] のように，話者性の柔軟な制御を可能とするためには，DNN による声質変換においても複数話者からの事前知識の獲得と，それを用いた話者依存処理と非依存処理の分離が重要と考えられる．

そこで本研究では，DNN を用いた声質変換において複数話者を学習に用いることの有用性を示しつつ，より多対多声質変換に適した形の DNN の実装することで，より少量のデータによる高精度な声質変換の実現を目的とする．

本研究では初めに多言語認識における松田らの手法 [13] を参考とし，DNN による声質

変換の処理を，浅い層における話者非依存な特徴量抽出と，深い層における話者依存の話者性の変換・生成に分離することを試み，それによる変換精度の変化を実験的に検討した．松田らの手法では，ネットワークの浅い層 (入力に近い層) では言語に非依存な特徴量抽出のような処理が行われており，ネットワークの深い層 (出力に近い層) では言語に依存した識別のような処理が行われているという仮定を置いている．この仮定を基に，ネットワークの浅い層を 1 つの言語非依存サブネットワークによって構成し，深い層を複数の言語依存サブネットワークによって構築することで，浅い層に複数の言語のデータを，深い層に認識する言語に対応したデータをそれぞれ学習させることで，認識率の改善を行っている．本研究では，この松田らの手法を参考に，DNN に対して変換先の話者毎のサブネットワークを導入した，多対一型の変換手法を提案している．この手法を通して，複数話者によって学習を行った DNN の有用性を示すと共に，松田らの仮定が変換結果にどのように反映されるかの確認を行った．

一方で，サブネットワークを用いた提案手法では変換先の話者は学習データ中に存在する話者しか用いることができず，複数話者とのパラレルデータがなければ新しい話者への変換を行うことができない．そこで，より少量のデータで適応可能な多対多 DNN 声質変換手法として，固有声 GMM の枠組みを用いた話者空間基底への分解に基づく手法を提案する．本手法では，初めに EVGMM によって音声を「平均話者」と「話者基底成分」に相当する特徴量へと変換し，それらと元々の特徴量の組み合わせを擬似的なパラレルコーパスの様に使用することで，DNN によって元々の特徴量から「平均話者」と「話者基底成分」への分解を行うような変換を学習する．最終的な出力特徴量はこの分解された特徴量と，出力話者固有の重みの積によって表現される．すなわち，本手法は複数話者から獲得する事前知識を，基底と平均という形にした上で DNN の学習を行った手法である．

次に，多くの多対多声質変換において事前学習ではパラレルデータを用いているという点に着目した．事前学習におけるパラレルデータ制約を無くすことができれば，任意の音声データを用いた初期モデル構築が可能となり，ビッグデータを用いた学習への応用なども考えられる．そこで，上述した提案手法において基底及び平均成分の導出をパラレルデータを用いずに行う手法の提案を行った．

また，Factorized Hidden Layers (FHL) を用いて DNN の各層に対して話者適応を行う声質変換を提案し，話者空間基底を用いた多対多声質変換手法との比較実験を行うことで，話者空間基底を用いることの有用性を示した．加えて，FHL による各層の適応パラメータを可視化することで，DNN 声質変換の分析を試みた．

1.2　本稿の構成

本稿は，全 8 章で構成される．2 章では声質変換の基本的な考え方と統計的声質変換において一般的に用いられている特徴量，データの前処理について示す．それに加えて声質変換で用いられる代表的な統計的手法として GMM，DNN について示し，それぞれについて声質変換以外でも用いられているパラメータ適応手法を挙げる．3 章では，既存の多対多声質変換および提案手法においても使用する話者表現ベクトルについて示す．また，学

習過程において一切パラレルデータを必要としない声質変換の既存手法について述べる．4 章では，提案手法であるサブネットワークを用いた DNN 声質変換の着想・目的・具体的な手法について説明し，提案手法の有用性を示すために行った DNN の pre-training に関する予備的実験，学習データ中の話者に関する提案手法と既存手法 (GMM，DNN) の間での変換精度の比較，未知話者のデータに対する提案手法と既存手法の間での精度比較を行い，結果について示す．5 章では，DNN と EVGMM を組み合わせた話者空間基底に基づく多対多声質変換を提案し，実験によって有用性を示す．そして，6 章において完全パラレルデータフリーへと拡張を行った提案手法とその着想及び予備実験について示し，比較実験の結果を示す．7 章では FHL を用いた層単位での適応が可能な多対多声質変換を提案し，5 章の提案手法と比較することで話者空間基底を用いることの有用性を示す．それに加えて，DNN の適応パラメータを可視化することで，各層において入出力話者の適応がどのように行われているかを確認する．最後に 8 章で研究全体のまとめと今後の課題・応用について述べる．

第2章

一対一声質変換及びその拡張

2.1　声質変換の概要

近年，Text-to-Speech システムや自動翻訳システムの発展によって，合成，及び変換音声に対して多様な声質が求められている．医療の分野においても，喉の障害などによってはっきりとした発音ができなくなった人や，人工声帯による発話を行っている人の音声をその人本人の声であり且つ自然な発話となるような音声合成器が必要とされている．このような需要に応える技術として声質変換が存在する．

声質変換は，入力された合成音声や自然音声に対して，言語内容を損なうこと無くその音声の話者性 (個人性，性別など) や感情などの発話様式を他のものに変換する技術のことをいう．音声の声質を決定する要因としては，声道特性と音源特性の 2 つがある．音源特性は声帯の開閉によって発生する振動から生成される音源の特性のことをいい，音声に置ける韻律 (イントネーション) に寄与する．一方で，声道特性は声帯から発せられる音源に対して口腔や鼻腔，舌の動きによって韻律を付加する特性のことをいう．声質変換では，音声波形からこれらの特性を特徴量として抽出し，変換前の話者 (入力話者) の特徴量から目的とする変換先の話者 (出力話者) の特徴量への変換を行う．すなわち，声質変換は入力音声と出力音声の特徴量空間上でのマッピングを構築するというタスクと考えられる．そのため，声質変換では入力話者と出力話者によって同一の文章を発声してもらい，その特徴量系列の間で変換モデルを学習する統計的手法が用いられている．

2.2　音声特徴量

2.2.1　ソースフィルタモデル

音声の分析は一般的にソースフィルタモデルにしたがって行われる．ソースフィルタモデルは, 音声を声帯の振動 (ソース) による音源特性と声道の形状特性 (フィルタ) による声道特性の 2 つに分けて考えるものであり, 声帯による音源を $s(t)$, 声道の形状特性を $v(t)$ とすると音声 $x(t)$ は以下の式のような関係で表される.

$$|X(\omega)| = |S(\omega)||V(\omega)| \tag{2.1}$$

ここで, $X(\omega)$, $S(\omega)$, $V(\omega)$ は音声, 音源, 声道特性のフーリエ変換である. 音声における声質は，主に声道特性によって表されるものであるため, 声質変換ではこの声道特性をよく表す特徴量を主な特徴量として用いる.

2.2.2　基本周波数 (F0)

音声信号において，母音は周期的な波形を持つ．この周期的な波形の 1 周期を取り出して周期関数と考えることで，以下の式のようなフーリエ級数に展開することができる．

$$x(t) = \sum_{n=0}^{\infty} A_n \cos(2\pi v F_0 t + \theta_n) \tag{2.2}$$

式 (2.2) 中の F_0 を基本周波数という．この基本周波数は音声の高さ (ピッチ) に寄与する値であり，声帯の振動の周期を表す．声質変換において基本周波数の変換は，ソース話者とターゲット話者の声の高さの変換となる．

2.2.3　ケプストラム

一般的に音声波形は初めに得られる時間系列の波形信号から，フーリエ変換によって周波数領域の信号に変換して扱われる．この周波数スペクトルの振幅を絶対値化したものを周波数パワースペクトル (power spectrum) という．

音声信号のパワースペクトルは音声に含まれる韻律情報を担う重要な変量であり，音源特性と音源位置，声道特性，口唇からの放射特性などの総体として出力される特性である．音声信号におけるパワースペクトルの包絡抽出は概ね声道特性の抽出に相当する．また，抽出される包絡は，包絡に対するモデルの違いと元の波形のフーリエ・スペクトルに対する誤差尺度の違いによって僅かに異なるものとなる．

ここにケプストラムという尺度を導入する．ケプストラムは信号波形のパワースペクトルの対数のフーリエ変換として定義される値であり，複数の信号が畳み込みのような形で結合されているような信号の解析に用いられる．ここで，ソースフィルタモデルを考えると，音声 $x(t)$ は声帯による音源 $s(t)$ と声道の $v(t)$ の畳み込みによって以下の式で表現される．

$$x(t) = \int_{-\infty}^{+\infty} s(\tau)v(t-\tau)d\tau \tag{2.3}$$

それぞれのフーリエ変換を考えると，

$$|X(\omega)| = |S(\omega)||V(\omega)| \tag{2.4}$$

ここで対数を取ることで，

$$\log|X(\omega)| = \log|S(\omega)| + \log|V(\omega)| \tag{2.5}$$

フーリエ変換を $\mathcal{F}[\cdot]$ とすると，ケプストラムは以下のように表される．

$$\mathcal{F}[\log|X(\omega)|] = \mathcal{F}[\log|S(\omega)|] + \mathcal{F}[\log|V(\omega)|] \tag{2.6}$$

式 (2.5) のように，対数パワースペクトルでは声道成分を表す $V(\omega)$ に音源特性，すなわち基本周波数成分を表す $S(\omega)$ が足し合わされており，実際のスペクトル上でも声道成分の連続的なスペクトル上に基本周波数による周期的な離散値が重畳しているような形が見られる．この対数パワースペクトルにフーリエ変換を行うことで，連続的なスペクトルを持っていた声道成分は低次のケプストラムとして現れ，離散的なスペクトルを持っていた基本周波数はケプストラム上の対応する周期の位置にピークとして現れる．また，ケプストラムの 0 次元には元の音声波形のパワー成分となる．声質変換ではこのケプストラムが主な変換の対象として扱われ，基本周波数に関してはソース話者とターゲット話者の平均と分散を用いた線形変換で簡易的に行っている研究も多い．

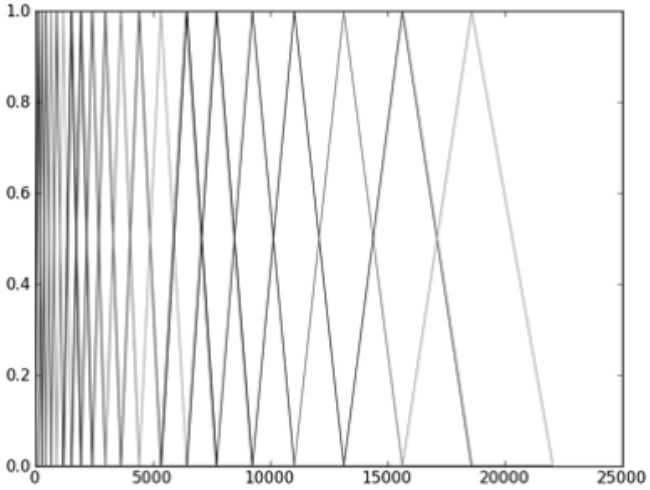

図 2.1: メル尺度に基づく帯域フィルター

2.2.4　メルケプストラム

式 (2.6) で表されるケプストラムは時間領域で等間隔にサンプリングされた標本値を用いて計算されたものであり，周波数軸尺度は線形である．しかし，人間の聴感覚特性は低周波数の音声に対しては高い分解能，高周波数の音声に対しては低い分解能を持ち，全体的には対数に近い，メル尺度 (mel-scale) と呼ばれる周波数感度となっている．そこで，スペクトルを扱う際にも実際の人間の聴覚特性を表すメル尺度を導入することで，人間の聴覚にとって重要な周波数成分をより重点的に扱うことが考えられる．このメル尺度を導入して計算されたケプストラムのことをメルケプストラムという．

メルケプストラムの計算方法としては，スペクトルを周波数軸上でメル尺度で等間隔になるように再サンプリングし，その標本値を用いてスペクトルを再推定するというものがある．この方法では音声のスペクトルに対し，図 2.1 のような周波数軸上でメル尺度において等間隔となるような帯域フィルタ群を用いてフィルタリングを行い，その出力に対して離散フーリエ変換を行うことでメルケプストラムを得る．

2.3　統計的声質変換

これまでに示したような，声道特徴量および音源特徴量を変化させることで音声の変換を行うことができる．例として，声道特徴量であるスペクトル包絡を周波数軸上で伸ばすことで音声を子供のような高い音に変化させたり，縮めることで音声を大人の男性のような太い低い声にすることができる．音源特徴量に関しても，基本周波数を高くすれば高い声，低くすれば低い声にでき，非周期成分を大きくすればかすれた声に変換することができる．しかし，これらの変換を話者毎にヒューリスティックに決定するのは，発声に含ま

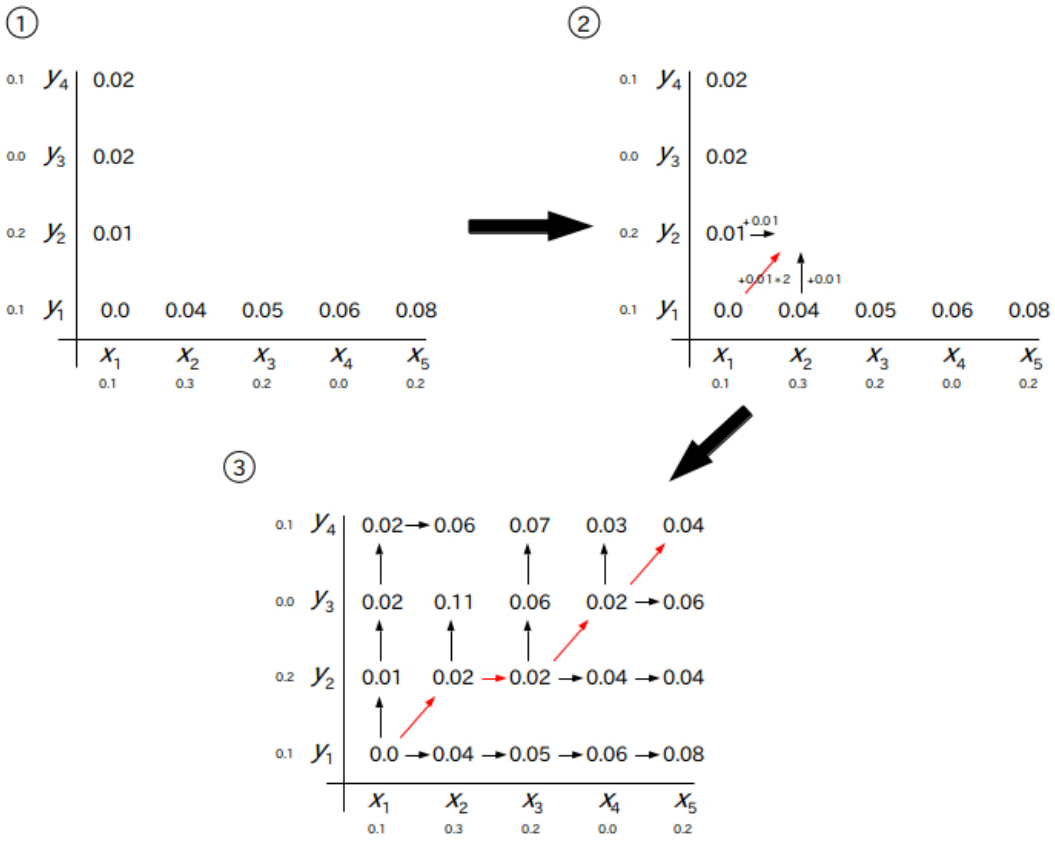

図 2.2: DTW による系列アラインメントの概要

れる音素の数及び組み合わせを考えると現実的ではない．そのため，声質変換では一般的に統計的手法を用いた変換処理が用いられている．

統計的声質変換では，ソース話者の特徴量ベクトルを x，ターゲット話者の特徴量ベクトルを y としたとき，x が与えられた時の y の事後確率 $P(y \mid x)$ をモデル化し，生成モデルによって x から y への変換を行う手法と，変換モデルによって x から y への変換を直接行うものとがある．どちらのモデルにおいても，モデルのパラメータの学習には一般的にソース話者とターゲット話者による同一の言語情報を発話した音声データ (パラレルコーパス) を学習データとして使用する．特徴量の変換にはコードブックマッピングを用いたもの [3] なども存在するが，現在は Neural Network を用いたもの [5] や，Gaussian Mixture Models を用いたものが主流となっている [12].

2.4　パラレルコーパス間のアラインメント

一般的な統計的声質変換では，入力話者と出力話者による同一の言語情報を発話したパラレルコーパスによってモデルの学習を行う．しかし，同一の発話内容であっても話者毎に発話速度は異なるため，入力話者の特徴量ベクトル群と出力話者の特徴量ベクトル群は殆

どの場合同じデータ数にはならない．そのため，入力話者と出力話者の特徴量間でフレームのアラインメントを考える必要がある．このアラインメントには Dynamic Time Warping (DTW) と呼ばれる動的計画法によるマッチング手法が用いられる．

元々の DTW は長さの異なる 2 つの系列データ間の特徴量を照合し，距離を求めることで系列データの類似度を測るものであり，音声認識などに用いられていた．一方，声質変換を行う際のアラインメントを取るための DTW では，2 つの系列データ (音響特徴量) 間に何らかの距離尺度を導入し，その系列データ間の距離が最小となるように一方の各フレームがもう一方のどのフレームに対応するかを決定する．

図 2.2 に DTW の概要を示す．図 2.2 では簡単のため，特徴量を 1 次元としている．初めに 2 つの系列 (それぞれ長さ m，n とする) をそれぞれ横軸，縦軸とし，$m * n$ 行列の空行列を 2 つ (それぞれ H, C とする) 作る．次に，系列間の全てのフレーム間の距離 D_{ij} を計算し，保持しておく．$(i = 1, 2, ..., m,\ j = 1, 2, ..., n)$ この D_{ij} を基に，$(1, 1)$ から順に，以下の式に従って隣接する座標から各座標までの最小距離を求め, H_{ij} に保存していく．

$$H_{ij} = \min \begin{cases} H_{i-1,\ j-1} & + & 2D_{ij} \\ H_{i-1,\ j} & + & D_{ij} \\ H_{i,\ j-1} & + & D_{ij} \end{cases}$$

この際，どの式に従って H_{ij} を計算したかを C_ij に保存しておく (図 2.7 の座標中の矢印に相当する)．式 (2.7) 中の $H_{i-1,\ j-1}$ は両系列を伸縮無しに 1 フレームを対応させ，$H_{i-1,\ j}$，$H_{i,\ j-1}$ はどちらかの系列の 1 フレームに複数のフレームを対応させることを意味する．また，$2D_{ij}$ では経路毎に通過する格子点の数が異なってくるため，フレーム間の距離に重みを掛けることで，通過する格子点の数を実効的に等しくしている．全ての座標の H および C を計算した後，座標 (m, n) から C に保持されている移動に従って座標 $(0, 0)$ までの経路を辿ることで，2 系列のアラインメントを得る．

DTW の手順をまとめると以下の様になる．

1. 長さ m，長さ n である 2 つの系列を x 軸・y 軸として置く
2. 2 つの系列の全フレーム間の距離 D_{ij} を計算する $(i = 1, 2, ..., m,\ j = 1, 2, ..., n)$
3. $(x, y) = (1, 1)$ の点を始点とし，$x = 1$ または $y = 1$ である座標の距離とその座標への経路重みから最短経路と累積距離を計算する
4. (3) を x と y を大きくしながら全座標が埋まるまで繰り返す
5. 終点 (座標 (m, n)) から保存しておいた経路を辿ることによって最短経路を求める

2.5　コードブックマッピング

統計的声質変換の初期の 1 手法としてベクトル量子化によるコードブックマッピング法を用いたものがある [3]．この手法は，コードブックのマッピングを生成する学習ステップと，そのマップに従って音声の変換を行う変換ステップという 2 つのステップから構成される．

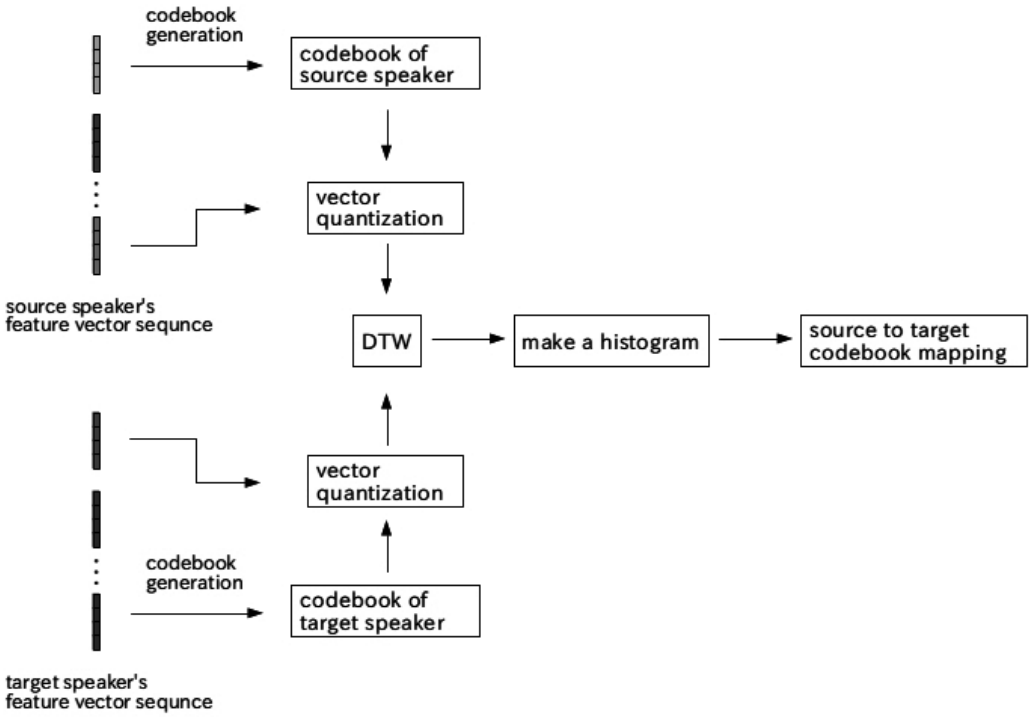

図 2.3: コードブックマッピングにおける学習ステップ

ベクトル量子化とは，入力ベクトル群にクラスタリングを行った後，各クラスタに含まれるベクトルからそのクラスタのセントロイドを計算し，入力ベクトル群をそのセントロイドで置換するというものであり，元々は情報圧縮などの分野に用いられていた．このときの各クラスタに宛てがわれるセントロイド群をコードブックと言う．コードブックマッピングによる声質変換では，ベクトル量子化における入力ベクトル群を特定話者の音響特徴量とし，そのコードブックが特定話者の個人性を表すと考えている．以下に，その具体的な処理を示す．

初めに，学習ステップについて述べる．学習ステップでは，2 話者の特徴量から張られるベクトル空間を表すコードブックを生成し，その間のマッピング関数を計算する．このマッピング関数をコードブックマッピングという．2 話者の音声特徴量からコードブックマッピングを生成する過程を図 2.3 に示す．詳細な処理は以下のようになる．

1. 2 話者 (入力話者，出力話者) による同一発話内容からなる単語コーパス (パラレルコーパス) を基に各フレーム毎の特徴量にベクトル量子化を行う
2. コーパス中の各単語に対応する 2 話者の量子化ベクトルに対して DTW によってアラインメントを取る
3. 量子化ベクトルのアラインメントから 2 話者間の対応関係をヒストグラムとして求める
4. 2 話者間の量子化ベクトルのヒストグラムを重み付けと考え 2 話者間の量子化ベクトルのマッピング (コードブックマッピング) を取り，出力話者のコードブックの線形組

み合わせで入力話者のコードブックを表現する
5. (2)，(3)，(4) の処理を精度が十分になるまで繰り返す
6. 基本周波数及びパワーに関してはスカラー量子化を行い，同様にコードブックマッピングを学習して求める

次に，変換ステップについて示す．変換ステップでは，生成したコードブックマッピングに従って入力話者の発話を出力話者のものに変換する．概要を図 2.3 に示す．具体的な処理は以下の様になる．

1. 入力として与えられた入力話者の音声特徴量群 (スペクトル，基本周波数，パワー) を線形予測分析によってクラスタリング
2. クラスタリングされた特徴量をコードブックに置換する
3. コードブックに置換された入力話者の音声特徴量群をコードブックマッピングに従って出力話者のコードブックにデコーディングする

この処理によって得られた変換後の音声特徴量を合成することで，出力話者の音声への変換を実現する．

コードブックマッピングによる声質変換の問題点としては，量子化ベクトルを用いて入力話者，及び出力話者の特徴量空間を離散的に表現してしまうために，最終的に合成される変換音声が不連続なものになってしまうというものがある．この点を改善する手法としてファジーベクトル量子化と差分ベクトルに基づくコードマッピング法 [14] も提案されている．しかし，現在は連続的かつより高精度な変換手法として Gaussian Mixuture Models (混合正規分布モデル) や Artificial Neural Network を用いた変換手法が主に扱われている．

2.6　Gaussian Mixture Models を用いた声質変換

Gaussian Mixture Models (GMM) は，未知の確率変数ベクトルの確率密度関数を，混合ガウス分布として推定するモデルである．図 2.4 に簡略化した混合ガウス分布を示す．

D 次元の確率変数ベクトル $\boldsymbol{x}$ が与えられたとき，GMM ではその確率密度関数 $p(\boldsymbol{x})$ を (2.7) 式として表す．

$$p(\boldsymbol{x}) = \sum_{i=1}^{M} P(w_i)\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_i^x, \boldsymbol{\Sigma}_i^{xx}) \tag{2.7}$$

ここで，M は混合数，$P(w_i)$ は各ガウス分布の重み $\boldsymbol{\mu}, \Sigma$ はそれぞれ i 番目のガウス分布における平均ベクトルと分散共分散行列を表し，$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ は以下で表されるガウス分布である．

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

このモデルに対して，訓練データとして入力話者の特徴量の時間系列 $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n]$ と出力話者の特徴量の時間系列 $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_n]$ のパラレルデータを用い，$\boldsymbol{X}$ と $\boldsymbol{Y}$ の各

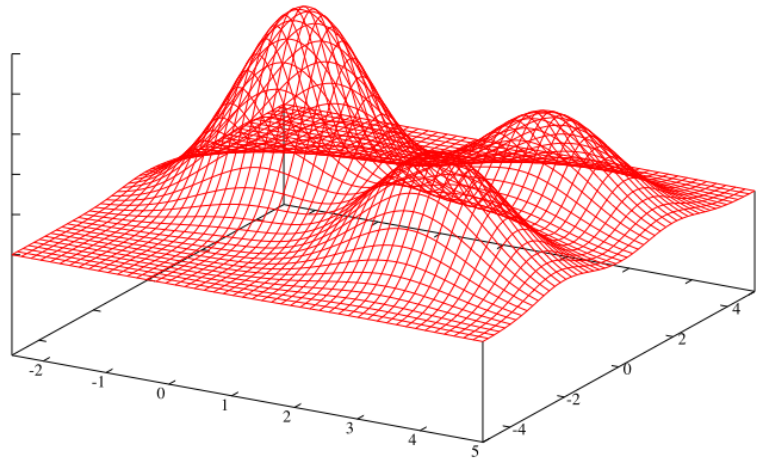

図 2.4: 混合ガウス分布の概略図

要素の結合ベクトル $\boldsymbol{z}_i = [\boldsymbol{x}_i^\top, \boldsymbol{y}_i^\top]^\top$ の時間系列 $\boldsymbol{Z} = [\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_n]$ の確率密度関数を考え，EM アルゴリズムによって二乗誤差平均が最小となる適切なパラメータを推定する．このとき，入力話者の特徴量 x_k を出力話者の特徴量 y に変換する関数は (2.8) 式のように表される．

$$\begin{aligned} F(\boldsymbol{x}_k) &= E(\boldsymbol{y} \mid \boldsymbol{x}_k) \\ &= \sum_{i=1}^{M} P(w_i \mid x_k) \left[\boldsymbol{\mu}_i^y + \boldsymbol{\Sigma}_i^{yx} \boldsymbol{\Sigma}_i^{xx^{-1}} (\boldsymbol{x}_k - \boldsymbol{\mu}_i^x) \right] \end{aligned} \tag{2.8}$$

ここで，$E(.)$ は期待値を表し，条件付き確率 $P(w_i \mid \boldsymbol{x}_k)$ は式 (2.9) で与えられる．

$$P(w_i \mid \boldsymbol{x}_k) = \frac{P(w_i)\mathcal{N}(\boldsymbol{x}_k; \boldsymbol{\mu}_i^x, \boldsymbol{\Sigma}_i^{xx})}{\sum_{j=1}^{M} P(w_j)\mathcal{N}(\boldsymbol{x}_k; \boldsymbol{\mu}_j^x, \boldsymbol{\Sigma}_j^{xx})} \tag{2.9}$$

EM アルゴリズムによる学習においては，時間対応の取れた $\boldsymbol{x}_k$ と $\boldsymbol{y}_k$ の組が必要となるため，パラレルコーパスが必要となる．

2.6.1　EM アルゴリズム

EM アルゴリズムは Expectation Step(E-step) と Maximization Step(M-step) の 2 ステップを繰り返すことで最尤推定を行うアルゴリズムをいう [15]．$\boldsymbol{X}$ と $\boldsymbol{Y}$ の結合ベクトルの時間系列 $\boldsymbol{Z} = [\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_n]$ の確率密度関数を計算するのに必要なパラメータは各ガウス分布 w_i に対してのガウス分布の重み $P(w_i)$, 平均ベクトル $\boldsymbol{\mu}_i^z$，共分散行列 $\boldsymbol{\Sigma}_i^{zz}$ であり，k-means などによって定めた初期値に従ってこれらのパラメータを推定する．

E-step では，現在のパラメータの値から式 (2.10) の条件付確率確率を計算する．ここで t

は反復回数を示す.

$$P^{(t)}(w_i \mid \boldsymbol{z}_k) = \frac{P^{(t)}(w_i)\mathcal{N}(\boldsymbol{z}_k; \boldsymbol{\mu}_i^{(t)z}, \boldsymbol{\Sigma}_i^{(t)zz})}{\sum_{j=1}^{M} P^{(t)}(w_j)\mathcal{N}(\boldsymbol{z}_k; \boldsymbol{\mu}_j^{(t)z}, \boldsymbol{\Sigma}_j^{(t)zz})} \tag{2.10}$$

M-step では，E-step によって得られた条件付確率 $P^{(t)}(w_i \mid \boldsymbol{z}_k)$ によって GMM のパラメータを再推定する．各パラメータは式 (2.11)(2.12)(2.13) によって計算される.

$$P^{(t+1)}(w_i) = \frac{1}{n}\sum_{k=1}^{n} P^{(t)}(w_i \mid \boldsymbol{z}_k) \tag{2.11}$$

$$\boldsymbol{\mu}_i^{(t+1)z} = \frac{\sum_{k=1}^{n} P^{(t)}(w_i \mid \boldsymbol{z}_k)\boldsymbol{z}_k}{\sum_{k=1}^{n} P^{(t)}(w_i \mid \boldsymbol{z}_k)} \tag{2.12}$$

$$\Sigma_i^{(t+1)zz} = \frac{\sum_{k=1}^{n} P^{(t)}(w_i \mid \boldsymbol{z}_k)(\boldsymbol{z}_k - \boldsymbol{\mu}_i^{(t+1)z})(\boldsymbol{z}_k - \boldsymbol{\mu}_i^{(t+1)z})^\top}{\sum_{k=1}^{n} P^{(t)}(w_i \mid \boldsymbol{z}_k)} \tag{2.13}$$

この E-step と M-step を各パラメータおよび条件付確率が収束するまで繰り返す．最終的な結果から，入力特徴量 x_k を特徴量 y に変換する (2.8) 式,(2.9) 式の計算において必要となる共分散行列 $\boldsymbol{\Sigma}_i^{xx}$，$\boldsymbol{\Sigma}_i^{yx}$，平均ベクトル $\boldsymbol{\mu}_i^x$，$\boldsymbol{\mu}_i^y$ を推定されたパラメータから以下の式によって得る.

$$\boldsymbol{\Sigma}_i^{zz} = \begin{bmatrix} \boldsymbol{\Sigma}_i^{xx} & \boldsymbol{\Sigma}_i^{xy} \\ \boldsymbol{\Sigma}_i^{yx} & \boldsymbol{\Sigma}_i^{yy} \end{bmatrix} \tag{2.14}$$

$$\boldsymbol{\mu}_i^z = \begin{bmatrix} \boldsymbol{\mu}_i^x \\ \boldsymbol{\mu}_i^y \end{bmatrix} \tag{2.15}$$

2.7　Gaussian Mixture Models における話者適応手法

GMM による声質変換では学習の際に入力話者と出力話者による時間対応の取れたパラレルコーパスが必要となり，他の話者を入出力話者とするモデルを作ろうとする度に大規模なコーパスが必要となってしまうという問題がある．このような問題を改善することを目的とした, すなわち声質変換を話者適応的なシステムに改善することを目的とした手法が提案されている [7]．ここではその中の代表的な手法を挙げ, その概要を示す.

2.7.1　parameter adaptation

Parameter Adaptation は，新たな入力話者から出力話者への変換モデルを学習する際に, 新しくモデルを作り直すのではなく, 他の話者間のモデルに対して新たな入出力話者のデータでモデルの適応を行うという手法である. 手法の概要を図 2.5 に示す.

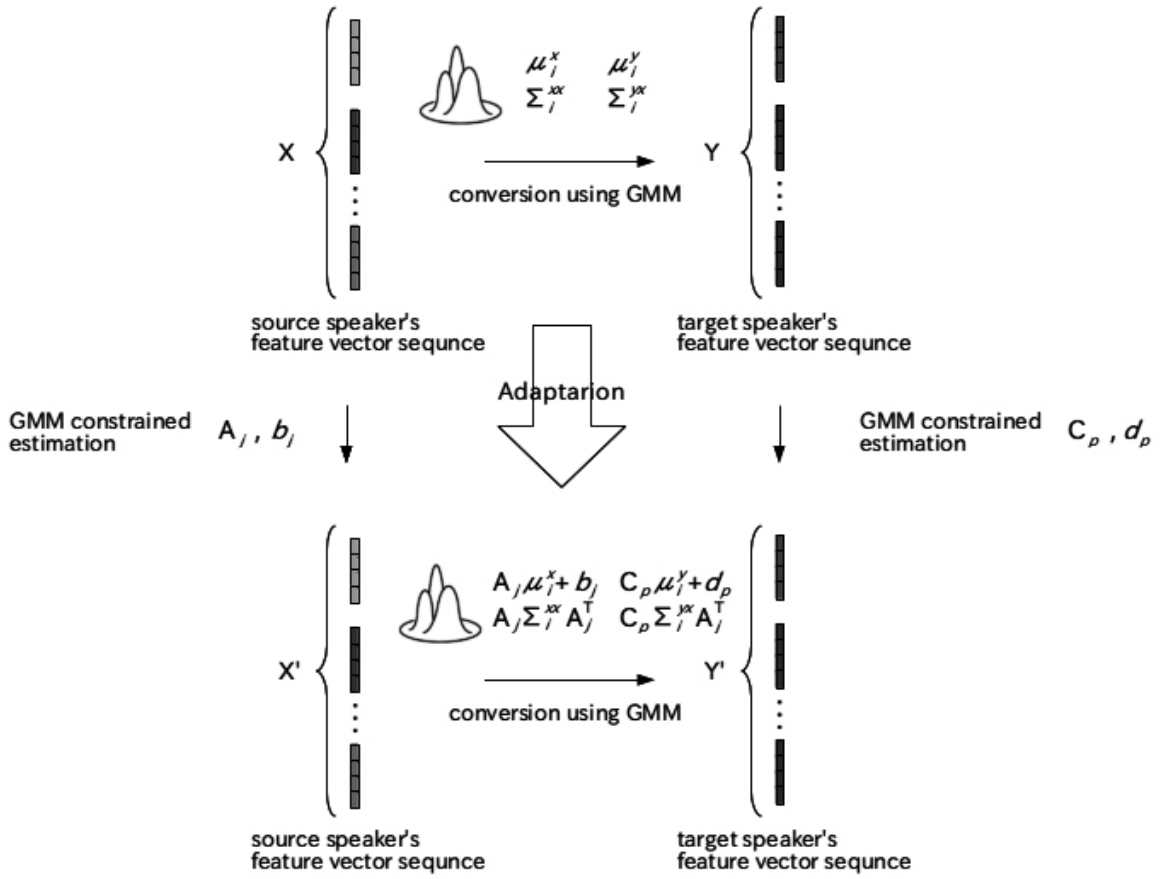

図 2.5: パラメータ適応の概要

特定話者 X，Y 間の GMM から新たな話者 X'，Y' の GMM のパラメータを推定する場合を考える．基の GMM の入力話者の確率変数ベクトルを $\boldsymbol{x}$ としたとき，対象話者の確率変数ベクトル $\boldsymbol{x}'$ を $\boldsymbol{x}$ の確率的線形変換として (2.16) 式のように定義する．

$$\boldsymbol{x}' = \begin{cases} \mathbf{A}_1\boldsymbol{x} + \boldsymbol{b}_1 & \text{with probability } p(\lambda_1|w_i) \\ \mathbf{A}_2\boldsymbol{x} + \boldsymbol{b}_2 & \text{with probability } p(\lambda_2|w_i) \\ \vdots & \vdots \\ \mathbf{A}_N\boldsymbol{x} + \boldsymbol{b}_N & \text{with probability } p(\lambda_N|w_i) \end{cases} \tag{2.16}$$

λ_j は x の各ガウス分布 w_i に対応した変換を表し，$p(\lambda_j \mid w_i)$ は (2.17) 式を満たす．

$$\sum_{j=1}^{N} p(\lambda_j \mid w_i) = 1, \quad i = 1, ..., M. \tag{2.17}$$

また，M は適応元の GMM の混合数，$\mathbf{A}_j$ は $K * K$ 行列 (K は x の次元) である．出力話者

$\boldsymbol{y}$, $\boldsymbol{y}'$ に関しても同様に (2.18) 式のように定義する．

$$\boldsymbol{y}' = \begin{cases} \mathbf{C}_1\boldsymbol{y} + \boldsymbol{d}_1 & \text{with probability } p(\kappa_1|w_i) \\ \mathbf{C}_2\boldsymbol{y} + \boldsymbol{d}_2 & \text{with probability } p(\kappa_2|w_i) \\ \vdots & \vdots \\ \mathbf{C}_L\boldsymbol{y} + \boldsymbol{d}_L & \text{with probability } p(\kappa_L|w_i) \end{cases} \tag{2.18}$$

$$\sum_{\rho=1}^{L} p(\kappa_j \mid w_i) = 1, \quad i = 1, ..., M. \tag{2.19}$$

式 (2.16) から，x' の確率密度関数は w_i，λ_j が与えられたとき，

$$g(\boldsymbol{x}' \mid w_i, \lambda_j) = \mathcal{N}(\boldsymbol{x}'; \mathbf{A}_j\boldsymbol{\mu}_i^x + \boldsymbol{b}_j, \mathbf{A}_j\boldsymbol{\Sigma}_i^{xx}\mathbf{A}_j^\top) \tag{2.20}$$

$$g(\boldsymbol{x}') = \sum_{i=1}^{M}\sum_{j=1}^{N} p(w_i)p(\lambda_j \mid w_i)\mathcal{N}(\boldsymbol{x}'; \mathbf{A}_j\boldsymbol{\mu}_i^x + \boldsymbol{b}_j, \mathbf{A}_j\boldsymbol{\Sigma}_i^{xx}\mathbf{A}_j^\top) \tag{2.21}$$

となり，混合数 $M * N$ の GMM と考えることができる．そのため，式 (2.16), 式 (2.17), 式 (2.18), 式 (2.19) における未知パラメータ $\mathbf{A}_j$，$\mathbf{C}_\rho$，$\boldsymbol{b}_j$，$\boldsymbol{d}_\rho$ は $\boldsymbol{x}$ と $\boldsymbol{y}$ のパラレルコーパスによる GMM を基に，$\boldsymbol{x}'$ と $\boldsymbol{y}'$ の非パラレルコーパスから，EM アルゴリズムによって推定することができる．

t 回目の試行における E-step では，以下の式でパラメータを計算する．

$$n_{ij}^{(t)} = \sum_{k=1}^{n} p^{(t)}(w_i \mid \boldsymbol{x}'_k)p^{(t)}(\lambda_j \mid \boldsymbol{x}'_k, w_i) \tag{2.22}$$

$$\boldsymbol{\mu}_{ij}^{(t)x'} = \frac{1}{n_{ij}^{(t)}}\sum_{k=1}^{n} p^{(t)}(w_i \mid \boldsymbol{x}'_k)p^{(t)}(\lambda_j \mid \boldsymbol{x}'_k, w_i)\boldsymbol{x}'_k \tag{2.23}$$

$$\Sigma_{ij}^{(t)x'x'} = \frac{1}{n_{ij}^{(t)}}\sum_{k=1}^{n} p^{(t)}(w_i \mid \boldsymbol{x}'_k)p^{(t)}(\lambda_j \mid \boldsymbol{x}'_k, w_i) \cdot(\boldsymbol{x}'_k - \boldsymbol{\mu}_{ij}^{(t)x'})(\boldsymbol{x}'_k - \boldsymbol{\mu}_{ij}^{(t)x'})^\top \tag{2.24}$$

このとき，$p^{(t)}(w_i \mid \boldsymbol{x}'_k)$, $p^{(t)}(\lambda_j \mid \boldsymbol{x}'_k, w_i)$ は以下のようになる．

$$p^{(t)}(w_i \mid \boldsymbol{x}'_k) = \frac{p(w_i)\sum_{j=1}^{N} p^{(t)}(\lambda_j|w_i)g^{(t)}(\boldsymbol{x}'_k|w_i,\lambda_j)}{\sum_{i=1}^{M}\sum_{j=1}^{N} p(w_i)p^{(t)}(\lambda_j|w_i)g^{(t)}(\boldsymbol{x}'_k|w_i,\lambda_j)} \tag{2.25}$$

$$p^{(t)}(\lambda_j \mid \boldsymbol{x}'_k, w_i) = \frac{p^{(t)}(\lambda_j|w_i)g^{(t)}(\boldsymbol{x}'_k|w_i,\lambda_j)}{\sum_{j=1}^{N} p^{(t)}(\lambda_j|w_i)g^{(t)}(\boldsymbol{x}'_k|w_i,\lambda_j)} \tag{2.26}$$

同様に，M-step では以下の式でパラメータを計算する．

$$p^{(t+1)}(\lambda_j \mid w_i) = \frac{n_{ij}^{(t)}}{\sum_{j=1}^{N} n_{ij}^{(t)}} \tag{2.27}$$

$$\sum_{i=1}^{M} n_{ij}^{(t)}\{\mathbf{A_j^{(t+1)}}-\mathbf{\Sigma}_i^{xx^{-1}}[\mathbf{A_j^{(t+1)^{-1}}}(\boldsymbol{\mu}_{ij}^{(t)x'}-\boldsymbol{b}_j^{(t+1)})-\boldsymbol{\mu}_i^x](\boldsymbol{\mu}_{ij}^{(t)x'}-\boldsymbol{b}_j^{(t+1)})^\top-\mathbf{\Sigma}_i^{xx^{-1}}\mathbf{A_j^{(t+1)^{-1}}}\mathbf{\Sigma}_{ij}^{(t)x'x'}\}=0 \quad (2.28)$$

$$\boldsymbol{b}_j^{(t+1)}=\left[\sum_{i=1}^{M} n_{ij}\mathbf{A_j^{(t+1)^{-\top}}}\mathbf{\Sigma}_i^{xx^{-1}}\mathbf{A_j^{(t+1)^{-1}}}\right]^{-1}\left[\sum_{i=1}^{M} n_{ij}\mathbf{A_j^{(t+1)^{-\top}}}\mathbf{\Sigma}_i^{xx^{-1}}\mathbf{A_j^{(t+1)^{-1}}}\right]\left(\boldsymbol{\mu}_{ij}^{(t)x'}-\mathbf{A_j^{(t+1)}}\boldsymbol{\mu}_\mathbf{i}^\mathbf{x}\right) \quad (2.29)$$

$\mathbf{C}_\rho$，$\boldsymbol{d}_\rho$ に関しても，$\boldsymbol{x}$ と $\boldsymbol{x}'$ の代わりに $\boldsymbol{y}$ と $\boldsymbol{y}'$ を用いることで同様に計算することができ，最終的に，新たな入力話者特徴量 x'_k から y' への変換関数は以下の式から計算される．

$$\begin{aligned} F(\boldsymbol{x}'_k) &= E(\boldsymbol{y}' \mid \boldsymbol{x}'_k) \\ &= \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=1}^{L} p(w_i \mid \boldsymbol{x}'_k, w_i)p(\kappa_\rho \mid w_i) \\ &\quad \cdot[\mathbf{C}_\rho\boldsymbol{\mu}_i^y + \boldsymbol{d}_\rho + \mathbf{C}_\rho\mathbf{\Sigma}_i^{yx}\mathbf{\Sigma}_i^{xx^{-1}}\mathbf{A}_j^{-1} \\ &\quad (\boldsymbol{x}'_k - \mathbf{A}_j\boldsymbol{\mu}_i^x - \boldsymbol{b}_j)] \end{aligned}$$

$$p(w_i \mid \boldsymbol{x}'_k) = \frac{p(w_i)\sum_{j=1}^{N} p(\lambda_j \mid w_i)g(\boldsymbol{x}'_k \mid w_i, \lambda_j)}{\sum_{i=1}^{M}\sum_{j=1}^{N} p(w_i)p(\lambda_j \mid w_i)g(\boldsymbol{x}'_k \mid w_i, \lambda_j)}$$

$$p(\lambda_j \mid \boldsymbol{x}'_k, w_i) = \frac{p(\lambda_j \mid w_i)g(\boldsymbol{x}'_k \mid w_i, \lambda_j)}{\sum_{j=1}^{N} p(\lambda_j \mid w_i)g(\boldsymbol{x}'_k \mid w_i, \lambda_j)}$$

このような，既に学習済みのモデルを初期値とし少量のデータによって新たな話者への変換モデルへ適用を行う parameter adaptation を主軸とした手法が GMM ではよく用いられている．

2.7.2 MAP-based parameter adaptation

GMM における parameter adaptation を用いた手法の 1 つとして MAP-Based parameter adaptation がある (MAP: maximum a posteriori probability) [7]．この手法では上述したような線形変換を元にした parameter adaptation と同様に，パラレルコーパスの存在する特定話者間の同時確率をあらかじめ GMM により学習し，そこに新しい話者の非パラレルデータを少量用いてパラメータの適応を行う．手法の概要を図 2.6 に示す．図 2.6 のように，入力話者と出力話者の同時確率を学習した GMM のパラメータに対して MAP 推定を行うことによって，入力話者と新たな出力話者の同時確率を表す GMM へとパラメータの適応を行う．[7] では，分散共分散行列に関しては入出力話者による変化が少ないと仮定し，適応前のものをそのまま用いる．また，話者間の変換は非線形であるとし，GMM による変換式を以下の式に変更する．

$$\boldsymbol{y} = \boldsymbol{x} + \sum_{i=1}^{M} P(i \mid x)(\boldsymbol{\mu}_\boldsymbol{i}^\boldsymbol{Y} - \boldsymbol{\mu}_\boldsymbol{i}^\boldsymbol{X}) \quad (2.30)$$

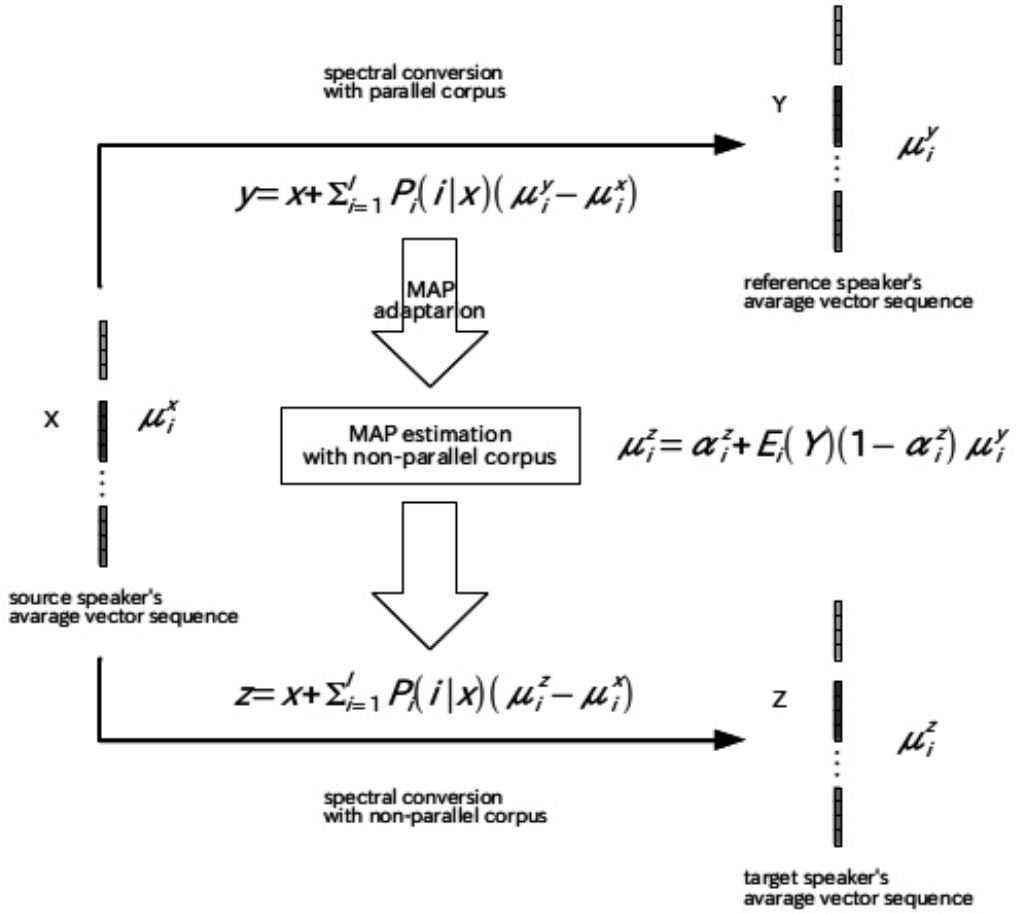

図 2.6: MAP 適応による声質変換の概要

MAP 推定は入力データ列 $\boldsymbol{X}$ が与えられたとき，モデルパラメータ θ を確率変数とし，その事後確率が最大となるような θ を推定する．これを式で表すと以下のようになる．

$$\hat{\theta} = \arg\max_{\theta} P(\theta \mid \boldsymbol{X})$$

この式は，ベイズの定理より以下のように変形できる．

$$\hat{\theta} = \arg\max_{\theta} P(\boldsymbol{X} \mid \theta) P(\theta) \tag{2.31}$$

ここで，$P(\theta)$ は入力データ列 $\boldsymbol{X}$ が与えられていないときの θ の事前分布となり，$P(\boldsymbol{X} \mid \theta)$ は $\boldsymbol{X}$ に関する最尤推定値となる．即ち，MAP 推定は入力データ列 $\boldsymbol{X}$ のサイズが十分大きいときは $\boldsymbol{X}$ に関する最尤推定値に近くなり，十分でないときは事前情報として得られている $P(\theta)$ による値に近くなる．これは声質変換のモデルに当てはめると，事前分布 $P(\theta)$ がパラレルコーパスの存在している特定話者間の GMM のパラメータを表し，それを初期値として非パラレルデータによって適応を行う．

声質変換における GMM の出力話者に関する平均ベクトルの適応を考える．入力話者の特徴量の時間系列 $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n]$ と学習済みの出力話者 (リファレンス話者とする) の特徴量の時間系列 $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_n]$ のパラレルデータを用い, $\boldsymbol{X}$ と $\boldsymbol{Y}$ の各要素の結合ベクトル $\boldsymbol{z}_i = [\boldsymbol{x}_i^\top, \boldsymbol{y}_i^\top]^\top$ の時間系列を $\boldsymbol{Z} = [\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_n]$ としたとき，$\boldsymbol{Z}$ の GMM を以下の式で表す．

$$P(\boldsymbol{z}) = \sum_{m=1}^{M} \alpha_m \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_m^{(Z)}, \boldsymbol{\Sigma}_m^{(Z)})$$

$$\boldsymbol{\mu}_m^{(Z)} = \begin{bmatrix} \boldsymbol{\mu}_m^{(X)} \\ \boldsymbol{\mu}_m^{(Y)} \end{bmatrix} \tag{2.32}$$

$$\boldsymbol{\Sigma}_m^{(Z)} = \begin{bmatrix} \boldsymbol{\Sigma}_m^{(XX)} & \boldsymbol{\Sigma}_m^{(XY)} \\ \boldsymbol{\Sigma}_m^{(YX)} & \boldsymbol{\Sigma}_m^{(YY)} \end{bmatrix} \tag{2.33}$$

このとき，$\boldsymbol{X}$ および $\boldsymbol{Y}$ の GMM も同様に以下の式で表される．

$$P(\boldsymbol{x}) = \sum_{m=1}^{M} \alpha_m \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_m^{(X)}, \boldsymbol{\Sigma}_m^{(XX)})$$

$$P(\boldsymbol{y}) = \sum_{m=1}^{M} \alpha_m \mathcal{N}(\boldsymbol{y}; \boldsymbol{\mu}_m^{(Y)}, \boldsymbol{\Sigma}_m^{(YY)})$$

このとき入力話者とリファレンス話者の変換式は式 (2.30) で表される．この特徴量ベクトル $\boldsymbol{x}$ から特徴量ベクトル $\boldsymbol{y}$ への変換モデルを，特徴量ベクトル $\boldsymbol{x}$ から新しい出力話者 (W とする) の特徴量ベクトル $\boldsymbol{z}$ への変換に適応させる．

変換式 (2.30) を考えると，適応するパラメータは平均ベクトルのみ，すなわち $\boldsymbol{\mu}_m^Y$ を $\boldsymbol{\mu}_m^W$ に適応すればよい．新しい出力話者の非パラレルコーパスからデータ $\boldsymbol{w}_i$ が与えられたとき，この $\boldsymbol{w}_i$ がリファレンス話者の GMM における k 番目のガウス分布から生成されている確率は，

$$P(k \mid \boldsymbol{w}_i)_{k,i} = \frac{\alpha_k \mathcal{N}(\boldsymbol{w}_i; \boldsymbol{\mu}_k^{(Y)}, \boldsymbol{\Sigma}_k^{(YY)})}{\sum_{m=1}^{M} \alpha_m \mathcal{N}(\boldsymbol{w}_i; \boldsymbol{\mu}_m^{(Y)}, \boldsymbol{\Sigma}_m^{(YY)})}$$

となる．このとき，非パラレルコーパス中のデータ $\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_n$ について，k 番目のガウス分布から生成されるデータに関する確率的サンプル数 N_k と確率的平均ベクトル $\boldsymbol{e}_k$ は以下の式で表される．

$$N_k = \boldsymbol{\Sigma}_{i=1}^{n} P(k \mid \boldsymbol{z}_i)_{k,i} \tag{2.34}$$

$$\boldsymbol{e}_k = \frac{1}{N_k} \boldsymbol{\Sigma}_{i=1}^{n} P(k \mid \boldsymbol{w}_i)_{k,i} \boldsymbol{w}_i \tag{2.35}$$

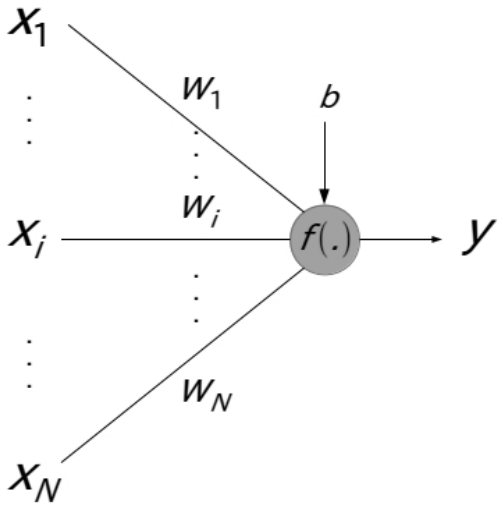

図 2.7: Artificial neuron

これらの式を用いて，$\boldsymbol{\mu}_m^Y$ を以下の様に更新する．

$$\hat{\boldsymbol{\mu}}_m^Y = \frac{N_k}{N_k + \gamma}\boldsymbol{e}_k + \frac{\gamma}{N_k + \gamma}\boldsymbol{\mu}_k^Y \tag{2.36}$$

γ は事前分布，すなわちリファレンス話者の特徴量系列における k 番目のガウス分布から生成される確率的サンプル数を表す．

2.8 Artificial Neural Network

GMM 以外の入力特徴量を出力特徴量に直接変換するモデルの 1 つとして，Artificial Neural Network (ANN) がある．ANN は，人間のニューロン (神経細胞) を数式的に模した Artificial Neuron を用いてネットワーク構造を構築したモデルのことをいう．Artificial Neuron の概要を図 2.7 に示す． Artificial Neuron は数式化すると，以下の式 (2.37) で表される．

$$y = f(\sum_{i=1}^{N}(w_i x_i + b_i)) \tag{2.37}$$

ここで，x_i は入力信号を表し，w_i は各入力信号にかけられる結合重み，y は出力信号をそれぞれ表す．$f(\cdot)$ は活性化関数であり，シグモイド関数などが用いられる．また，b はバイアス項，N は入力の次元数をそれぞれ表す．これは他のニューロンからの入力信号がシナプスを通して現在のニューロンに伝達し，それが閾値を越えたとき現在のニューロンが発火し，結合している他のニューロンへ信号を伝達するという仕組みをモデル化してい

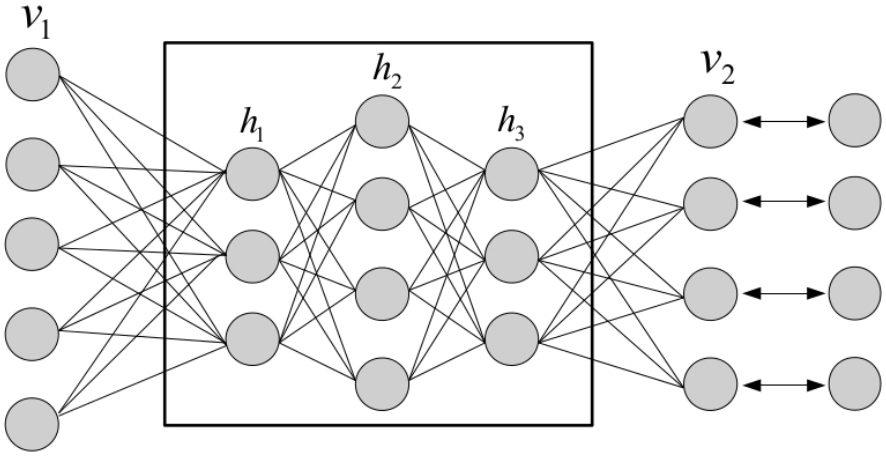

図 2.8: Multi-layer perceptron

る．この Artificial Neuron を複数接続し，ネットワーク構造を成したものを ANN と呼ぶ．ANN の 1 つであり，後述する Deep Neural Network と同じ構造を持つモデルである多層パーセプトロンの例を図 2.8 に示す．多層パーセプトロンは，複数の Artificial Neuron を層状に配置し，隣接する層との間で結合したものである．h_i は隠れ層，v_1 と v_2 は可視層と呼ばれ，v_1 と v_2 がそれぞれ入力層と出力層となる．このモデルに対して，入力特徴量と正解データ (とする特徴量) の組からなるパラレルコーパスを用いて，入力に対する出力と正解データの間の誤差を計算し，その誤差の値によって出力層側から順に各層の重み w_{ij}，バイアス b_j を更新することで学習を行う．この学習を誤差逆伝搬法という．

ANN は，十分な層数・ノード数を用いて構成することで，任意の関数を再現できるという非常に高い表現力を持つ．一方で ANN は，層を増やす程に，誤差逆伝搬法によるパラメータの更新が入力側に近い層まで伝達しづらくなり，また，その高い表現力のために過学習が起きやすくなるという問題点がある．この問題点を改善するために後述する Deep Learning という ANN の学習手法が提案された．

2.8.1　Artificial Neural Network による声質変換

ANN による実際の学習の例として，M 層の ANN を用いて声質変換を行う場合を考える ($m = 1, ..., M$ とする) [5]．この DP マッチングを行ったソース話者とターゲット話者の D 次元特徴量系列を $\boldsymbol{X} = \boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n$，$\boldsymbol{Y} = \boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_n$ とし，m 層目の隠れ層での値を $\boldsymbol{h}^{(m)}$ とすると，ANN によって t 番目の入力データ $\boldsymbol{x}_t$ に対して各層で行われる処理は以下の式で表される．

$$\boldsymbol{h}_t^{(1)} = \mathrm{sigm}(\mathbf{W}^{(1)}\boldsymbol{x}_t + \boldsymbol{b}^{(1)}) \quad : \text{input layer} \tag{2.38}$$

$$\boldsymbol{h}_t^{(m+1)} = \mathrm{sigm}(\mathbf{W}^{(m+1)}\boldsymbol{h}_t^{(m)} + \boldsymbol{b}^{(m+1)}) \quad : \text{hidden layer}(m < M) \tag{2.39}$$

$$\hat{\boldsymbol{y}}_t = \mathbf{W}^{(M)}\boldsymbol{h}_t^{(M-1)} + \boldsymbol{b}^{(M)} \quad : \text{output layer} \tag{2.40}$$

$\hat{\boldsymbol{y}}_t$ はソース話者からターゲット話者に変換された特徴量，$\mathbf{W}^{(m)}$，$\boldsymbol{b}^{(m)}$ は m 層目の結合重み行列とバイアスベクトルをそれぞれ表す．また，sigm は式 (2.41) で定義されるシグモイド関数を表し，緩やかな閾値関数のように働く．

$$\mathrm{sigm(x)} = \frac{1}{1+\exp^{-ax}} \tag{2.41}$$

上述した出力層の式で得られた変換特徴量 $\hat{\boldsymbol{y}}_t$ と正解データ $\boldsymbol{y}_t$ の間で誤差を計算する．ANN による連続値変換の誤差関数としては主に以下の式で表される二乗平均誤差 (MSE) が用いられる．

$$\boldsymbol{MSE} = \sum_{k=1}^{n} \mid \hat{\boldsymbol{y}}_k - \boldsymbol{y}_k \mid^2 \tag{2.42}$$

ANN ではこの誤差関数の値が最小となる様にパラメータ $\mathbf{W}$，$\boldsymbol{b}$ を誤差逆伝搬法によって更新する．初めに 1 フレームの特徴量に対して，出力層において $\hat{y}_j$ を値として持つ j 番目のノードに関する誤差逆伝搬法を考える．$(j = 1, 2, ..., D)$ このとき，出力層の j 番目のノードにおける出力の誤差は $E_j = \mid \hat{y}_j - y_j \mid^2$ で求められるとする．誤差逆伝搬法におけるパラメータの更新は勾配法によるものであり，m 層における入力側 i 番目のノードと出力側 j 番目のノードの結合重みの更新式は以下の様に定義される．

$$W_{ij}^{m} = \mathbf{W}_{ij}^{m} - \epsilon \frac{\partial E}{\partial W_{ij}^{m}} \tag{2.43}$$

ϵ は学習率である．出力層の前の層 ($M-1$ 層) の i 番目のノードの出力値を $h_i^{(M-1)}$ とすると，$\hat{y}_j$ の変化量に対する $W_{ij}^{(M)}$ の変化量の関係は以下の式で示される．

$$\Delta \hat{y}_j = \Delta W_{ij}^{(M)} h_i^{(M-1)} \tag{2.44}$$

$E_j = \mid \hat{y}_j - y_j \mid^2$ なので，

$$\Delta E = 2(\hat{y}_j - y_j)\Delta \hat{y}_j \tag{2.45}$$

よって出力層における更新式は式 (2.46) の様になる．

$$W_{ij}^{(M)} = W_{ij}^{(M)} - 2\epsilon(\hat{y}_j - y_j)h_i^{(M-1)} \tag{2.46}$$

出力層以外の層，すなわち活性化関数としてシグモイド関数を用いている層でも同様の手順によって以下の更新式が得られる．

$$\begin{aligned} z_j^{(m)} &= a(1-h_j^{(m)})h_j^{(m)} \sum_{k=1}^{L} W_{jk}^{(m+1)} z_k^{(m+1)} \\ W_{ij}^{(m)} &= W_{ij}^{(m)} - \epsilon z_j^{(m)} h_i^{(m-1)} \end{aligned} \tag{2.47}$$

ここで，L は $m+1$ 層におけるノード数である．また，$x = \mathrm{sigm}(s)$ のとき，$x' = ax(1-x)$ を利用している．

バイアス項に関しては結合重みの更新の際に求めた $z^{(m)}$ を用いて以下の式で更新する．

$$\boldsymbol{b}^{(M)} = \boldsymbol{b}^{(M)} - 2\epsilon(\hat{\boldsymbol{y}} - \boldsymbol{y}) \quad : \mathrm{outputlayer} \boldsymbol{b}^{(m)} = \boldsymbol{b}^{(m)} - \epsilon z^{(m)} \quad : \mathrm{other} \tag{2.48}$$

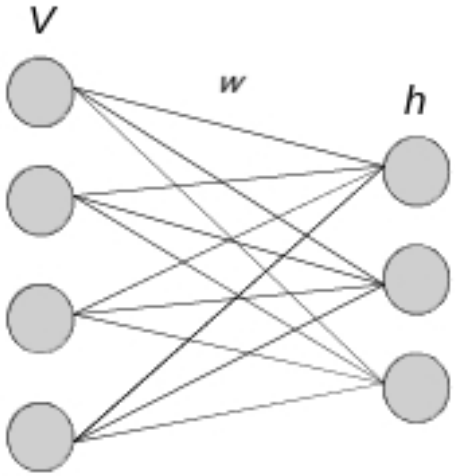

図 2.9: Restricted Boltzmann Machine のグラフィカルモデル

2.8.2 Deep Learning

Deep Learning (深層学習) は，ANN における収束が遅いという問題点と過学習に陥りやすいという問題点を改善するために提案された手法である．初めに，Deep Learning の要素技術である Restricted Boltzmann Machine，Denoising Auto Encoder について述べる．

i) Restricted Boltzmann Machine

Restricted Boltzmann Machine (RBM) はニューラルネットの特殊形であり，可視層と隠れ層の間にのみ結合が存在し，可視層間・隠れ層間での結合が存在しない無向グラフィカルモデルで表される [16]．RBM では通常の ANN と同様に，可視層への入力に対して重み付け和をとりバイアス項を足し，シグモイド関数を掛けたものを隠れ層の値とする．図 2.9 に RBM のグラフィカルモデルを示す．

図 2.9 中の v は可視層, h は隠れ層，w は結合の重みをそれぞれ表す．

数式で示すと，RBM では可視素子の集合 $\boldsymbol{v} = \{v_1, v_2, ..., v_N\} \in \{0, 1\}$ と隠れ素子の集合 $\boldsymbol{h} = \{h_1, h_2, ..., h_N\} \in \{0, 1\}$ からなる結合確率 $p(\boldsymbol{v}, \boldsymbol{h})$ を，以下の式 (2.49)(2.50)(2.51) で定義する．

$$p(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{Z} \exp(-E(\boldsymbol{v}, \boldsymbol{h})) \tag{2.49}$$

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\boldsymbol{b}^\top \boldsymbol{v} - \boldsymbol{c}^\top \boldsymbol{h} - \boldsymbol{v}^\top \mathbf{W} \boldsymbol{h} \tag{2.50}$$

$$Z = \Sigma_{\boldsymbol{v}, \boldsymbol{h}} \exp(-E(\boldsymbol{v}, \boldsymbol{h})) \tag{2.51}$$

$\mathbf{W}$ は結合重，$\boldsymbol{b}$ は可視素子のバイアスパラメータ，$\boldsymbol{c}$ は隠れ素子のバイアスパラメータをそれぞれ表す．

RBM には可視層間・隠れ層間での結合が存在しないという制約があるため，条件付確率

$p(v_j = 1 \mid \boldsymbol{h})$，$p(h_i = 1 \mid \boldsymbol{v})$ は以下の式のようになる．

$$p(v_j = 1 \mid \boldsymbol{h}) = \mathrm{sigm}(b_j + \mathbf{W}_j \boldsymbol{h}) \tag{2.52}$$
$$p(h_i = 1 \mid \boldsymbol{v}) = \mathrm{sigm}(c_i + \mathbf{W}_i^\top \boldsymbol{v}) \tag{2.53}$$

ここで，sigm はシグモイド関数を示す．

パラメータの推定には，$p(\boldsymbol{v})$ に対する最尤推定を行う．つまり，対数尤度の任意のパラメータ θ に関する最大化を行う．$p(\boldsymbol{v})$ の対数尤度を J とすると，その任意のパラメータ θ により微分は, 以下の様に表される．

$$\frac{\partial J}{\partial \theta} = -\left\langle \frac{\partial E}{\partial \theta} \right\rangle_{\mathrm{data}} + \left\langle \frac{\partial E}{\partial \theta} \right\rangle_{\mathrm{model}} \tag{2.54}$$

ここで $\langle . \rangle_{\mathrm{data}}$，および $\langle . \rangle_{\mathrm{model}}$ はそれぞれ観測データに対しての期待値，内部モデルに対しての期待値を表す．第一項は，式 (2.52)(2.53) から比較的簡単に求めることができるが，第二項は全ての $\boldsymbol{v}$，$\boldsymbol{h}$ の組み合わせを考えなければならないのでノード数によっては困難となる．そのため，条件付確率 $p(\boldsymbol{v} \mid \boldsymbol{h})$ と $p(\boldsymbol{h} \mid \boldsymbol{v})$ によって可視素子の状態集合 $\boldsymbol{v}$ を再構成した $\boldsymbol{v}'$ を用いて，最急降下法によって近似的にパラメータの更新を行っていく方法がよく取られる (Contrastive Divergence 法) [23]．

ii)　Denoising Auto Encoder

RBM 以外の Deep Learning の pre-training に用いられる特徴量抽出器として Denoising Auto Encoder (以下 dAE) がある．初めに通常の Auto Encoder (AE) について示す．AE は RBM と同様にニューラルネットの特殊形であり，入力層・隠れ層・復元層の 3 層からなる各層内に結合が存在しない無向グラフィカルモデルで表される．図 2.10 に dAE のグラフィカルモデルを示す．

dAE では入力層に与えられた値を隠れ層を通した上で復元層で再構成し，入力データそのものを教師データとして誤差が最小になる様にパラメータの学習を行う．このとき，隠れ層のノード数を入力データの次元数より少なくした場合，再構成の際により少ない情報から元のデータを復元する必要がある．そのため，隠れ層には元のデータよりも情報が圧縮された特徴量が生成されると考えられる．また，AE では通常の ANN と同様に，可視層への入力に対して重み付け和をとりバイアス項を足し，シグモイド関数を掛けたものを隠れ層の値とする．そして隠れ層の値に対して，同様に重み付け和をとりバイアス項を足し，シグモイド関数を掛けることで復元層の値を出力する．式で表すと以下のようになる．

$$\boldsymbol{h} = \mathrm{sigm}(\mathbf{W}\boldsymbol{x} + \boldsymbol{b})$$
$$\boldsymbol{y} = \mathrm{sigm}(\mathbf{W}^\top \boldsymbol{h} + \boldsymbol{b}')$$

ここで，$\mathbf{W}$，$\mathbf{W}^\top$ はそれぞれ可視層から隠れ層への結合重みと隠れ層から復元層への結合重み (可視層から隠れ層への結合重みの転置)，$\boldsymbol{b}$，$\boldsymbol{b}'$ は可視層から隠れ層へのバイアスパ

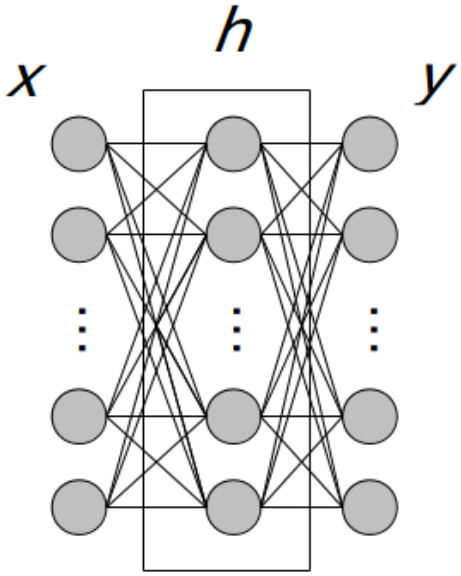

図 2.10: Denoising Auto Encoder の概略図

ラメータと隠れ層から復元層へのバイアスパラメータをそれぞれ表す．$\boldsymbol{x}$，$\boldsymbol{h}$，$\boldsymbol{y}$ はそれぞれ入力層と隠れ層と復元層の値を表す．

パラメータの更新は，ANN の時と同様に誤差関数 E が最小となるように勾配法によって得られる以下の式で行う．

$$\begin{aligned}\mathbf{W} &= \mathbf{W} - \epsilon\frac{\partial E}{\partial \mathbf{W}} \\ \boldsymbol{b} &= \boldsymbol{b} - \epsilon\frac{\partial E}{\partial \boldsymbol{b}} \\ \boldsymbol{b}' &= \boldsymbol{b}' - \epsilon\frac{\partial E}{\partial \boldsymbol{b}'}\end{aligned}$$

dAE では，AE での入力層の値 x に対してノイズを加えた $\hat{\boldsymbol{x}}$ を入力値として用い，$\hat{\boldsymbol{x}}$ から $\boldsymbol{x}$ を復元するようなパラメータ $\mathbf{W}$，$\boldsymbol{b}$，$\boldsymbol{b}'$ をそれぞれ求める．これにより，隠れ層の次元数が入力の次元数よりも大きい場合であっても，恒等関数が最適にならないような問題に変形することができ，より頑健かつ汎化性能の高い特徴量の抽出を行うことができる．

iii)　Deep Neural Network

DNN では ANN における収束が遅いという問題点と過学習に陥りやすいという問題点を改善するために，誤差逆伝搬法を行う前に pre-training と呼ばれる各パラメータの初期値を計算する処理が行なわれていた [9]．近年では Rectified Linear Units (ReLU)[19] や Leaky ReLU[20] などの勾配消失が発生しにくい活性化関数が用いられるようになったことにより，

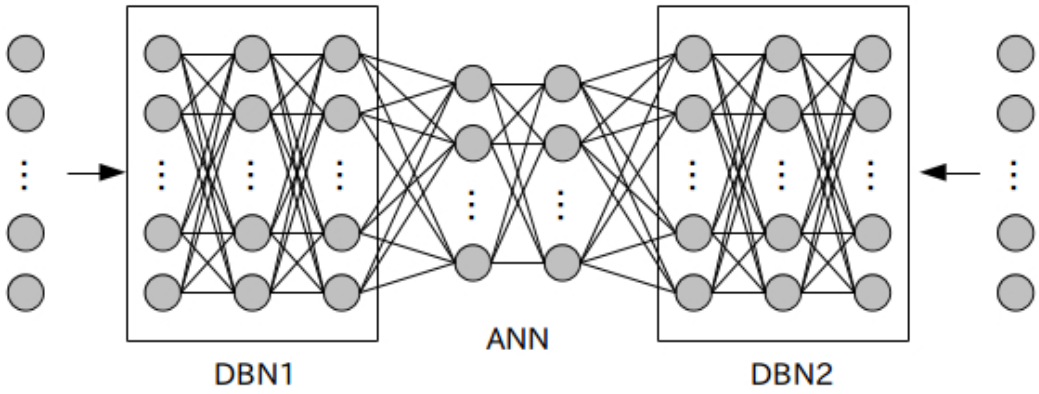

図 2.11: Deep Belief Nets による低次元空間表現を用いた声質変換

相対的にこれらの pre-training の重要性は下がったものの，DNN におけるモデルの事前学習の有用性が示されている手法も存在している [21, 22].

pre-training では，dAE または RBM を用いて学習データに対して教師なし学習を行ったものを 1 層目とする．そして，学習の終わった 1 層目の隠れ層の値を学習データとして，同じように 2 層目を dAE，RBM によって学習する．この処理を繰り返すことによって全ての層の初期値を決定する．pre-training によって初期値が決定した後は，通常の ANN と同様に誤差逆伝搬法によって教師あり学習を行う．DNN における，この誤差逆伝搬法による教師あり学習を fine-tuning と呼ぶ.

2.8.3　Deep Belief Nets による声質変換手法

Deep Learning を用いた声質変換手法の 1 つとして Deep Belief Nets による低次元空間表現を用いた声質変換という手法が存在する [10]．Deep Belief Nets(DBN) は通常の 2 層からなる RBM を学習し，その隠れ層を次の RBM の入力と考えることで RBM を多層化した特徴量抽出器のことをいう．図 2.11 に手法の概要を示す.

この研究では，深い階層を持つ DBN では各層のノード数で入力特徴量を表現するため，層の数が増えるほど入力特徴量が基底集合に近くなると仮定している．その考えを基に，ソース話者の特徴量と出力話者の特徴量をそれぞれ別の DBN によって抽出し，それぞれの最上位を ANN で接続する．このモデルによって fine-tuning を行うことで，より話者性の薄れた低次元空間において変換を行うことができるため，最適な非線形変換が可能となると考えている．実験として GMM による変換音声との主観評価と客観評価による比較を行っており，両尺度において GMM を上回る結果が得られていた.

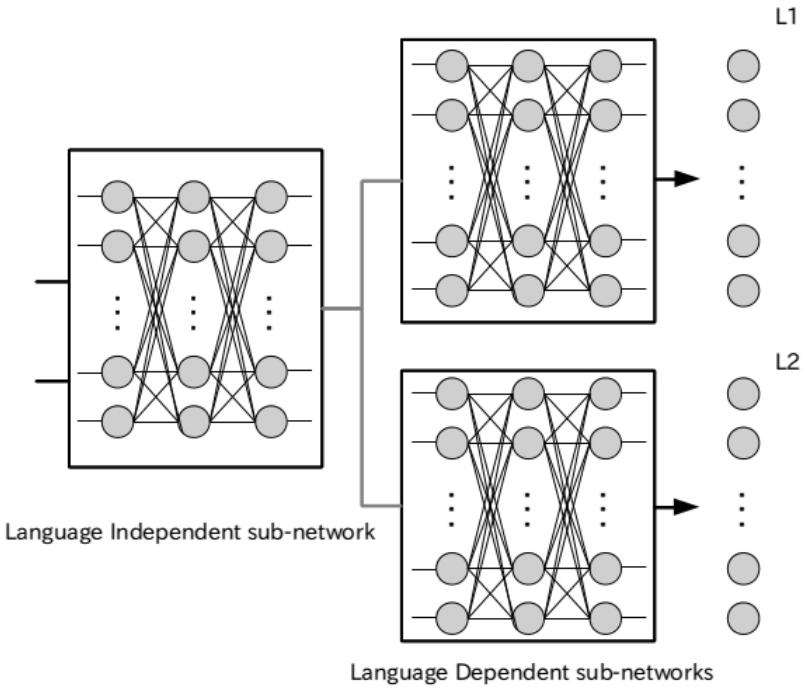

図 2.12: サブネットワーク構造を用いた Deep Neural Networks

2.8.4　多言語音声を学習した Deep Neural Network における言語非依存サブネットワークの自動適応

DNN のパラメータの適応を試みた手法として，松田らによる多言語音声を学習した Deep Neural Network における言語非依存サブネットワークの自動適応がある [13]．この手法は，多言語音素認識というタスクを対象としており, 入力として複数の国の言語による音声が与えられ, その音声中の音素を識別するという一種の classifier の実装を目的としている．

手法の概略を図 2.12 に示す．この手法では，大枠としては前節で説明した Deep Belief Nets を使用しており，pre-training においては，複数言語の音声からなる訓練データによって RBM を入力層から順に学習していく．次に pre-training の終わったネットワークを一定の層で区切り，前半はそのまま言語非依存のサブネットワークとして，後半は言語の数だけネットワークを複製し言語依存のネットワークとして使用する. 具体的には，教師あり学習 (fine-tuning) の際に前半の層は学習データの言語に関わらず 1 つのものを使用し，後半の層は学習データの言語毎に対応するネットワークを選択し学習を行う．実際の classtering の際には，全てのサブネットワークの出力値 (入力特徴量が与えられた時の各言語，各音素である事後確率を表す) から最大値となるものを最終的な認識結果とする．

この手法ではまず初めに仮定として，Deep Learning では前半の浅い層において殆どの音響的事象に共通する時間変動や周波数などの特徴量の識別を行っており，逆に深い層では音素や言語依存の特徴量などの複雑な情報を扱っていると考えている．そこで，前半の層を入力言語に対して共通にする一方で，後半で使用する層を言語毎に変更することで，前半の言語非依存のサブネットワークでは言語に依存しない処理が集中し，後半の言語依

存のサブネットワークでは言語に依存する処理が集中するということが，前述した仮定がより顕著になることで実現できるのではないかと考えており，実験結果においても一対一で学習を行った DBN に対して高い，もしくは同程度の精度を出している．

2.8.5　Factorized Hidden Layers を用いた DNN のパラメータ適応

音声認識において DNN の各層に直接パラメータ適応を行うことを試みた手法として，Factorized Hidden Layers (FHL) が存在する [41]．ここでは，FHL を用いた DNN の基本的な構造及び，FHL を用いた話者適応処理のための学習について示す．

i)　FHL の構築

大量の事前収録話者によって学習した話者非依存の DNN において，l 層目の隠れ層における変換行列及びバイアス項をそれぞれ $\mathbf{W}^l$, $\boldsymbol{b}^l$ とすると，隠れ層における変換は以下の様に表される．

$$\boldsymbol{h}^l = \sigma(\mathbf{W}^l \boldsymbol{h}^{l-1} + \boldsymbol{b}^l) \tag{2.55}$$

ここで $\boldsymbol{h}^l \in \mathbb{R}^{n_l \times 1}$ は l 層目の隠れ層の出力 n_l 次元ベクトルを表す．σ は活性化関数を表す．

この話者非依存 DNN のパラメータを基に FHL では話者依存の各隠れ層を以下の様にモデル化する．

$$\boldsymbol{h}^l = \sigma(\mathbf{W}_s^l \boldsymbol{h}^{l-1} + \boldsymbol{b}_s^l) \tag{2.56}$$

$$\mathbf{W}_s^l = \mathbf{W}^l + \mathbf{\Gamma}^l \mathbf{D}_s^l \mathbf{\Psi}^{l\top} \tag{2.57}$$

$$\boldsymbol{b}_s^l = \boldsymbol{b}^l + \mathbf{U}^l \boldsymbol{v}_s^l \tag{2.58}$$

ここで，$\mathbf{D}_s^l \in \mathbb{R}^{|\boldsymbol{d}_s^l| \times |\boldsymbol{d}_s^l|}$ は，適応対象となる話者 s の話者表現ベクトル $\boldsymbol{d}_s^l$ を対角成分に持つ行列を表す ($\mathbf{D}_s^l = \mathrm{diag}(\boldsymbol{d}_s^l)$)．話者表現ベクトルとは，ある話者の GMM スーパーベクトル空間における平均話者からの偏差を表すベクトルであり，i-vector や EVGMM などの手法によって求めることができる [8, 28]．$\mathbf{\Gamma}^l \in \mathbb{R}^{n_l \times |\boldsymbol{d}_s|}$，及び $\mathbf{\Psi}^l \in \mathbb{R}^{n_{l-1} \times |\boldsymbol{d}_s|}$ は話者空間の基底に相当するパラメータである．またバイアス項も同様に，$\boldsymbol{v}_s^l \in \mathbb{R}^{|\boldsymbol{v}_s^l| \times 1}$ は適応対象となる話者の話者表現ベクトルに相当し，$\mathbf{U}^l \in \mathbb{R}^{n_l \times |\boldsymbol{v}_s^l|}$ は話者空間の基底に相当する．

ここで，$\boldsymbol{d}_s^l$ と $\boldsymbol{v}_s^l$ の両方を零ベクトルとすると，式 (2.56) は FHL を用いない通常の DNN と同等となる．このため，大量の事前収録話者を用いた話者非依存 DNN によって初期パラメータの決定を行うことができる．また，$\boldsymbol{d}_s^l$ を零ベクトルとすると，FHL は音声認識等のタスクで広く用いられているバイアス適応の式となり，バイアス項の話者表現として i-vector や話者コードを用いることで一般的な話者適応と同等となる．

FHL による変換の構造の図を図 2.13 に表す．図 2.13 に表されている様に FHL による隠れ層間の変換は，隠れ層と隠れ層の間の通常のアフィン変換に加えて，行列積とバイアス適応が追加されている形になる．

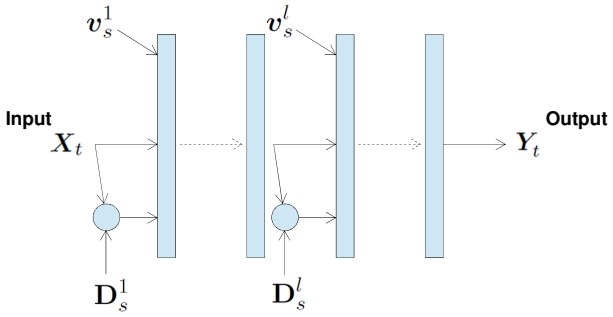

図 2.13: FHL による DNN パラメータ適応

ii)　FHL の学習

ここでは，パラメータ $\boldsymbol{\Gamma}^l$，$\mathbf{D}_s^l$，$\boldsymbol{\Psi}^l$，$\mathbf{U}^l$，$\boldsymbol{v}_s^l$ の学習について示す．

まず話者表現ベクトル $\boldsymbol{d}_s^l$ 及び $\boldsymbol{v}_s^l$ に関しては，既存の話者表現手法 (i-vector，話者コードなど) の値を与え，話者非依存 DNN のパラメータと共にその値で固定し更新を行わない．この状態で，大量の事前収録話者のパラレルデータ及び話者表現ベクトルを用い，$\boldsymbol{\Gamma}^l$，$\boldsymbol{\Psi}^l$，$\mathbf{U}^l$ を乱数で初期化した上で，通常の誤差伝播学習によってパラメータの推定を行う．

$\boldsymbol{\Gamma}^l$，$\boldsymbol{\Psi}^l$，$\mathbf{U}^l$ の推定後，今度はこれらのパラメータを固定した上で話者表現ベクトル $\boldsymbol{d}_s^l$ 及び $\boldsymbol{v}_s^l$ の更新を行う場合もある．

第3章

従来の多対多声質変換手法

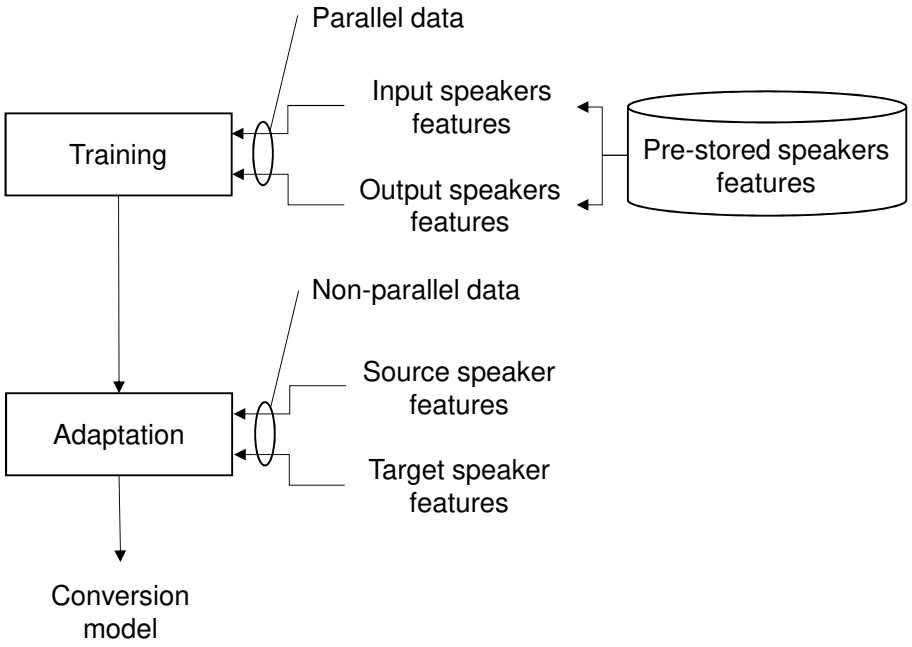

図 3.1: 多対多声質変換の概略図

3.1　多対多声質変換

一般的な声質変換手法は特定の入力話者から特定の出力話者への変換を対象としており，学習データ外の未知話者に対してはパラメータのミスマッチから，変換精度が著しく低下してしまう．学習データに含まれていない話者を変換対象にしようとする度に，数十文の音声データを収録するのはコストが大きい．また，新たに音声データを収録するのが難しく，パラレルデータに必要な特定の音声データを用意できない場合も考えられる．そのため，声質変換において，より少量のデータによる柔軟な話者性の制御という課題は重要といえる．このような問題を改善するための，より少量のデータによって学習データ外の新たな話者を用いた変換に適応する声質変換手法を多対多声質変換と呼ぶ．図 3.1 に一般的な多対多声質変換の概要を示す．多対多声質変換の多くは，2.8 節で示したようなパラメータ適応によって行われる．すなわち，予め用意されたパラレルデータによって入出力話者に依存しない情報を事前知識として学習した初期モデルを構築し，少量の非パラレルデータによって新たな話者を対象とした変換に適応するという枠組みによって多対多声質変換は実現される．

加えて近年では，少量の非パラレルデータによる適応を可能とするため，話者情報をより適切な形で表現することを目的とした特徴量抽出手法が提案されている．

ここでは，2.8 節で示した GMM における話者適応手法に加え，多対多声質変換を目的とした話者表現及び変換手法の既存研究について示す．

3.1.1　i-vector による話者表現

音声認識の分野で広く用いられている話者情報表現の 1 つとして i-vector が存在する [28]．ここでは最も一般的なものとして GMM の super-vector を用いた i-vector について示す．GMM super-vector は学習データ中の各発話に関して GMM の学習を行い，GMM の全混合の平均ベクトルを連結した特徴量のことを言う．GMM，特に大量の話者によって学習を行ったモデル (Universal background model: UBM) は話者情報は平均的な話者として学習され，各混合に音素を表すような情報が集約されることが期待される．ここで，音素はある言語において区別されている音声の最小の単位のことを言う．

GMM super-vector に対して因子分析を行うことによって，低次元空間へ射影を行うことによって i-vector を得る．特に，大量の話者によって学習をおこなった UBM を事前分布として 1 発話によって一人の話者への MAP 適応を行い，そこから抽出した GMM super-vector から得られる i-vector を話者表現として使用することが多い．UBM の GMM super-vector を $\boldsymbol{m}$，1 発話によって一人の話者へと適応を行った GMM の super-vector を $\boldsymbol{M}$ としたとき，因子分析によって低次元空間への射影行列 $\boldsymbol{T}$(Total variability matrix と呼ぶ) を用いて以下の様に分解を行うことができる．

$$\boldsymbol{M} = \boldsymbol{m} + \boldsymbol{T}\boldsymbol{w} \tag{3.1}$$

この $\mathcal{N}(0, \boldsymbol{I})$ に従う $\boldsymbol{w}$ が話者情報を制御する i-vector と呼ばれる．

3.1.2　Average voice mode と i-vector に基づく声質変換

ここでは DNN を用いた Average voice model と i-vector に基づく多対多声質変換手法 (以下 AVMVC) について示す [29]．図 3.2 に手法の概要を示す．図 3.2 で示されているように，AVMVC では通常の DNN による声質変換の枠組みにおいて，入力特徴量としてメルケプストラムに加えて入力話者及び出力話者の i-vector を使用している．i-vector を入力に加えた上で大量話者からなるパラレルデータによって 1 つの DNN を学習することで，i-vector によって入出力話者の話者性を制御することが可能なモデルを構築する．[29] においては，この大量の話者の情報を持った DNN を AVM と呼ぶ．この AVM を基に，i-vector によって未知の入出力話者の話者性の情報を明示的に与えることによって，多対多声質変換を実現するというのがこの AVMVC の大枠である．

手法の具体的な手順は以下の様になる．

- **学習データの準備：** AVM の学習に用いるための大量のパラレルデータを用意する．(このとき，全話者が同一の発話内容である必要は無い．)
- **学習データの特徴量抽出：** 音声からのスペクトル特徴量の抽出と，i-vector 抽出器による学習データ中の全話者に関する i-vector の抽出を行う．
- **AVM の学習：** DTW によってアラインメントを行った音声特徴量によって DNN を学習することで AVM を構築する．ここで，入力には音声特徴量に加えて入力話者と出力話者の i-vector の組を使用する．(i-vector は各話者の学習データ中の文に関して平均したものを用いる．)

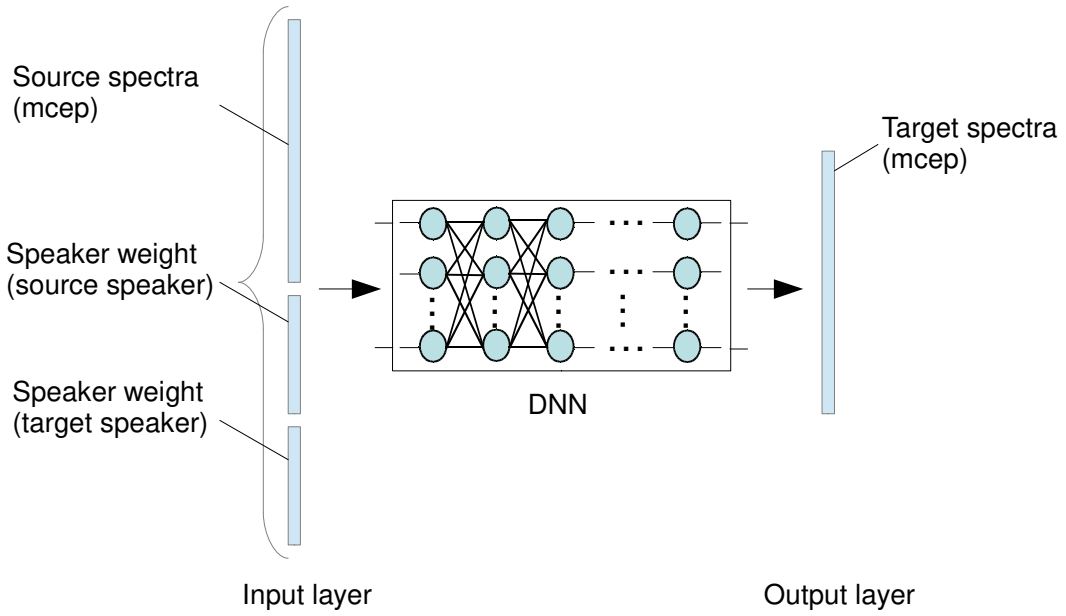

図 3.2: Average voice model による声質変換の概要

・変換対象話者の特徴量抽出及び変換： 実際に変換したい入出力話者に関して，音声特徴量及び i-vector を抽出し，それを学習された AVM に対して入力することで変換を行う．

初期モデルとなる AVM の学習には複数話者からなるパラレルデータが必要となるが，実際に変換を行う未知入出力話者の i-vector の抽出はパラレルデータを必要としないため，教師無し適応によって未知話者を対象とした変換が可能となる．

3.1.3 Eigenvoice conversion

Eigenvoice conversion [8] の概要を図 3.3 に示す．Eigenvoice conversion では, 初めに変換元の話者と多数の事前収録話者からなるパラレルコーパスを用いて, 話者非依存の GMM の学習を行う．次に, この話者非依存の GMM を初期モデルとして, 各事前収録話者の話者依存 GMM を学習する．ここで, 特徴量ベクトルの次元を D, ガウス分布の混合数を $M(m = 1, 2, ..., M)$, 添字 s を各話者 $(s = 0, 1, 2, ..., S$ で, $p = 0$ は不特定話者) とすると, 変換元の特徴量 $\boldsymbol{X}_t$ と s 番目の事前収録話者の特徴量
$vector Y_t^{(s)}$ は eigenvoice GMM (EVGMM) として以下の様にモデル化される．

$$
\begin{aligned}
&P(\boldsymbol{X}_t, \boldsymbol{Y}_t^{(s)} \mid \lambda^{(EV)}, \boldsymbol{w}) \\
&= \sum_{m=1}^{M} \alpha_m \mathcal{N}([\boldsymbol{X}_t^{\top}, \boldsymbol{Y}_t^{(s)\top}]^{\top}; \boldsymbol{\mu}_m^{(Z)}(\boldsymbol{w}^{(s)}), \boldsymbol{\Sigma}_m^{(Z)})
\end{aligned}
\tag{3.2}
$$

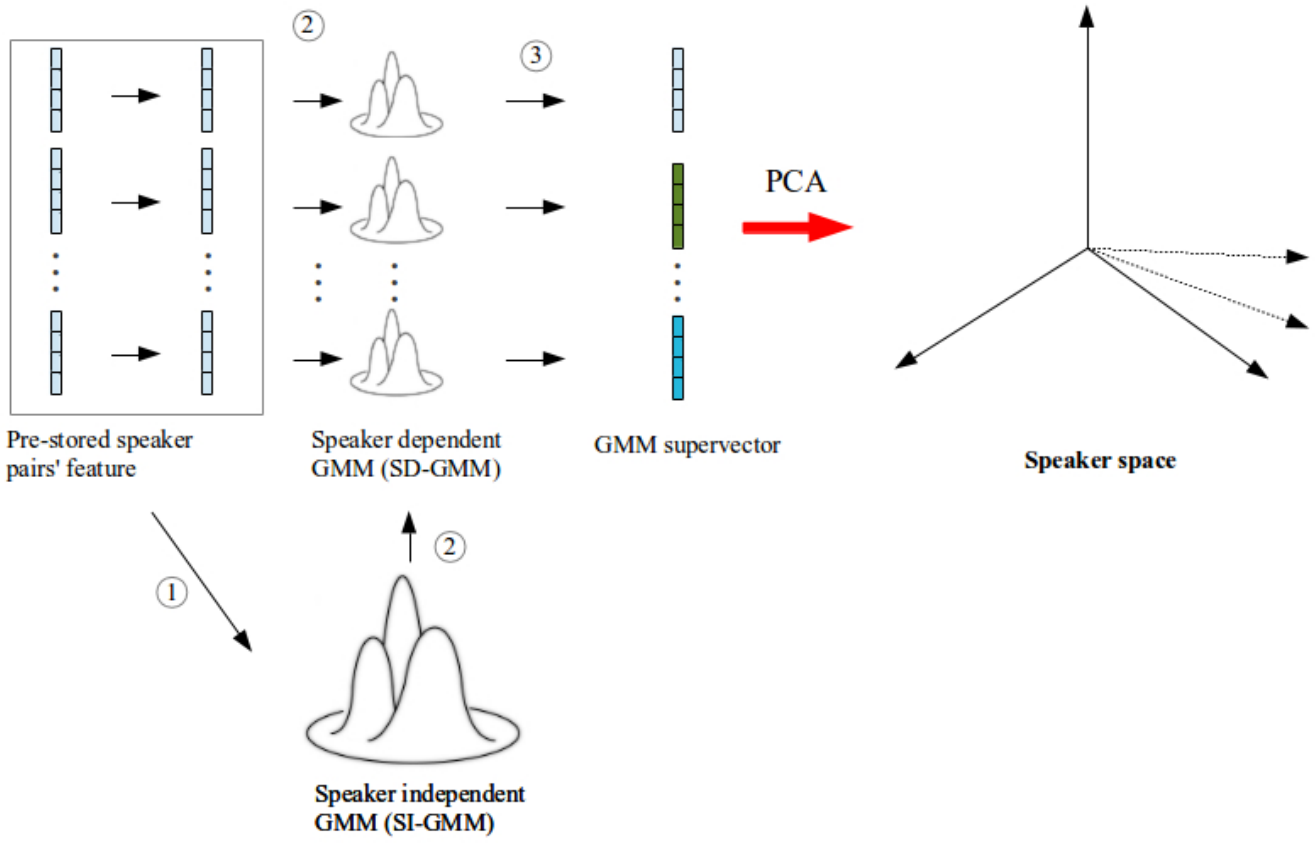

図 3.3: Eigenvoice conversion の概要

$$\boldsymbol{\mu}_m^{(Z)}(\boldsymbol{w}^{(s)}) = \begin{bmatrix} \boldsymbol{\mu}_m^{(X)} \\ \mathbf{B}_m \boldsymbol{w}^{(s)} + \boldsymbol{b}_m^{(0)} \end{bmatrix} \tag{3.3}$$

$$\boldsymbol{\Sigma}_m^{(Z)} = \begin{bmatrix} \boldsymbol{\Sigma}_m^{(XX)} & \boldsymbol{\Sigma}_m^{(XY)} \\ \boldsymbol{\Sigma}_m^{(YX)} & \boldsymbol{\Sigma}_m^{(YY)} \end{bmatrix} \tag{3.4}$$

ここで α_m は m 番目のガウス分布の重みを表す. EVGMM では, S 人の事前収録話者を用いて出力話者の平均ベクトル $\boldsymbol{\mu}_m^{(Y)}$ をバイアスベクトル $\boldsymbol{b}_m^{(0)}$ と K 個の表現ベクトルの線型結合で表す. すなわち, 話者空間が K 個の基底スーパーベクトルとバイアススーパーベクトルによって張られる.

EVGMM を用いた声質変換 (eigenvoice conversion) では, 初めに変換元の話者に対して全ての事前収録話者とのパラレルデータによる学習を行い, 話者非依存の GMM を構築する. それを初期モデルとして, 各事前収録話者の話者依存 GMM を学習する. ここから, 話者空間の特徴量ベクトルとして各事前登録話者の GMM の平均ベクトルを連結し, 得られたスーパーベクトルに主成分分析を行うことで基底ベクトルとし, バイアスベクトル $\boldsymbol{b}$ と表現ベクトル $\mathbf{B}$ を計算する.

任意の話者に対しての EVGMM の適応は, 出力話者のデータを用いた重みベクトル $\boldsymbol{w}$ の最尤推定によって行う. 出力話者の特徴量系列を $\mathbf{Y}^{(\mathbf{tar})}$ としたとき, $\boldsymbol{w}$ は以下の様に推定される.

$$\hat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} P(\mathbf{Y}^{(tar)} \mid \lambda^{(EV)}, \boldsymbol{w}) \tag{3.5}$$

出力の確率密度関数は GMM で表されるため, 補助関数として (3.6) 式を導入し, EM アルゴリズムを用いることで重みベクトル $\boldsymbol{w}$ を最適化する.

$$(\boldsymbol{w}, \hat{\boldsymbol{w}}) = \sum_{m} P(m \mid \mathbf{Y}^{(tar),\lambda^{(EV)}}) \log P(\mathbf{Y}^{(tar)}, m \mid \lambda^{(EV)}, \boldsymbol{w}) \tag{3.6}$$

この補助関数により, $\hat{\omega}$ に関する以下の更新式が得られる.

$$\hat{\boldsymbol{w}} = \{\sum_{m=1}^{M} \bar{\gamma}_m^{(tar)} \mathbf{B}_m^{\top} {\boldsymbol{\Sigma}_m^{(YY)}}^{-1} \mathbf{B}_m\}^{-1} \sum_{m=1}^{M} \mathbf{B}_m \top {\boldsymbol{\Sigma}_m^{(YY)}}^{-1} \bar{\mathbf{Y}}_m^{(tar)} \tag{3.7}$$

$$\bar{\gamma}_m^{(tar)} = \sum_{t=1}^{T} \gamma_{m,t} \tag{3.8}$$

$$\bar{\mathbf{Y}}_m^{(tar)} = \sum_{t=1}^{T} (\mathbf{Y}_t^{(tar)} - \boldsymbol{b}_m^{(0)}) \tag{3.9}$$

$$\gamma_{m,t} = P(m \mid \mathbf{Y}_t^{(tar)}, \lambda^{(EV)}, \boldsymbol{w}) \tag{3.10}$$

式 (3.7) は話者空間の基底ベクトルへの射影重みを推定していることに相当する.

これにより, 推定する必要のあるパラメータが少量かつ出力話者の発話内容を知る必要が無いため, 通常の GMM の学習に比べて極小量のデータによる, 教師無し適応が可能となる.

3.1.4　i-vector と話者固有重みの関係

ここで，i-vector と eigenvoice における話者重みの関係について考える．i-vector における Total variability matrix による射影は主成分分析に相当する．GMM super-vector に対して主成分分析を行うという観点から，i-vector と eigenvoice における話者重みは基本的に同一であると言える．

より厳密には，eigenvoice における話者重みは GMM super-vector に対して主成分分析を行うことで決定的に得られるものであるが，i-vector における重みベクトル $\boldsymbol{w}$ は，確率的主成分分析によって得られる $\mathcal{N}(0, \mathbf{I})$ に従うベクトルである．確率的主成分分析では基底数を事前に設定し，その条件のもとでデータに従うように基底を計算することで特徴量空間を構築する．そのため，確率的主成分分析によって得られる基底は斜交基底に，主成分分析によって得られる基底は直交基底となる．

3.2　パラレルデータフリー声質変換手法

ここまで，変換対象とする話者ペアに関してパラレルデータを必要としない，多対多声質変換手法について示してきた．しかし，前節で示した既存の多対多声質変換手法では，EVGMM における話者空間の構築や AVM の構築などの事前知識の学習において依然としてパラレルデータを使用している．事前知識の獲得においてもパラレルデータを使用することなく学習を行うことができれば，全学習過程において一切の発話内容の制約が無くなる．自由発話によってモデルの学習が実現すれば，より大規模な学習データの利用が可能となり，データ効率の上でも実用面でも非常に有用であると言える．ここでは既存の全学習過程においてパラレルデータを使用していない，完全なパラレルデータフリー声質変換手法について示す．

3.2.1　Iterative combination of a Nearest Neighbor search step and a Conversion step Alignment method

Iterative combination of a Nearest Neighbor search step and a Conversion step Alignment method (INCA)[33, 34] は Unit Selection に基づく声質変換手法である．INCA では異なる話者間の自由発話より得られた音響特徴量から，音素的に同一であると推定される組を抽出し，それらの間で変換を学習する．最も単純には，2 話者の音声特徴量間で適当な距離尺度によって類似度を計算し，最も距離が近いものを対応付けることで擬似的にパラレルデータを作るという処理となる．具体的には，入力話者と出力話者の音声特徴量系列 $\boldsymbol{X} = \{\boldsymbol{x}_k\}$, $\boldsymbol{Y} = \{\boldsymbol{y}_j\}$ に対して，以下のアルゴリズムを収束まで繰り返すことによってアラインメントを行う．

1. **初期化**　補助入力音声の特徴量系列 $\boldsymbol{X}' = \{\boldsymbol{x}'_k\}$ を定義する．補助入力音声は入力話者の音声 $\boldsymbol{X}$ を初期値とし，学習のイテレーションと共に変換されていく音声である．
2. **最近傍アラインメント**　何らかの距離尺度 $d(\cdot)$ を用い，補助入力音声及び出力話者の音声に関して最も距離の近いフレームの組を計算する．

$$p(k) = \arg\min_j d(\boldsymbol{x}'_k, \boldsymbol{y}_j) \tag{3.11}$$

$$q(j) = \arg\min_k d(\boldsymbol{y}_j, \boldsymbol{x}'_k) \tag{3.12}$$

3. **変換の学習**　2. で得られたアラインメントを用いて入力話者特徴量 $\boldsymbol{X}$ と出力話者特徴量 $\boldsymbol{Y}$ で結合ベクトルを構築し，GMM を用いて同時確率密度の推定を行う．
4. **補助入力特徴量の変換**　学習した GMM を用いて通常の GMM 声質変換のように $\boldsymbol{X}$ を目的話者の音声に近づくように変換を行う．この変換された音声を用いて再び 1. の手順から学習を繰り返す．

この手法では，入出力話者の学習データに関して音素的・発話内容的な制約が無いことが特徴である．しかし，特徴量の対応付けを行うと言う処理の関係上，実際には入出力話者の

学習データ中に同一の音素が出現していなければならないというデータ制約が暗に存在する．このため実際には発話に制約を与えない場合，大量の音声データが無ければミスマッチが生じやすいという問題がある．

3.2.2 話者適応型 Restricted Boltzmann Machine を用いた多対多声質変換

話者適応型 Restricted Boltzmann Machine を用いた多対多声質変換 [35] では，確率モデルの 1 つである Restricted Boltzmann Machine (RBM) を用いた声質変換を拡張することで，全学習過程においてパラレルデータフリーな多対多声質変換を実現している．

2.9 節で示したように，RBM は可視層と隠れ層からなる 2 層のネットワークからなる無向グラフィカルモデルであり，可視層と隠れ層の同時確率分布を表すモデルである [16]．元々はバイナリデータを可視層に与えることを考慮したモデルであったが，連続値を可視層に与えることができるように拡張が行われており，声質変換のような連続値マッピングのタスクではそれらが用いられる [36, 37]．D 次元の連続値を入力として与える場合の可視素子の集合を $\boldsymbol{v} = [v_1, v_2, ..., v_D]$，$n$ 個からなる隠れ素子の集合を $\boldsymbol{h} = [h_1, h_2, ..., h_n]$ としたとき，同時確率 $p(\boldsymbol{v}, \boldsymbol{h})$ は以下の式で表される．

$$p(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{Z} \exp(-E(\boldsymbol{v}, \boldsymbol{h})) \tag{3.13}$$

$$E(\boldsymbol{v}, \boldsymbol{h}) = \| \frac{(\boldsymbol{v} - \boldsymbol{h})}{2\boldsymbol{\sigma}} \| - \boldsymbol{c}^\top \boldsymbol{h} - (\frac{\boldsymbol{v}}{\boldsymbol{\sigma}^2})^\top \mathbf{W} \boldsymbol{h} \tag{3.14}$$

$$Z = \sum_{\boldsymbol{v}, \boldsymbol{h}} \exp^{-E(\boldsymbol{v}, \boldsymbol{h})} \tag{3.15}$$

2.9 節で示したバイナリデータを対象とした RBM と同様に，$\mathbf{W}$ は結合重み，$\boldsymbol{b}$ は可視素子のバイアスパラメータ，$\boldsymbol{c}$ は隠れ素子のバイアスパラメータをそれぞれ表し，$\boldsymbol{\sigma}$ は可視素子の偏差を表す．この GBRBM においても，通常の RBM と同様に条件付確率を定義し，各パラメータを Contrastive Divergence 法によって推定することができる．

[35] で提案されている適応型 RBM は，この GBRBM を拡張したモデルとして提案されている．複数話者の音声によって学習した話者非依存 RBM を初期値とした上で，図 3.4 に適応型 RBM のグラフ構造を示す．図のように，適応型 RBM では可視層と隠れ層の素子に加えて，識別素子 $\boldsymbol{s}$ を導入している．この識別素子は 1 つの要素が 1 で他の要素は 0 となる one-hot vector であり，ベクトルの各要素が入力として与えられた音声 $\boldsymbol{v}$ がどの話者に属するかを表現する．この識別素子によって可視層の素子に与えられた入力がどの話者による発話であるのかを識別し，それによって話者毎の可視素子と隠れ素子の間の結合重みを制御する．話者 k の発話が与えられたときの結合重みの式は以下のようになる，

$$\mathbf{W}(\boldsymbol{s}) = \mathbf{A}_k \hat{\mathbf{W}} + \mathbf{B}_k \tag{3.16}$$

ここで $\mathbf{A}$ および $\mathbf{B}$ は話者非依存な重み行列 $\hat{\mathbf{W}}$ を，特定の話者を対象とした行列へと適応するための行列であり，識別素子によって選択される．この識別素子によって制御される

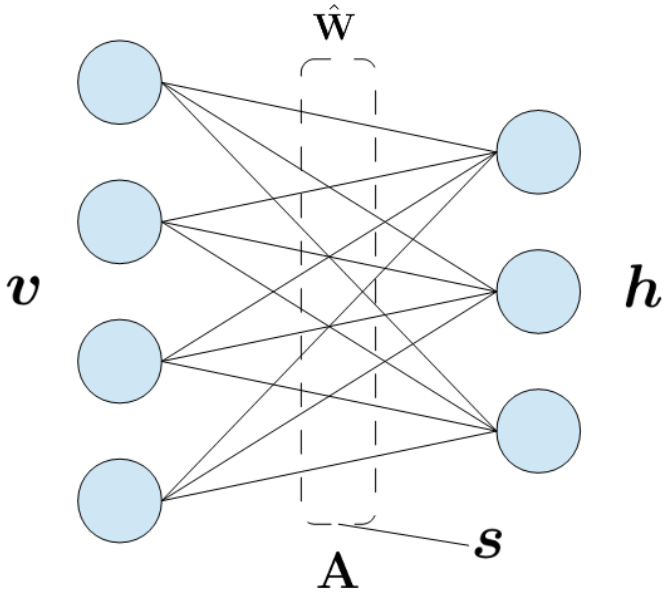

図 3.4: 適応型 RBM のグラフ構造

結合重みを RBM のエネルギー関数に導入することで，通常の RBM と同様に Contrastive Divergence 法によって教師無し適応を行う．適応においては入力された特徴量系列に関して，話者非依存の結合重みは固定した上で，確率的勾配法によって話者非依存結合重みを特定の話者に適応するためのパラメータを推定する．これによって得られた隠れ素子の潜在特徴量から，目的とする出力話者の適応パラメータを用いて出力素子の特徴量を推定することで，最終的な変換を実現する．

第4章

提案手法：複数のサブネットワークを有するDNNに基づく多対一声質変換

4.1　目的

4.1.1　GMM による声質変換と DNN による声質変換

3 章で示したように，GMM においては MAP 適応などを用いたパラメータ適応による多対一および一対多声質変換手法が多く提案されてきた．一方で，近年は増えてきているものの，DNN に関してはそういった多対一や一対多の声質変換を目的とした手法が GMM に比べて非常に少なかった．GMM においてパラメータ適応を扱う手法が多く存在しているのは，適応するべきパラメータが各話者における平均ベクトルや話者間の分散共分散行列というように，それぞれ何を表しているかが分かりやすく，明示的なためである．しかし，DNN では，各層および各ノードの持つ結合重みやバイアス項が声質変換においてどういった情報を持つのかが明示的でなく，GMM のように柔軟なパラメータ適応を行うことが難しい．

一方で，単純な一対一の声質変換においては，ANN を用いた声質変換や DNN を用いた声質変換による変換精度が GMM の変換精度を上回っているという研究が報告されている [5][10]．これは，入力話者および出力話者の音声が発せられる声道の形状は非線形的であることから，非線形的な変換を行う ANN が音声情報を扱うのに適しているからであると考えられる．このことから，DNN を用いた声質変換において話者適応を行うことができれば，GMM によるものよりもより高い精度の変換が可能であると考える．

そこで，本研究では音声を扱うのにより適していると考えられる ANN，DNN の枠組みを用いて一対多や多対一のような柔軟な声質変換を可能とする変換モデルの構築を目的とする．

4.1.2　着想・理論

GMM におけるパラメータ適応手法として挙げた MAP 適応の枠組みを考える．MAP 適応では，あらかじめ用意してある話者ペア (リファレンス話者ペア) からなるパラレルコーパスによって GMM を学習する．そこから分散共分散行列を固定したまま，目的とする話者に近づくように平均ベクトルを更新していく．このとき，適応先の話者のデータ数及びリファレンス話者のデータ数に応じて，それぞれのデータから得られたパラメータ更新の重みを調整する．これによって，少量の適応データしか存在しない場合には事前学習の結果を重点的に使用し，大量の適応データが存在する場合には適応データから得られるパラメータ更新を重視するような学習が可能となっている．この MAP 適応の処理では，分散共分散行列で表される入出力特徴量の各次元間の関係は入力話者と出力話者に大きく依存しない情報であると考え，平均ベクトルのみが入力話者と出力話者に大きく依存すると考えている．

MAP 適応に代表される多くの適応手法では，特定の話者に依存しないと考えられるパラメータに関しては，理想的なデータ (パラレルコーパス) が揃っている話者を用いて予め事前知識として推定しておき，新しい話者に関する変換を行う際には，その新しい話者に依存するパラメータのみを更新するという考え方が共通している．これを GMM のような

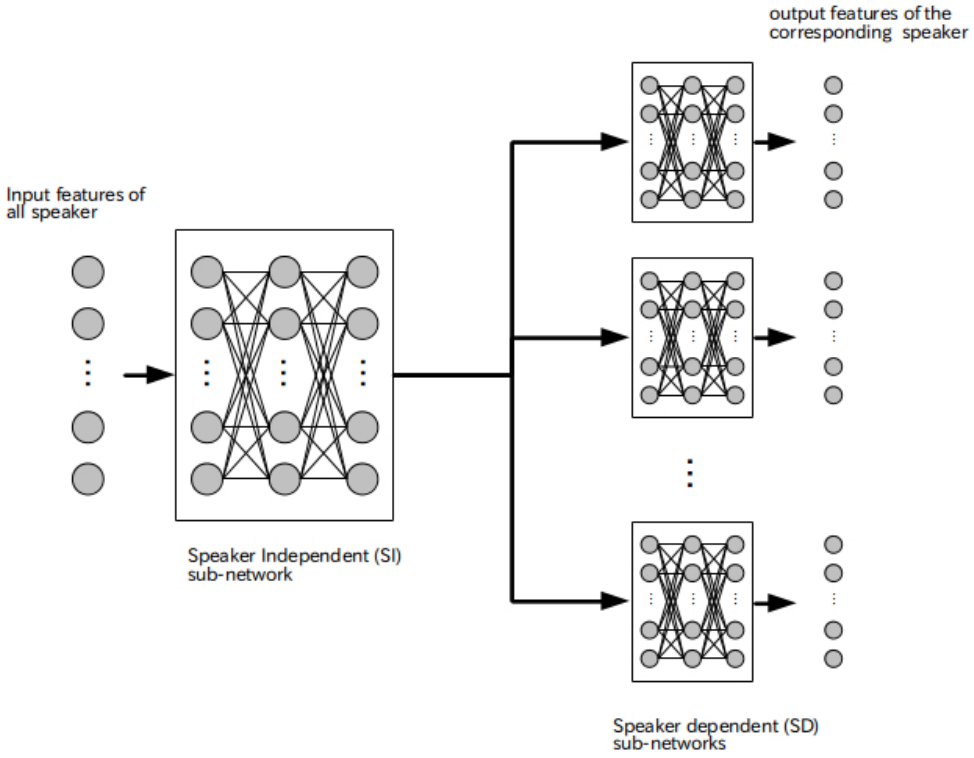

図 4.1: 話者非依存ネットワークと話者依存サブネットワークを用いた提案手法

生成モデルではなく ANN や DNN のような識別モデルに当てはめると，声質変換においては話者非依存な変換処理と話者依存な変換処理を分けて考えることが有効であり，話者非依存な変換処理と話者依存な変換処理を分離することができれば，そこから話者依存な変換処理のみを更新するような柔軟な声質変換が実現できると考えられる．

DNN においてこのような処理を可能とするための枠組みとして，多言語音素認識という複数の言語による音声中の音素を識別するという一種のクラスタリングタスクで提案されている言語非依存サブネットワークの手法を参考にした [13].

4.2　マルチ出力サブネットワークを用いた DNN による声質変換

4.2.1　提案手法の概要

本節では，松田らによる多言語音声を学習した Deep Neural Network における言語非依存サブネットワークの自動適応の手法を参考とした，提案手法であるマルチ出力サブネットワークを持つ DNN の構造について説明する．提案手法では，DNN に対して変換先の出力話者毎に異なる話者依存サブネットワークを導入し，複数の話者からなるコーパスを用

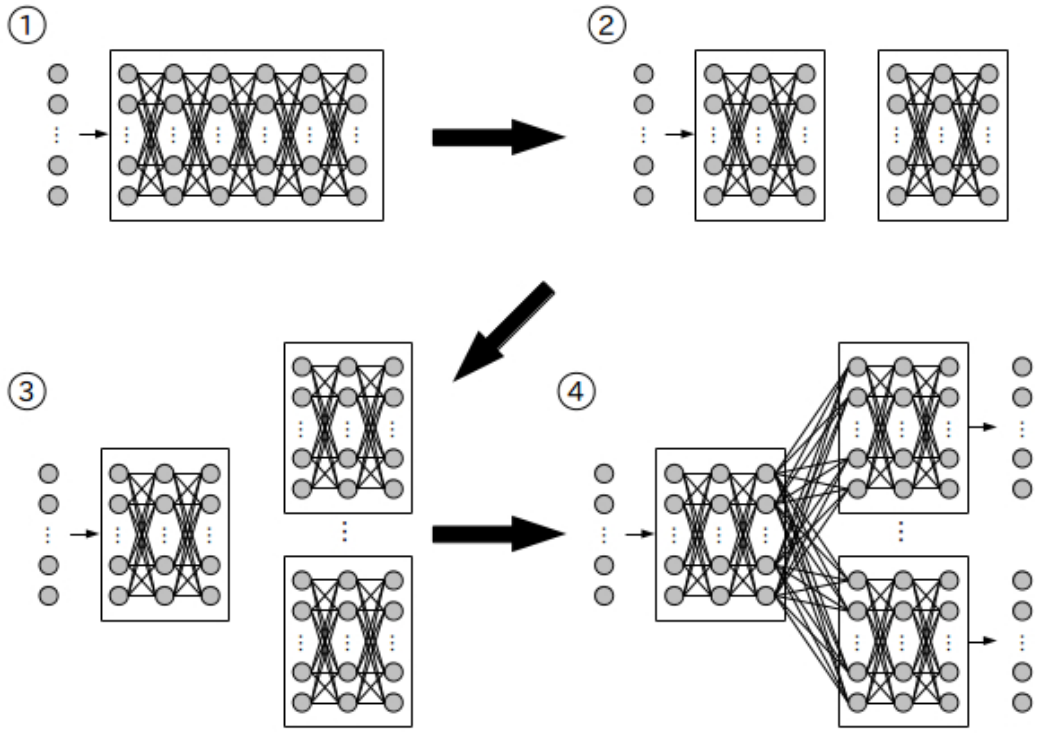

図 4.2: 提案手法における pre-training

いて声質変換モデルの学習を行う．手法の概要を図 4.1 に示す．

提案手法では松田らの手法と同様の仮定を置く，すなわち，Deep Learning では前半の浅い層において, 殆どの音響的事象に共通する時間変動や周波数などの特徴量の識別を行っており, 逆に深い層では音素や言語依存の特徴量などの複雑な情報を扱っていると考えているこの過程を基に，提案手法における DNN は，1) 出力話者に依存しない特徴量抽出器のような処理を行うと考えられる入力層近傍のサブネットワーク (SI サブネットワーク), そして 2) 話者性の再構成のような処理を行うと考えられる出力用の複数のサブネットワーク (SD サブネットワーク)，という 2 つのサブネットワークによって構成される．このネットワークを，複数の話者からなるパラレルコーパスによって学習する．ここに，ある入力話者の特徴量ベクトルを入力することで，複数の SD サブネットワークによって入力話者から各出力話者に対して変換を行った複数の出力特徴量ベクトルが得られる．

i)　提案手法における pre-training

学習では，初めに pre-training の処理を行う．提案手法における pre-training の概要を図 4.2 に示す．pre-training では，通常の Deep Belief Netwotk (DBN) の学習方法と同様に，RBM によってネットワークの層を 1 層ずつ積み上げることで, ANN の各パラメータの初期値を計算する (図中の 1)．このとき，学習データとしてはコーパス中に含まれる全ての

話者の特徴量を使用する．次に，出力層付近の何層かのネットワークを学習データ中の話者の数だけ複製し，話者依存マルチ出力サブネットワークとする (図中の 2，3)．この複数の話者依存サブネットワークを元のネットワークに接続し，途中で枝分かれしているような構造とする (図中の 4)．これにより，1 つの根ネットワークとマルチ出力を為す複数の枝サブネットワークによって構成される DNN が得られる．

ii)　提案手法における fine-tuning

次に，fine-tuning を行う．fine-tuning の際には，ある話者を入力話者とした場合，その話者を含むコーパス中の全ての話者を出力話者と考え，モデルの学習に用いる (コーパスに N 人の話者データが含まれているとすると，一人の話者から N 人の話者への変換を同時に学習する)．これは，特徴量の次元を D とすると，入力 D 次元，出力 ND 次元の変換を行っているとも考えられる．複数の話者の特徴量間でアラインメントを取るため，本手法では，DP マッチングによる特徴量のアラインメントを，各発話毎にデータの系列長が最も長いデータの長さを基準とし，データの系列長が短いデータを引き伸ばすことでマッチングを行う．具体的には，2.4 節で示したアラインメントの経路重みを計算する配列 H を以下の様に変形する．

$$H_{ij} = \min \begin{cases} H_{i-1,\,j-1} & + & 2D_{ij} \\ H_{i,\,j-1} & + & D_{ij} \end{cases}$$

このとき，x 軸 (変数 i) に相当する特徴量系列には各発話毎に最も発話長の長いデータを用いるとする．これにより，全ての話者の特徴量系列の長さを統一し，入力 D 次元，出力 ND 次元の変換として誤差逆伝搬法による学習を行うことができるようになる．

iii)　提案手法の利点

提案手法の pre-training について考える．提案手法の pre-training では 2.9 節で挙げた DBN による声質変換手法と同様に RBM を複数積み上げる手法を用いている．[10] の手法では，深い階層を持つ DBN では各層のノード数で入力特徴量を表現するため，層の数が増えるほど入力特徴量が話者性の薄れた基底集合に近くなると仮定している．これは，一対一の変換では入力側・出力側それぞれの RBM の学習に用いられる特徴量の話者が一定であるため，話者性という学習データに共通の情報が層間の変換によって表され，言語情報 (音素情報) という特徴量のフレーム毎に異なる情報が深い層 (入力から遠い層) に集約されるという仮定である．提案手法の場合を考えると，RBM によって学習する特徴量は複数の話者からなる音声コーパスであるため，学習に用いられる特徴量の話者が一定ではなく，[10] らと同様の仮定を置くことはできない．一方で，入力されるデータが複数の話者からなる音声コーパスであるために，特徴量のフレーム毎に異なる情報，すなわち話者性に関しては RBM の深い層に集約されると考えられる．これにより，通常の 1 話者のデータによる pre-training よりもより話者性をよく表現する特徴量の抽出が可能になっていると考えられる．

提案手法による fine-tuning について考える．提案手法による変換では，複数話者からなる学習データは常に根ネットワークである SI サブネットワークを経て，SD サブネットワークとの結合部分でコーパス中の話者の数に分岐する．一方で，枝ネットワークである SD サブネットワークでは変換先の話者は常に固定である．誤差逆伝搬法によって出力層での誤差が伝搬していくことを考えると，SI サブネットワークと SD サブネットワークとの結合部分では，1 人の入力話者とコーパス中の全ての話者を出力話者と考えたときの誤差が伝搬することとなり，この学習を全ての話者を入力話者として行うことになる．これにより，SI サブネットワークにおける SD サブネットワークとの結合部分付近では，1 人の話者と複数の話者との誤差を最小化するような学習が行われる．そのため，SI サブネットワークの結合部分付近では話者性を正規化，または除去するような変換が学習され，入力された特徴量から話者非依存な特徴量を抽出する一種の特徴量抽出器の働きを持つことが期待される．一方で，各 SD サブネットワークはそのサブネットワークに対応する話者を出力としたデータによってのみ学習されるため，この SD サブネットワークが話者非依存な特徴量から出力話者依存な特徴量の再構成器としての働きをすることが期待される．

また，提案手法は話者変換以外のタスクへの利用も考えられる．例として，声質変換におけるもう 1 つの課題である感情変換・感情付与を考えると，話者変換において複数話者のデータを入力し各話者への変換をそれぞれのサブネットワークで学習していたものを，複数の感情データを入力とし，出力のサブネットワークを個別の感情への変換用として学習を行えばよい．その他にも，出力に新しい話者を加える際には，前半の SI サブネットワークの値は学習済みのものをそのまま使用し，SD サブネットワークのみを pre-training 済みの状態から fine-tuning によって更新することで，計算量を削減することができるなどの応用が考えられる．

4.3　pre-training に用いる話者数による変換精度への影響

提案手法による声質変換の有効性を示すために評価実験を行った．具体的な実験としては，以下のものを行った．

1. pre-training に複数話者のデータを用いることによる変換精度への影響の確認
2. 学習データ中の話者に関する従来手法と提案手法の間の変換精度の比較
3. 提案手法の変換精度が学習に用いる話者数によってどのように変化するかの確認
4. 学習データ外の未知話者に関する従来手法と提案手法の間の変換精度の比較

以下で各実験について示す．

提案手法と通常の DNN の違いとして，pre-traning と fine-tuning の両方に複数話者のデータを用いるという点がある．そのため，提案手法全体の精度を比較しただけでは，その精度の差が pre-training によるものなのか fine-tuning によるものなのか，または両方が組み合わせられたことによるものなのかが判別できない．そこで初めに予備実験として，pre-training に用いる話者数及びデータ数を変化させた際の通常の DNN による一対一声質変換の精度比較を行った．

表 4.1: 学習に使用した各話者のデータ数毎の客観評価結果 ((M1/M2/M3) : 話者 M1，M2，M3 から使用した文の数)

Condition	#utterances	Mel-CD[dB]
A	(50/0/0)	4.293
B	(50/50/0)	4.281
C	(50/0/50)	4.270
D	(50/50/50)	4.270
E	(150/0/0)	4.264

実験においてデータセットとしては，ATR 日本語音声データベースの B セット [26] を使用した．ATR 日本語音声データベースから男性話者 3 名 (以下 M1，M2，M3 とする) を選択し，各話者について学習に 50 文 (subset A)，テストに 50 文 (subset B) の計 100 文を用いた．特徴量としては，STRAIGHT 分析 [27] によって得られたメルケプストラムの 0 次元目 (パワーに相当する) を除く 24 次元を使用した．非周期性指標に関しては今回の実験では変換を行わず，全ての帯域に関して −30dB で固定とした．パワーと基本周波数に関しては，学習データから得られる平均と分散による線形変換によって変換を行った．

実験に使用する DNN の構成としては，層数を 6 層，隠れ層のノード数を 1024 ユニットとした (入力・出力は 24 次元)．変換精度は変換された特徴量と正解データの特徴量の間で，式 (4.1) で表されるメルケプストラム歪み (Mel cepstral distortion:Mel-CD) をとり，客観評価を行った．

$$\mathrm{MelCD[dB]} = \frac{10}{\ln 10}\sqrt{2\sum_{d=1}^{24}(mc_d - \bar{mc}_d)^2} \tag{4.1}$$

ここで，mc_d と $\bar{mc}_d$ は出力話者の正解データの特徴量ベクトルと，入力話者の特徴量ベクトルから変換された特徴量ベクトルをそれぞれ表す．

pre-training に複数話者のデータを用いることによる変換精度への影響を確認するため，学習に用いる話者 M1，M2，M3 のデータ量を 5 通りに変化させ，各条件において pre-training を行った通常の DNN によって声質変換を行った．実際に変換を行う話者は入力話者を M1，出力話者を M3 とした．

4.3.1　結果・評価

各条件における変換の結果を表 4.1 に示す．Condition A から D の結果から，より多くの話者を pre-training に用いた場合の方が変換精度が良くなっていることが分かる．また Condition D と E の結果から，複数の話者のデータを学習に用いた変換精度と入力話者の

データをより多く用いた変換精度がほぼ同程度となっており，pre-training に複数の話者のデータを学習に使用することは，変換に直接関与していない話者であっても有用であると考えられる．

4.4　既存手法との変換精度の比較

4.4.1　目的・実験条件

次に，学習データ中の話者に関する提案手法の変換精度を，既存手法である GMM，DNN と比較することで提案手法の有効性を確認した．

4.3 節の実験条件で述べたように，提案手法と通常の DNN の違いとして，pre-traning と fine-tuning の両方に複数話者のデータを用いるという点がある．pre-training に用いる話者数の増加による変換精度の向上が確認されたため，実験条件を平等にするため，既存手法である DNN の pre-training においても提案手法と同じようにコーパス中の話者を全て用いて学習を行った．

実験では実験 1 と同様に ATR 日本語音声データベースから男性話者 3 名 (以下 M1，M2，M3 とする) を選択し，各話者について学習に 50 文 (subset A)，テストに 50 文 (subset B) の計 100 文を用いた．具体的な実験としては，提案手法と GMM および DNN の 3 手法に対して，話者 M1 から M2 への変換と，話者 M1 から M3 への変換の精度比較を行った．また，pre-training と同様に，提案手法では入力話者として複数の話者を用いて fine-tuning を行っており，学習データの条件が他の 2 手法と異なる．そこで，より正確な比較を行うために，GMM と DNN でも入力話者に関して以下の 2 種類の学習を行ったものを用意した．

- 実際に変換を行う話者間の変換のみを学習 (pair-specific)
- 変換先の話者を固定して残りの 2 話者を入力話者として学習 (target-specific)

例として，話者 M1 から M2 への変換を学習する場合，前者では話者 M1 を入力話者として話者 M2 を出力話者として学習を行い，後者では話者 M1 を入力話者として話者 M2 を出力話者とした学習に加えて話者 M3 を入力話者，話者 M2 を出力話者とした変換の学習も行う．

提案手法では，SD サブネットワークの複製を行った後に，話者 M1，M2，M3 をそれぞれ入力話者と出力話者の両方に使用し，fine-tuning を行った．

実験に使用する DNN の構成としては実験 (1) と同様に層数を 6 層，隠れ層のノード数を 1024 ユニットとし (入力・出力は 24 次元)，SI サブネットワークと SD サブネットワークをそれぞれ 3 層とした．

4.4.2　結果・評価

表 4.2 に各手法の客観評価の結果を示す．表 2 から，従来手法である GMM と DNN とで異なる傾向が見られる．GMM では，target-specific な学習を行ったモデルが pair-specific な学習を行ったモデルに対して変換精度が下回っているのに対して，DNN では 2 種類のモ

表 4.2: 3 手法による声質変換の客観評価結果．(Pair-specific：実際に変換を行う話者間の変換のみを学習したモデル，Target-specific：変換先の話者を固定して残りの 2 話者を入力話者として学習したモデル)

Methods	Mel-CD [dB] (M1 to M2)	Mel-CD [dB] (M1 to M3)
GMM (pair-specific)	4.324	4.313
GMM (target-specific)	4.417	4.571
DNN (pair-specific)	4.290	4.270
DNN (target-specific)	4.290	4.241
Proposed	**4.256**	**4.200**

デルがほぼ同等の変換精度か，target-specific な学習を行ったものの方が僅かに上回っている．この結果は，複数の話者を用いることで DNN の浅い層が効果的に学習されているためと考えられる．また，提案手法と他の 2 手法の変換精度を比較すると，話者 M1 から M2 への変換と話者 M1 から M3 への変換の両方において提案手法が従来手法を上回っていることが分かる．

4.5　学習に用いる話者数による提案手法の変換精度の比較

4.5.1　目的・実験条件

提案手法において，学習する話者の数が変換精度にどのような影響を与えるかを確認するため，pre-training と fine-tuning に用いる話者数を 3 話者，6 話者，9 話者とした場合 (それぞれ spk3，spk6，spk9 とする) の変換精度の比較を行った．コーパスとしては多数話者データベース[1]から音素バランス文 50 文 (APP-BLA) を使用し，男性話者 9 名の各話者 50 文のデータを，学習 40 文とテスト 10 文に分けて実験を行った．

また，メルケプストラム歪みによる客観評価は入力話者と出力話者によって値の範囲にばらつきがあるため，各条件の比較を行う場合には，比較を行う条件間で共通して用いられている話者ペアに関して，メルケプストラム歪みの平均を取ったものによって比較を行った．例として，spk6 と spk9 の比較を行う際には，spk9 の変換精度は spk6 で使用している 6 話者に関する変換結果を平均したものとし，残りの 3 話者の結果は含まないものとする．

実験に使用する DNN と提案手法の構成としては実験 (1) と同様に層数を 6 層，隠れ層のノード数を 1024 ユニットとし (入力・出力は 24 次元)，SI サブネットワークと SD サブネットワークをそれぞれ 3 層とした．

表 4.3: 学習に用いる話者数の変化に対する提案手法の客観評価結果 (テストデータ 3 話者)

#speakers	Mel-CD [dB]
spk3	4.637
spk6	4.459
spk9	4.460

表 4.4: 学習に用いる話者数の変化に対する提案手法の客観評価結果 (テストデータ 6 話者)

#speakers	Mel-CD [dB]
spk3	-
spk6	4.547
spk9	4.520

4.5.2　結果・評価

表 4.3 に 3 話者 (6 通り) の変換に関するメルケプストラム歪みの平均値を，表 4.4 に 6 話者 (30 通り) の変換に関するメルケプストラム歪みの平均値をそれぞれ示す．表 4.3 および表 4.4 から，多くの話者を学習に用いたモデルの方がより精度の高い変換を行っており，提案手法の学習に複数の話者を用いることの有効性が示されている．

4.6　未知話者に関する変換精度の比較

4.6.1　目的・実験条件

次に，学習データ外の話者 (未知話者) に関する提案手法の変換精度を，既存手法である GMM，DNN と比較することで提案手法の有効性を確認した．

学習データとしては実験 3 と同様に，多数話者データベースから音素バランス文 50 文 (APP-BLA) を使用し，男性話者 10 名 (M1, M2, ..., M10 とする) の各話者 50 文のデータを，学習 40 文とテスト 10 文に分けて実験を行った．この話者 10 名のうち 9 名を学習に使用し，残りの 1 名 (M10) を未知話者とした．具体的な実験としては，提案手法と DNN の 2 手法に対して学習に用いていない未知話者 M10 から話者 M1，M2，M3 それぞれへの変換の精度比較を行った．

DNN の pre-training には話者 M1 から話者 M9 の 9 話者のデータを使用した．DNN の fine-tuning に関しては実験 2 と同様に DNN における入力話者に関して以下の 2 種類の学習を行ったものを用意した．

- 特定の話者からの変換のみを学習 (pair-specific)

[1]http://www.atr-p.com/products/sdb.html#MS

表 4.5: 提案手法と DNN における未知話者入力に対する声質変換の客観評価結果．(Pair-specific：実際に変換を行う話者間の変換のみを学習したモデル，Target-specific：変換先の話者を固定して残りの 2 話者を入力話者として学習したモデル)

Methods	Mel-CD [dB] (M10 to M1)	Mel-CD [dB] (M10 to M2)	Mel-CD [dB] (M10 to M3)
DNN (pair-specific)	4.869	5.457	4.946
DNN (target-specific)	4.794	4.968	4.685
Proposed(spk3)	4.832	5.057	5.267
Proposed(spk6)	**4.701**	4.935	4.582
Proposed(spk9)	4.715	**4.908**	**4.547**

- 変換先の話者を固定して残りの 8 話者を入力話者として学習 (target-specific)

ただし，pair-specific に関しては出力話者が M1 のときは M2 を入力話者とし，それ以外の話者が出力話者のときは M1 を入力話者とした．例として，話者 M10 から M1 への変換を考えた場合，前者では話者 M2 を入力話者として話者 M1 を出力話者として学習を行い，後者では話者 M2 から話者 M9 までを入力話者，話者 M1 を出力話者とした学習を行う．

提案手法の pre-training にも同様に話者 M1 から話者 M9 の 9 話者のデータを使用し，fine-tuning では，学習に使用する話者数を 3 話者，6 話者，9 話者の 3 条件 (それぞれ spk3，spk6，spk9 とする) に変化させ，それぞれ入力話者と出力話者の両方に使用して学習を行った．

実験に使用する DNN と提案手法の構成としては実験 1 と同様に層数を 6 層，隠れ層のノード数を 1024 ユニットとし (入力・出力は 24 次元)，SI サブネットワークと SD サブネットワークをそれぞれ 3 層とした．

4.6.2　結果・評価

表 4.5 に DNN と提案手法における未知話者を入力とした際の話者 M1，M2，M3 への変換精度を示す．表 4.5 から，話者 M1，M2，M3 の全てに対する変換において，6 話者・9 話者を学習に用いた提案手法が従来手法である DNN の変換精度を上回っている．

DNN においては，実験 2 での傾向と同様に，pair-specific な fine-tuning よりも target-specific な fine-tuning を行った場合の方が変換精度が高くなっている．この結果からも，DNN の学習に複数の話者を用いることの有効性が示されているといえる．

一方で，提案手法において学習に 3 話者を用いたものは，target-specific な DNN よりも精度が悪くなっている．これは，学習に用いる話者数が十分でないときには局所解に陥りやすく，未知話者と話者性が近い話者を学習に用いているかが変換結果に大きな影響を与えるためだと考えられる．

また，話者 M1 への変換では 6 話者を用いて学習を行ったときよりも，9 話者で学習を行ったときの方が変換精度が悪くなっている．これは，実験 3 でも同様の傾向が見られていることから，コーパス中の話者 M7 から M9 の中に話者 M1 への変換が難しく，他の話者

からの変換に悪影響を与えている話者がいるためだと考えられる．

4.7　まとめ

本研究では，話者性の柔軟な制御を目的とした DNN による声質変換手法を提案した．話者性の柔軟な制御を目的とした手法としては，既存のモデルからパラメータを適応するというものが一般的であり，声質変換においては GMM による実装が多く取られている．この GMM による MAP 適応などの話者適応型声質変換手法を例に挙げ，複数話者コーパスを用いた学習を行うことで話者に依存していない，音声に共通な特徴量を抽出することが有用であると考えた．

また，DNN を用いた声質変換手法として，Deep Belief Nets による低次元空間表現を用いた声質変換という手法を挙げ，そこで用いられている，「深い階層を持つ DBN では各層のノード数で入力特徴量を表現するため，層の数が増えるほど入力特徴量が基底集合に近くなる」という考えを参考とした．この考えを仮定すると，DNN の pre-training において入力されるデータを複数の話者からなる音声コーパスとすることで，特徴量のフレーム毎に異なる情報，すなわち話者性が RBM の深い層に集約されると考えた．

そこで，多言語音素認識タスクに用いられている手法を参考に，1 つの話者非依存サブネットワークと複数の話者依存サブネットワークからなる DNN による声質変換の枠組みを提案した．この手法では pre-training を複数の話者によって行うことで，前述したように話者性を深い層に集約し，その上で fine-tuning を出力話者毎の話者依存サブネットワークと話者非依存サブネットワークを導入して行うことで，話者非依存サブネットワークと話者依存サブネットワークの分岐点に集約した話者性を正規化するように学習を行う．これにより，学習に用いていない未知話者入力に対しても話者性が正規化されるために，柔軟な変換が可能となると考えられ，また，ANN において問題であった入力層に近い浅い層の学習をより効率的に行うことが可能となると考えられる．

実験の結果，pre-training に複数話者を用いることによる変換精度の向上と，提案手法による学習データ中の話者と未知話者の両方の入力に対する変換精度の向上を確認し，提案手法が入力話者に対して柔軟な変換が可能であることを示した．

一方で本手法の課題としては，未知話者への頑健性が入力側にしか担保されておらず，変換先の話者に関してはパラレルデータを用いて学習を行わなければ適応できないという点が挙げられる．すなわち，大量の事前収録話者のパラレルデータを用いた学習によって，多対多声質変換を実現している固有声変換などの手法と比較すると，学習データへの制約が強いという問題がある．

そこで，本手法で得た知見の応用として，固有声変換と DNN を組み合わせた変換手法が考えられる．固有声変換やテンソル表現を用いた声質変換では，ある出力話者のモデルは複数の話者モデルの足し合わせによって表現できるとされている．例として，固有声変換では GMM 中の各ガウス分（eigenvoice conversion）布の平均ベクトルを，「全事前収録話者の平均値」と「平均値からの偏差」に分離している．この平均値からの偏差を，話者に依存しない共通の直交基底の重み付け足し合わせによって表現する．これは，通常の GMM

の学習で得られる平均ベクトルに相当するパラメータを，複数の基底 (に話者依存重みを掛けたもの) と平均値に分解していると解釈することができる．

この大量の話者のデータを利用することで，話者非依存な分解を行うと言う枠組みは，本手法において浅い層で行われていると仮定した処理に相当している [11, 12]．この処理に関して DNN を用いることでより高精度かつ柔軟な分解を行うことが考えられる．本手法及び [24] において示されている結果から，複数話者を入力として変換を学習した DNN は，未知話者に関しても適応を行うことなく変換を行うことができるという知見が得られている．そこで，固有声変換によって求められるパラメータを用いることで，入力側だけでなく，出力に関しても少量のデータで適応可能な，多対多声質変換を考える．

第5章

提案手法：EVGMMに基づく話者空間基底を用いたDNN声質変換手法

5.1　目的

5.1.1　多対多声質変換手法に共通する枠組み

3 章で挙げた多対多声質変換手法に共通する仕組みとして，大量の話者を用いて学習したモデルを初期値として少量のパラメータによって話者性を表現し，そのパラメータを少量のデータによって教師無し適応することで未知話者を用いた変換を実現しているという点が挙げられる．一方で，4 章で提案した複数のサブネットワークを用いた多対一声質変換では，大量の話者による学習によって DNN の入力側に汎化性能を与えているものの，出力側の話者に関しては教師無しの適応を行うことができなかった．

また，3 章では既存の DNN 多対多声質変換として，AVM と i-vector を用いた手法を挙げた．AVM と i-vector を用いた多対多声質変換手法では入力話者と出力話者の情報を i-vector によって与えることで，複数の入力話者から複数の出力話者への変換を 1 つの DNN によって実現していた．しかし，このモデルでは入力として与えた 2 話者の i-vector が DNN 中の実際の変換にどのような影響を与えているかは分からず，あくまで話者情報を与えてそれが変換に有効に働くことを期待しているだけであり，与えた i-vector によって制御される変換は考慮していない．また，学習及び変換の両面において i-vector が必要であり，実際に変換を行うネットワークとは別に i-vector 抽出器を保持しておく必要がある．また，変換の際にも i-vector の抽出をその都度行う必要がある．

そこで，本研究ではより明示的に話者性の制御を行う DNN による多対多声質変換を可能とする変換モデルの構築を目的とする．

5.1.2　着想・理論

3 章で挙げた eigenvoice conversion の枠組みでは，予め用意された複数の話者 (事前収録話者と呼ぶ) の音声によって学習を行った話者非依存 GMM を用いることで，学習時に用いる全ての話者を「平均話者」と「平均話者からの偏差」の加算モデルとして表現している．ここで後者について，共通の話者基底と対象話者に依存した重みの積で表した因子モデルとすることで，話者の特徴量空間を定義する．これにより，各話者基底に対応する重みという少数のパラメータ更新による対象話者のモデル化を実現している．このパラメータ更新は教師無し学習によって行うことができ，少量のデータによって任意の話者を声質変換の入力および出力として使用することができる．本研究ではこの固有声変換の枠組みを DNN に導入することで，より高精度かつ柔軟な声質変換を目的とする．

提案手法では，eigenvoice conversion において用いられていた固有声 GMM (Eigenvoice GMM:EVGMM) と DNN とを組み合わせることで，任意の入出力話者を対象とした多対多声質変換変換を行う．初めに EVGMM のパラメータを用いて，全事前収録話者の特徴量を「平均話者」と「話者基底成分」に相当する特徴量へと変換する．この「平均話者」と「話者基底成分」に相当する特徴量と，元々の特徴量の組み合わせを擬似的なパラレルコーパスの様に使用することで，DNN によって元々の特徴量から「平均話者」と「話者基底成分」への分解を行うような変換を学習する．最終的な出力特徴量はこの分解された特徴量

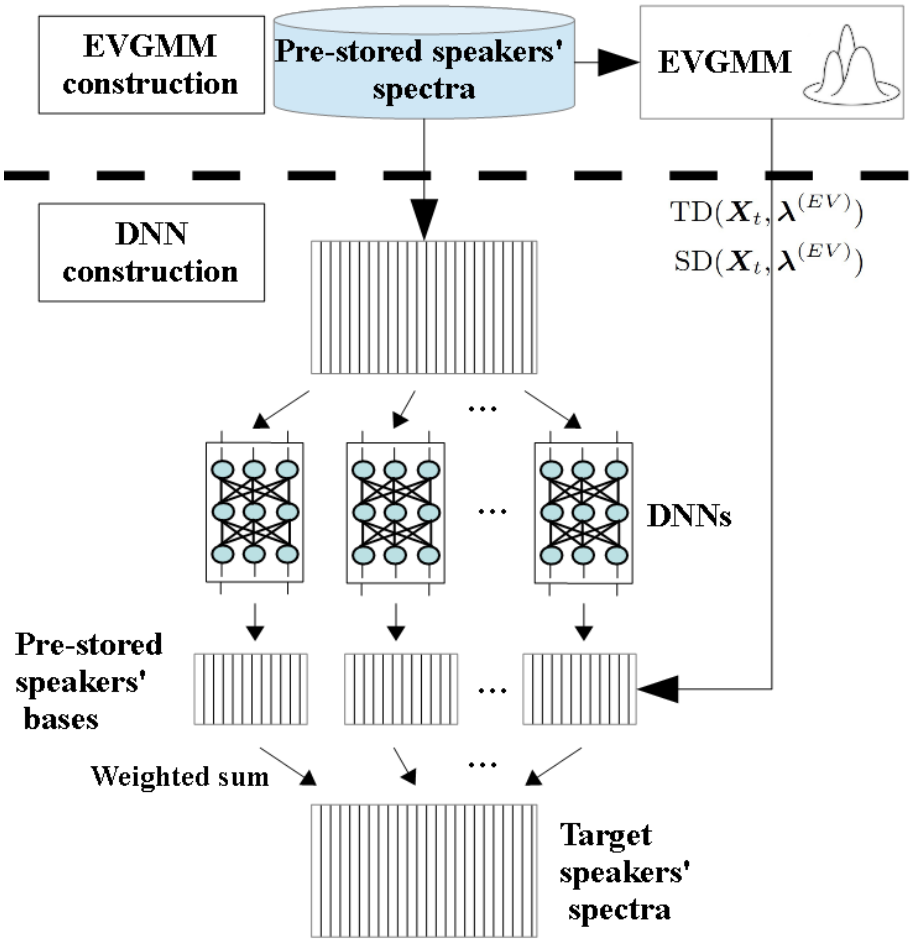

図 5.1: EVGMM に基づく話者空間基底を用いた DNN 声質変換の概要

と，出力話者固有の重みの積によって表現される．任意話者変換を扱う場合の未知出力話者の重みに関しても，教師無し適応によって求めることができる．

5.2　EVGMM に基づく話者空間基底を用いた DNN 声質変換

本研究では，EVGMM の枠組みを用いて特徴量を各話者空間基底へ分解し，その重み付け和によって特定の話者の特徴量へ変換する手法を DNN を用いて実装する．提案手法の概要を図 5.1 に示す．提案手法の学習は大きく 2 つの処理に分けられ，1) EVGMM を学習し，全事前収録話者の特徴量を話者空間基底へ射影，2) 全事前収録話者の特徴量から話者空間基底への分解を行う DNN と話者空間基底から出力話者の特徴量への再構成を行う変換を学習という処理からなる．以下で各処理について示す．

5.2.1　EVGMM による話者空間基底への射影

結合特徴量をモデル化した一対多 EVGMM による声質変換を考えたとき，入力話者 X から出力話者 Y への最小 2 乗誤差基準による変換は式 (5.1) で表される．

$$F(\boldsymbol{x}_t)=\sum_{m=1}^{M}\gamma_{m,t}(\mathbf{B}_m\boldsymbol{w}^{(Y)}+\boldsymbol{b}_m^{(0)}+\mathbf{A}_m(\boldsymbol{x}_t-\boldsymbol{\mu}_m^{(X)})) \tag{5.1}$$

ただし $\boldsymbol{x}_t$ は入力話者 X の特徴量，$\gamma_{m,t}$ は入力話者 X の特徴量から得られる各混合の事後確率 $P(m \mid \boldsymbol{x}_t, \lambda^{(EV)})$ である．また，$\mathbf{A}_m = \boldsymbol{\Sigma}_m^{(YX)}\boldsymbol{\Sigma}_m^{(XX)-1}$ である．

変換式 (5.1) で使用している共分散行列以外のパラメータは，単一話者空間で学習を行った EVGMM でも学習過程で得ることができるため，$\mathbf{A}_m$ をハイパーパラメータとすればこの変換式を単一話者空間で学習を行った EVGMM にも適用することができる．このとき，平均ベクトル $\boldsymbol{\mu}_m^{(X)}$ は EVGMM のパラメータを計算する過程で計算された，話者依存 GMM のパラメータから得られる．$\gamma_{m,t}$ も同様に話者依存 GMM から計算される．$\mathbf{A}_m = \boldsymbol{\Sigma}_m^{(YX)}\boldsymbol{\Sigma}_m^{(XX)-1}$ をハイパーパラメータと考えると，式 (5.1) は以下のように変形できる．

$$F(\boldsymbol{x}_t, \lambda^{(EV)})=TD(\boldsymbol{x}_t, \lambda^{(EV)})+SD(\boldsymbol{x}_t, \lambda^{(EV)}) \tag{5.2}$$

$$\begin{aligned} TD(\boldsymbol{x}_t, \lambda^{(EV)}) &= \sum_{k=1}^{K}\boldsymbol{w}_k^{(Y)}\sum_{m=1}^{M}\gamma_{m,t}\mathbf{B}_{m,k} \\ SD(\boldsymbol{x}_t, \lambda^{(EV)}) &= \sum_{m=1}^{M}\gamma_{m,t}\{\boldsymbol{b}_m^{(0)}+\mathbf{A}_m(\boldsymbol{x}_t-\boldsymbol{\mu}_m^{(X)})\} \end{aligned}$$

$\boldsymbol{w}_k^{(Y)}$ は重みベクトル $\boldsymbol{w}^{(Y)}$ の k 要素目，$\mathbf{B}_{m,k}$ は話者空間基底群 $\mathbf{B}_m$ の k 列目をそれぞれ表す．EVGMM において $\boldsymbol{b}_m^{(0)}$ は各混合における全ての事前収録話者の平均を表しているため，$TD(\lambda^{(EV)})$ と $SD(\boldsymbol{x}_t, \lambda^{(EV)})$ は，出力話者依存の成分と入力話者から平均話者への射影成分をそれぞれ表していると言える．

この式を用いて入力話者の特徴量を各話者空間基底の成分に分解することを考える．7 章で示したように，変換先の出力話者の話者性は重みベクトル $\boldsymbol{w}^{(Y)}$ に現れ，$\boldsymbol{w}^{(Y)}$ は，出力話者に関して，話者空間を構築する基底 $\mathbf{B}_m$ の各要素の重みを表している．そのため，$\boldsymbol{w}^{(Y)}$ を特定の要素のみ 1 で他が 0 となる 1-of-K 表現で表されるベクトルとすることで，入力話者の特徴量を特定の基底成分のみを持つ特徴量に変換することができる．よって，入力話者 X の特徴量 $\boldsymbol{x}_t$ の各基底成分を $\boldsymbol{E}_t^k$ とすると，$TD((\boldsymbol{x}_t, \lambda^{(EV)})$ と $SD(\boldsymbol{x}_t, \lambda^{(EV)})$ から，$\boldsymbol{E}_t^k$ は以下の式で表される．

$$\boldsymbol{E}_t^k = \sum_{m=1}^{M}\gamma_{m,t}\mathbf{B}_{m,k} \qquad (k=1,2,...,K) \tag{5.3}$$

$$\boldsymbol{E}_t^0 = \sum_{m=1}^{M}\gamma_{m,t}\{\boldsymbol{b}_m^{(0)}+\mathbf{A}_m(\boldsymbol{x}_t-\boldsymbol{\mu}_m^{(X)})\} \tag{5.4}$$

ここで，$k = 0$ はバイアス成分を表す．式 (5.3)(5.4) によって，入力特徴量を各話者空間基底上に射影する．また，式 (5.2) より，式 (5.3) から得られた各話者空間基底成分に対して出力話者の重みベクトルの各要素を掛けて式 (5.4) と足し合わせることで任意の出力話者への変換を行うことができる．

提案手法では，式 (5.3)(5.4) を用いることで得られる，入力特徴量と各話者空間基底成分の組をパラレルデータと考え，そのデータ間の変換を DNN によって学習する．1 つの話者空間基底成分への射影は，入力特徴量から基底成分を抽出する処理と捉えられ，入力話者に非依存の形で DNN を構成できると考えられる．そこで，基底と同じ数の DNN を用いて全ての事前収録話者に関して基底成分への射影を学習することで，未知話者入力にも対応可能な，柔軟な射影が実現できると考えられる．

5.2.2　DNN を用いた基底への射影および特徴量変換

次に「入力話者の特徴量から話者空間基底への分解を行う DNN，話者空間基底から出力話者の特徴量への再構成を行う変換を学習」に当たる処理について示す．

EVGMM の学習に使用した全事前収録話者を入力特徴量として，式 (5.3)(5.4) によって各基底成分に射影した特徴量との間でパラレルデータを構築する．このパラレルデータを用いることで，入力特徴量を各基底上に射影する DNN を基底の個数分だけ学習する．学習を行った DNN に対して，入力話者の特徴量を入力することで，複数のネットワークによって入力話者から各話者空間基底への射影を行い，得られた基底に対して出力話者依存の重みを掛けた足し合わせを行うことで出力話者への変換を実現する．この出力話者 s への変換 $f(\boldsymbol{x}_t)$ は k 個目の基底に関する DNN を $DNN^{(k)}$，とすると，以下の式で表される．

$$f(\boldsymbol{x}_t) \;=\; \sum_{k=1}^{K} w_k^{(s)} DNN^{(k)}(\boldsymbol{x}_t) + DNN^{(0)}(\boldsymbol{x}_t) \tag{5.5}$$

このとき，式 (5.2) においてハイパーパラメータとした $\mathbf{A}_m$，基底成分への変換を行う DNN^k による変換誤差を考えると，EVGMM で得られた $\boldsymbol{w}^{(s)}$ をそのまま用いるのは適していないと考えられる．そこで，より精度の高い変換となるように $\boldsymbol{w}^{(s)}$ の値を初期値として，DNN のパラメータは固定した上で重みの更新を行う．この学習は，正解データである出力話者の特徴量を $\boldsymbol{y}_t$ とすると，$\|\boldsymbol{y}_t - f(\boldsymbol{x}_t)\|^2$ を最小化する $\boldsymbol{w}^{(s)}$ を求めるものとなる．このとき，基底成分への分解を行う DNN のパラメータは固定されているため，入力話者と出力話者に同一話者のデータを用いることで，auto-encoder のような教師無し学習を行うことが出来る．未知話者 u への変換を行う際を考えた場合も，各基底成分に対する重み $\boldsymbol{w}^{(u)}$ のみを学習することができれば変換を行うことができる．この場合も DNN のパラメータを固定した上で auto-encoder のように未知話者の特徴量を再構成するように学習を行うことで，未知話者に対応する話者重みを推定することができる。これにより，未知の出力話者に対して適応を行う際にも，パラレルデータを必要とせず，出力側の線形変換の学習のみによって変換を実現することができる．

実際に変換を行う DNN 部分に関しては，入力側では基底成分・平均成分への分解は話者非依存なものであると仮定し，全事前収録話者で分解の学習を行うことで適応を行うことなく未知話者への汎化性能を向上させる．出力側では，分解した基底・平均成分の重み付き足し合わせによって変換を行っており，この重みは DNN のパラメータを固定した状態で入出力に同一の音声特徴量を用いて学習を行うことで教師無し適応が可能である．これらの枠組みによって，少量のデータで適応可能な多対多声質変換を実現する．

5.2.3　提案手法の利点

本手法の利点としては，出力話者固有の重みの学習をパラレルデータフリーに行うことが可能であることに加えて，入力に関しては一切適応を行うことなく未知話者に対する変換を実現できるということが挙げられる．また，AVM と i-vector を組み合わせた手法のような i-vector を用いた手法と比較すると，学習の過程では GMM によって i-vector を計算する処理が必要となるものの，1 度モデルを構築することができれば GMM における i-vector に相当する話者重みを DNN によって推定することができるという利点がある．また，4 章で示した実験結果や [24] において，DNN の学習の際に複数の話者を入力として用いることで未知話者入力への変換精度が向上するという報告がされており，本手法においても同様の効果が期待される．

5.3　EVGMM と DNN を組み合わせた多対多声質変換に関する実験

5.3.1　実験条件

提案手法による一対多声質変換の評価を行うため，既存手法との比較実験を行った．提案手法と比較を行う手法は EVGMM，AVM，及び GMM の 3 種類とした．ここで，AVM は 3 章で示した枠組みにおいて，入力に使用する話者表現として i-vector の変わりに EVGMM における話者重みを使用している．GMM は，教師あり学習を行ったモデルとの差を確認するための比較手法であり，この学習には最終的な変換対象の話者のパラレルデータを使用している．また，比較のために，提案手法においてパラレルデータを用いて出力層の話者依存重みを推定した場合の結果も確認した．実験としては，入力話者 1 名と出力話者 10 名の両者を学習データ外の未知話者とした場合の声質変換の精度比較を行った．それぞれの手法で用いる EVGMM の混合数は 256 とし，分散行列と相互共分散は対角行列とした．同様に，各条件で用いる DNN は隠れ層の数を 5，隠れ層のノード数を 512 とし，学習回数は 50 epoch とした．

データセットとして，ATR 日本語音声データベースの B セット [26] を使用した．出力対象の話者には JNAS から男性話者 5 名と女性話者 5 名の合計 10 名を使用し，各話者について，適応に使用する文の数を 2 文から 32 文まで変化させ，変換精度の平均を比較した [31]．EVGMM 及び提案手法の学習過程で構築する EVGMM の学習コーパスとしては，

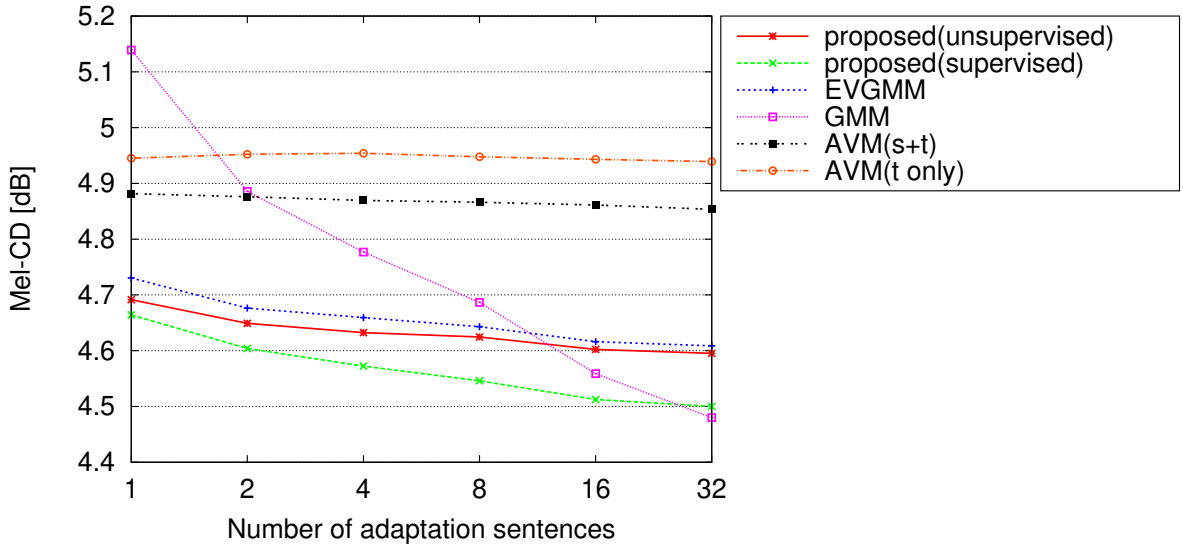

図 5.2: 10 話者への変換に関するメルケプストラム歪みを用いた客観評価結果

JNAS から 96 話者の音素バランス文 50 文を使用した．また，提案手法におけるハイパーパラメータ **A** としては，一対多 EVGMM における話者非依存分散と一対多共分散を使用した．AVM においては，EVGMM の学習と同一の 96 話者 50 文に関して，同一の音素バランス文を含む 1010 組の入出力話者ペアによって学習を行った．変換を行う際の入力話者としては，EVGMM の学習に使用したアンカー男性話者 1 名を使用した．特徴量としては，STRAIGHT 分析 [27] によって得られたメルケプストラム 24 次元とそのデルタ特徴量を使用し，変換精度の比較には以下の式で表されるメルケプストラム歪み (Mel cepstral distortion:Mel-CD) を用いた．

$$\mathrm{MelCD[dB]} = \frac{10}{\ln 10}\sqrt{2\sum_{d=1}^{24}(mc_d - \bar{mc}_d)^2} \tag{5.6}$$

ここで，mc_d と $\tilde{mc}_d$ は出力話者の正解データの特徴量ベクトルと，入力話者の特徴量ベクトルから変換された特徴量ベクトルをそれぞれ表す．

5.3.2　客観評価実験結果

図 5.2 に各適応文数に対する各条件のメルケプストラム歪みを示す．proposed(unsupervised) は教師無し適応によって求めた重みを使用した提案手法，proposed(supervised) はパラレルデータによって重みのみを更新した提案手法をそれぞれ表す．AVM(s+t) は変換元・変換先の両話者の話者重みを入力として使用した AVM，AVM(t only) は変換先の話者重みを入力として使用した AVM をそれぞれ表す．ここで，EVGMM は変換先の話者に関しては未知話者を使用しているが，変換元に関しては学習時にアンカー話者として使用しており既知となっている．加えて，変換精度の参考として入力話者 1 名と出力話者 10 名の全 10 組に

対して，一対一の教師あり学習を行った GMM の変換結果を GMM として図に示した．この際，GMM の混合数は 2 から 256 まで変化させ最も精度の高かった結果を図示している．

初めに一対多変換手法間の全体的な傾向に注目すると，結果から，proposed(unsupervised) は入力話者に関して一切の適応を行っていないにも関わらず，AVM(s+t) と EVGMM を上回る変換精度を実現している．この結果から，提案手法による話者空間への分解及び再構築が有効に働いていると言える．

次に，proposed(unsupervised) と AVM(t only) の結果に注目する．この 2 つの手法は，未知の入力話者に関して適応を行わずに変換を行っているという点で共通している．しかし，結果から，出力話者の適応文数に依らず，proposed(unsupervised) が AVM(t only) を上回っていることが分かる．サブネットワークを用いた提案手法において得られていた，DNN では複数の話者を入力として学習することで汎化性能が向上するという性質は共通していることを考えると，この proposed(unsupervised) が AVM(t only) の差異は「直接変換するのではなく話者空間基底への分解を経由している」,「出力層において話者表現を使用している」という 2 点から生じていると考えられる．特に，話者空間基底への分解は，任意の話者から話者空間の基底成分のみを抽出するという変換にあたり，DNN に学習させる処理が特徴量抽出に近い処理になっていると言える．これにより，DNN が学習する変換は話者空間の特定成分の抽出という入力話者に依存しない処理となり，その結果，複数話者で学習したことによる汎化性能の向上がより効果的になっていると考えられる．対して，AVM では付加した話者重みの情報のみによって変換目標の話者へ直接変換を試みている．その結果，DNN が行う処理は入力話者から出力話者への直接の射影となるため，入力話者の情報を明示的に与えなければ効果的な変換を行うことができていないと考えられる．

次に，教師あり学習を行っている GMM の結果に注目する．GMM の結果は教師有り学習を行っていることから適応文数の増加に対応する精度のゲインは最も大きくなっている．そのため，何文の適応データによって一対多変換手法を上回るかによって，その一対多変換手法の有用性を比較することができる．結果から，適応データが少ない場合に一対多変換手法が GMM を上回っていることは共通しているが，全体的に AVM を用いた結果に比べて EVGMM 及び提案手法が高い変換精度となっていることが確認できる．このことから，事前収録話者から事前知識を獲得する上で，入力層に話者表現を直接与えるという枠組みでは十分な結果が得られないと言える．この原因としては，複数話者間の変換を 1 つのネットワークで直接モデル化するのは不適当である，出力話者の話者重みを入力層という浅い層で与えているため出力層まで十分に出力話者の情報が伝達していない，などが考えられる．

また，提案手法において教師有り学習によって重みを更新した結果である proposed(supervised) に注目すると，他の一対多変換手法と比べて適応データに対する変換精度のゲインは大きくなっているが，同じく教師有り学習を行っている GMM と比べるとゲインは及ばず，32 文の段階で GMM を下回るという結果が確認できる．これは，提案手法では出力層における話者重みという少量のパラメータしか適応していないことが原因であると考えられる．改善案として，十分な量の学習データが存在する場合には，話者空間基底への分解を行う DNN のパラメータも一部更新することや，出力層での重み付け和を拡張して各次元毎に重

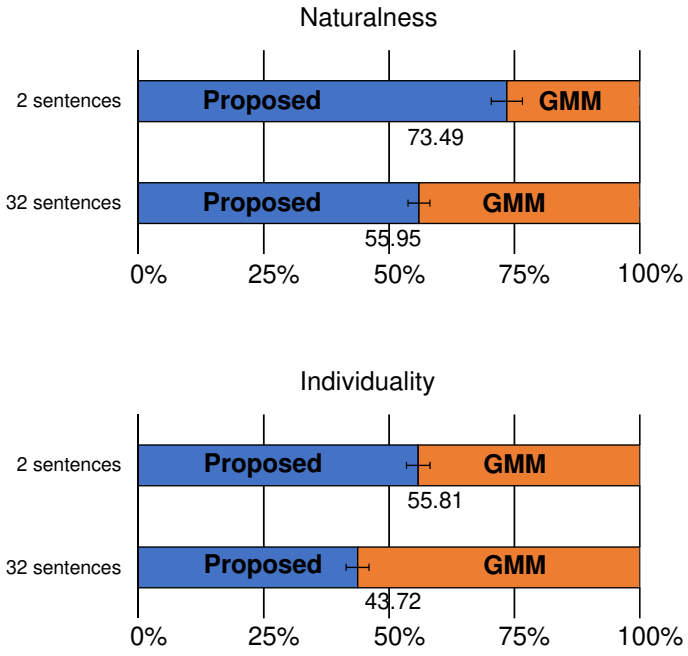

図 5.3: 提案手法と GMM に関する主観評価結果

みを付加するなどが挙げられる．

5.3.3　主観評価実験結果

MCD による客観評価に加えて，聴取実験による比較実験を行った．ここでは変換音声の主観評価尺度として，変換音声が自然発話にどの程度近いかの尺度である自然性 (naturalness) と，目標とする話者の声に近づいているかの尺度である個人性 (individuality) の 2 つを用いた．主観評価実験は 10 名の日本語を母語とする聴取者によって行った．自然性においては，一対比較法を用いた評価を行った．一対比較法では，2 つの異なる変換手法から得られた音声サンプルを提示し，どちらの音声サンプルがより自然な音声だと感じたかを 5 段階の指標 (A が自然である，どちらかと言えば A が自然である，A と B は同程度である，どちらかといえば B が自然である，B が自然である) によって評価した．個人性においては Reference AB test (RAB 法) による評価を行った．RAB 法では，変換先の話者の自然音声と共に，2 つの変換手法から得られた音声サンプルを提示し，どちらの音声がより自然音声の個人性に近いかを 5 段階の指標 (A の方が似ている，どちらかと言えば A の方が似ている，A と B は同程度である，どちらかといえば B の方が似ている，B の方が似ている) によって評価した．入力話者である男性話者 1 名から，男性 2 話者と女性 2 話者の計 4 話者を変換先の話者とし，各ペアに付き 5 文ずつ評価を行うことで，各手法ごとに計 20 文の比較を行った．比較する手法は提案手法 (unsupervised) と GMM，EVGMM，AVM(s+t) の 3 手法とし，2 文または 32 文で適応を行った計 6 通りの比較を行った．被験者の負担を軽減するため，ここでは提案手法以外の手法間の比較は行わなかった．

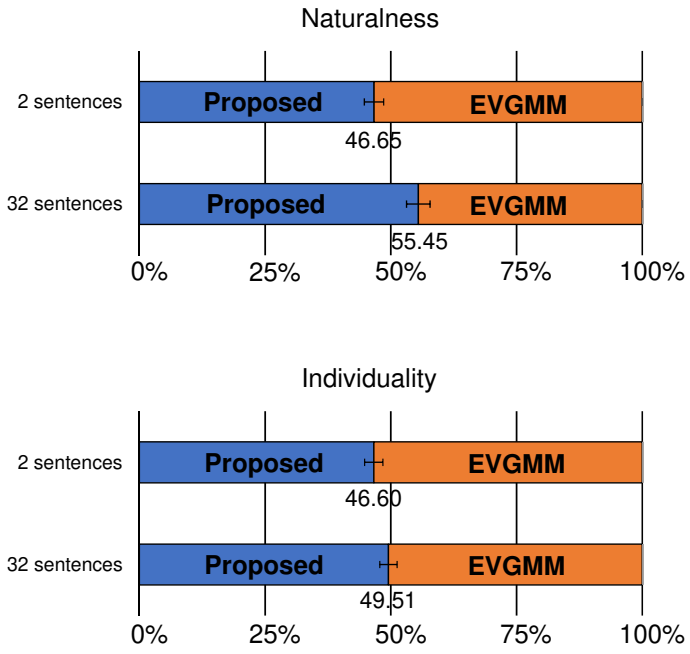

図 5.4: 提案手法と EVGMM に関する主観評価結果

図 5.3 に提案手法と GMM の主観評価の結果を示す．結果から，適応文が 2 文の場合には自然性と個人性の両評価尺度において提案手法が上回っていることが分かる．対して 32 文で適応を行った場合には，自然性は依然として提案手法が上回っているものの，個人性では GMM が提案手法を上回っている．

この結果は，32 文で適応を行った場合における自然性以外では，客観評価の結果と対応が取れていると言える．32 文で適応を行った場合でも提案手法の方が自然性が高い理由としては，予め大量の話者を用いて学習を行って得られた事前知識によって，話者に依存しない音声を高精度にモデル化できていることによる結果だと考えられる．個人性が下回っている点に関しては，客観評価の項で述べたように，適応するパラメータの少なさが原因であると考えられる．

図 5.4 に提案手法と EVGMM の主観評価の結果を示す．結果から，適応文が 2 文の場合には自然性と個人性の両評価尺度において提案手法が EVGMM を僅かに下回っていることが分かる．対して 32 文で適応を行った場合には，自然性は提案手法が EVGMM を上回り，個人性はほぼ同程度という結果が得られた．

2 文での適応において客観評価では提案手法が EVGMM を僅かに上回っているのに対して，主観評価では逆に EVGMM が提案手法を僅かに上回っている．提案手法と EVGMM では，大枠として学習しようとしている話者空間が同一であることを考えると，EVGMM では適応データとして与えられた特徴量に関して直接話者空間の基底への射影重みを推定しており，提案手法では適応データを話者空間基底に一度分解しその上で重みを推定しているという差から生じていると考えられる．すなわち，少量の適応データしか与えられていない場合には，その少量のデータに対して話者空間の基底への分離の精度が低かったと

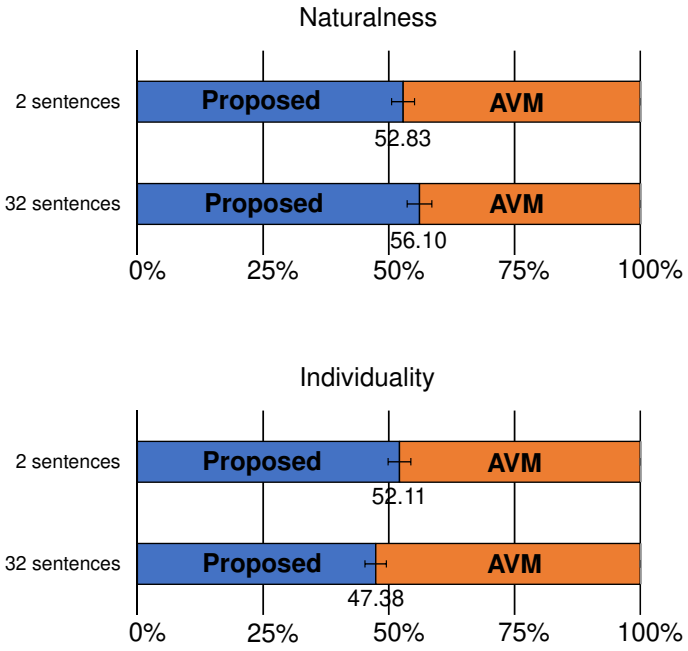

図 5.5: 提案手法と AVM に関する主観評価結果

き，推定される重みのミスマッチも大きくなってしまうことが考えられる．対して，多くの適応データが得られた場合には，DNN による高精度な話者空間への分解が有効に働き，客観評価での結果と同様に提案手法が EVGMM を上回っていると考えられる．

図 5.5 に提案手法と AVM の主観評価の結果を示す．結果から，適応文が 2 文の場合には自然性と個人性の両評価尺度において提案手法が AVM を僅かに上回っていることが分かる．対して 32 文で適応を行った場合には，自然性は提案手法が AVM を上回り，個人性は提案手法が AVM を僅かに下回っているという結果が得られた．

AVM の客観評価の結果と主観評価の結果を比較すると，客観評価では提案手法を大きく下回っているのに対して，主観評価では大きな差は無く，32 文で適応を行った個人性に関しては提案手法を僅かに上回っている．特に，2 文での適応の場合には，客観評価の結果は GMM とほぼ同程度であるのに対して，主観評価では GMM と比べて提案手法に近い精度となっていることが分かる．大量の話者で学習したことによる事前知識を用いているという点を考えれば，この主観評価の結果は自然であるとも言える．

全体的には，提案手法と GMM を除く 2 手法においては個人性の評価に大きな差は見られないという傾向が見られた．加えて自然性においては，2 文で適応を行った場合の提案手法と AVM・EVGMM の間に大きな差は見られなかったものの，32 文で適応を行った提案手法は教師有り学習を行った GMM を含む 3 手法を全て上回っていると言う結果が得られた．この提案手法の自然性評価の高さは，提案手法における平均話者成分を抽出する DNN が有効に働いた結果であると考えられる．

第6章

提案手法：EVGMMに基づく話者空間基底を用いたパラレルデータフリーDNN声質変換

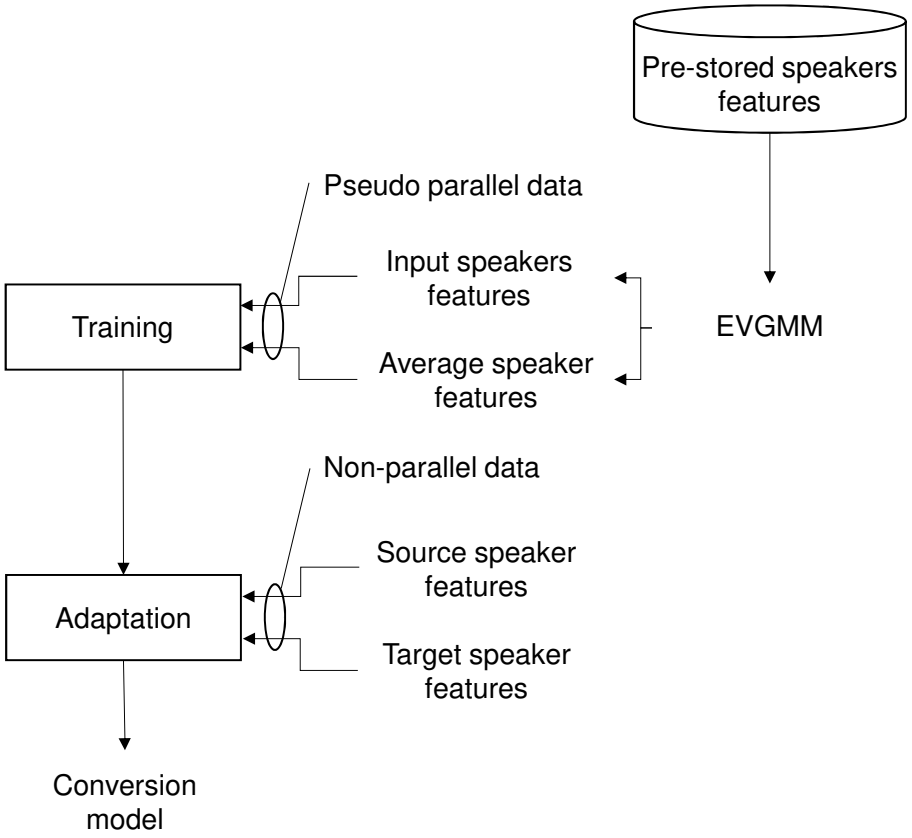

図 6.1: 提案手法における全学習過程パラレルデータフリー拡張の概要

6.1　提案手法のパラレルデータフリー拡張

5 章で提案した EVGMM と DNN を組み合わせた多対多声質変換手法では，EVGMM の構築を行う際に，アンカーとなる話者と全事前収録話者の間でパラレルデータが必要となるという問題が存在した．3.2 節示したように，全過程において発話内容の制約が無い学習が実現すれば，より大規模な学習データの利用が可能となり，データ効率の上でも実用面でも非常に有用であると言える．しかし，3.2 節で示した 2 つの手法では，INCA はデータ中の音素に明示的ではないものの制約が存在し，適応型 RBM では入力と出力両方に適応が必要という課題がそれぞれ存在する．そこで，これまでの提案手法における EVGMM の学習部分を改善することで，提案手法を全学習過程においてパラレルデータフリーな多対多声質変換への拡張を行うことで，学習データに音素的制約が存在せず，出力話者の適応のみからなる多対多声質変換を実現する．図 6.1 に手法の概要を示す．基本的な枠組みはこれまでの提案手法と同様であり，EVGMM のパラメータを用いて擬似的なパラレルデータを構築し，入力特徴量を話者空間基底と平均話者成分に分解した上で再構成を行うというものである．

ここで今一度提案手法における EVGMM を用いた擬似パラレルデータの計算について考える．5 章で示したように，提案手法では一対多 EVGMM の変換式 (5.1) に基づいて基

底成分及び平均話者への変換を行う．

$$F(\boldsymbol{x}_t)=\sum_{m=1}^{M}\gamma_{m,t}(\mathbf{B}_m\boldsymbol{w}^{(Y)}+\boldsymbol{b}_m^{(0)}+\mathbf{A}_m(\boldsymbol{x}_t-\boldsymbol{\mu}_m^{(X)}))$$

ここで，これまでの実験ではハイパーパラメータ $\boldsymbol{A}$ として一対多 EVGMM の話者非依存分散と一対多共分散を使用していた．

全学習過程においてパラレルデータフリーを実現するため，変換式 (5.1) で用いる EVGMM 学習の際に，一対多 EVGMM によってアンカー話者と全事前収録話者の特徴量に対する同時確率密度をモデル化するのではなく，全事前収録話者のみによる単一話者空間の特徴量に対する確率密度をモデル化することで，モデルの全学習過程においてパラレルデータを使用しない声質変換を実現する．この単一話者空間 EVGMM の式は以下の様に表される．

$$P(\boldsymbol{y}_t^{(s)}\mid\lambda^{(EV)},\boldsymbol{w}^{(s)})=\sum_{m=1}^{M}\alpha_m\mathcal{N}(\boldsymbol{y}_t^{(s)};\boldsymbol{\mu}_m^{(Y)}(\boldsymbol{w}^{(s)}),\boldsymbol{\Sigma}_m^{(YY)}) \tag{6.1}$$
$$\boldsymbol{\mu}_m^{(Y)}(\boldsymbol{w}^{(s)})=\mathbf{B}_m\boldsymbol{w}^{(s)}+\boldsymbol{b}_m^{(0)}$$

この単一話者空間における EVGMM を用いた 5 章と同様の枠組みによって，話者空間基底への分解と平均話者への分解を行う DNN を学習する．この際，変換式 (5.1) で必要なパラメータを考えると，共分散 σ^{YX} 以外のパラメータは単一話者空間 EVGMM においても学習過程で得ることができる．そのため，共分散 σ^{YX} をなんらかのハイパーパラメータとして適当な値を用いるか，何らかの方法で推定することができれば，提案手法を全学習過程でパラレルデータフリーな多対多声質変換に拡張することが可能となる．

まとめると，これまで提案してきた変換対象の話者に関してパラレルデータフリーな多対多声質変換の枠組みを拡張し，擬似パラレルデータを構築する EVGMM を単一話者空間で学習を行うことで全学習過程でパラレルコーパスを必要としない完全なパラレルデータフリーの声質変換システムを提案する．本手法の利点としては，話者基底成分・平均成分への分解を行う DNN の学習と，出力話者固有の重みの学習を，パラレルデータフリーに行うことが可能であり，任意の発話内容の音声を学習に使用できるという点が挙げられる．

6.2　パラレルデータフリー多対多声質変換に関する予備実験

6.2.1　実験条件

提案手法によるパラレルデータフリー声質変換の実験的検討として，共分散として使用するハイパーパラメータを単位行列とした場合の評価実験を行った．実験としては，入力話者 1 名と出力話者 20 名の両者を学習データ外の未知話者とした場合の声質変換の精度比較を行った．この実験では，参照話者ありの EVGMM を用いて学習した場合と参照話者なしの単一話者空間で学習した場合の提案手法の変換結果を比較することで，EVGMM の構築時に参照話者を用いることの影響を確認することを目的としている．式 (5.2) の $\mathbf{A}_m$ について、参照話者なしの場合は単位行列、参照話者ありの場合は $\boldsymbol{\Sigma}_m^{(YX)}\boldsymbol{\Sigma}_m^{(XX)-1}$ とした。こ

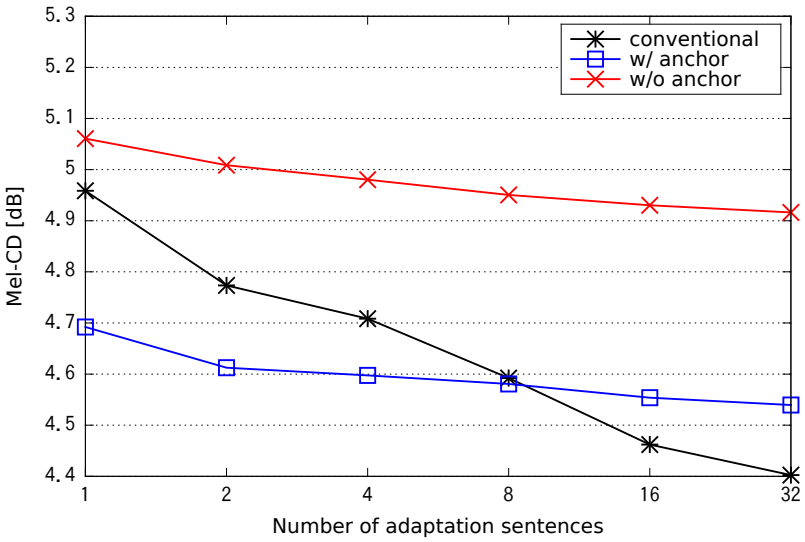

図 6.2: 20 話者への変換に関するメルケプストラム歪みによる客観評価結果

れは事前収録話者を入力とした変換の際に，参照話者からの分散共分散行列によって変換を行ったことを意味する．それぞれの条件で用いる EVGMM の混合数は 256 とし，分散行列と相互共分散は対角行列とした．同様に，各条件で用いる DNN は隠れ層の数を 5，隠れ層のノード数を 512 とし，学習回数は 50 epoch とした．

データセットとして，ATR 日本語音声データベースの B セット [26] を使用した．出力対象の話者には JNAS から男性話者 10 名と女性話者 10 名の合計 20 名を使用し，各話者について，適応に使用する文の数を 1 文から 32 文まで変化させ，変換精度の平均を比較した [31]．EVGMM の学習コーパスとしては，JNAS から 96 話者の音素バランス文 50 文を使用した．変換を行う際の未知入力話者としては，学習に使用していない男性話者一名を使用した．特徴量としては，STRAIGHT 分析 [27] によって得られたメルケプストラム 24 次元とそのデルタ特徴量を使用し，変換精度の比較には以下の式で表されるメルケプストラム歪み (Mel cepstral distortion:Mel-CD) を用いた．

$$\mathrm{MelCD[dB]} = \frac{10}{\ln 10}\sqrt{2\sum_{d=1}^{24}(mc_d - \bar{mc}_d)^2} \tag{6.2}$$

ここで，mc_d と $\bar{mc}_d$ は出力話者の正解データの特徴量ベクトルと，入力話者の特徴量ベクトルから変換された特徴量ベクトルをそれぞれ表す．

6.2.2　結果

図 6.2 に適応文数に対する各条件のメルケプストラム歪みを示す．参照話者なしの場合を “w/o anchor”，参照話者ありの場合を “w/ anchor” とした．また，参考として入力話者 1 名と出力話者 20 名の全 20 組に対して一対一の教師あり学習を行った GMM の変換結果

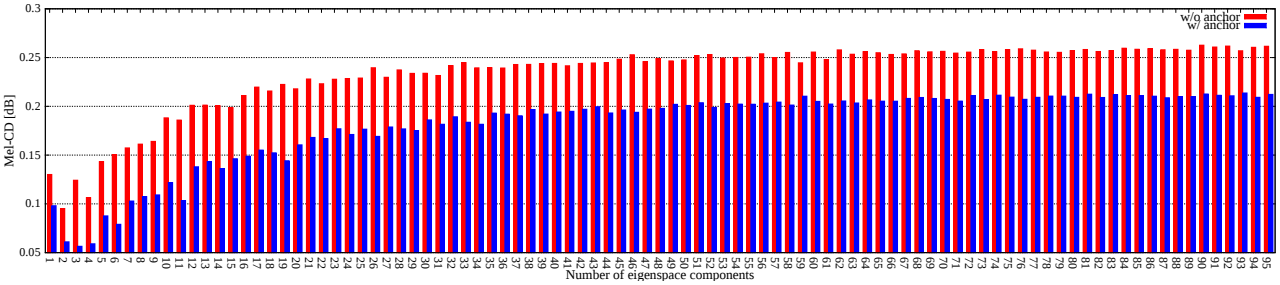

図 6.3: 開発データにおける各基底成分の変換誤差

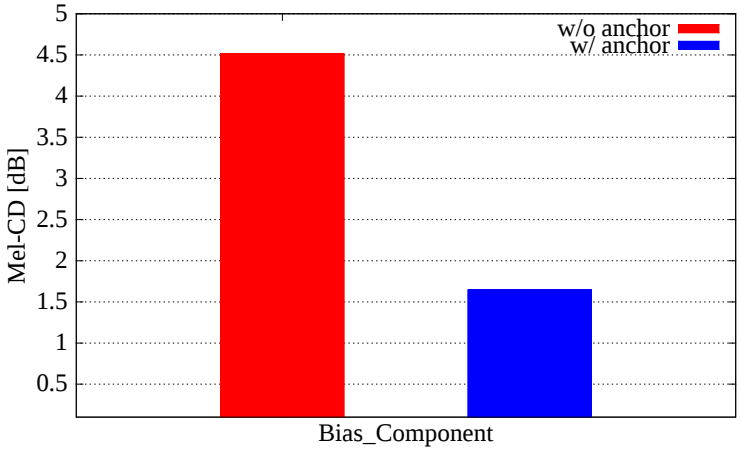

図 6.4: 開発データにおけるバイアス成分の変換誤差

を “conventional” として図に示した．この際，混合数は 2 から 256 まで変化させ最も精度の高かった結果を図示している．結果から，参照話者を用いない事で，変換精度が大きく劣化していることが分かる．一方適応文数の変化に対しては，どちらの条件でも同様の傾向が観察された．

6.2.3　考察

変換結果の比較から，参照話者の有無が最終的な変換精度に大きな影響を与えることが分かった．そこで，それぞれの条件における学習中の各パラメータおよび開発データに対する結果を比較し，参照話者の有無が実際にどのような部分において影響を与えているかについて分析を行った．

初めに，基底成分への分解を行う DNN の学習時の変換精度の比較を行った．DNN の学習の際には学習データのうち 10%を分割し，学習の評価を行う開発データとして使用している．この開発データに対する，各基底及びバイアス成分 (95 個の基底+バイアス) の学習終了時のメルケプストラム歪みの確認を行った．結果を図 6.3 および図 6.4 にそれぞれ示す．結果から，バイアスを含む全ての成分において参照話者を用いない場合に，変換精度が劣

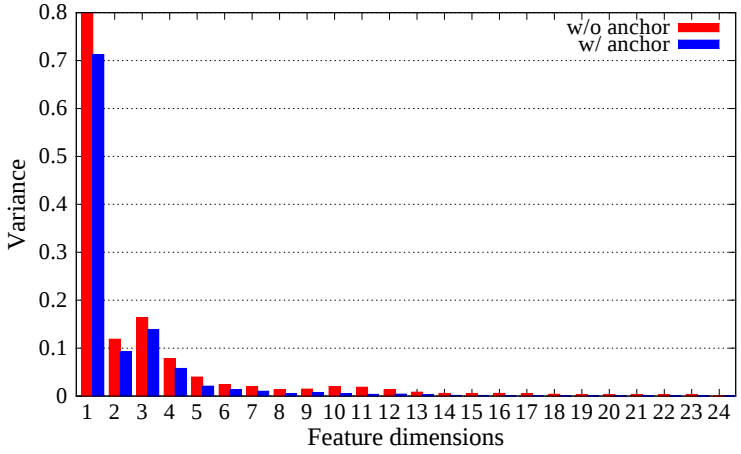

図 6.5: 参照話者の有無による話者非依存 GMM の平均ベクトルの分散の比較

化することが分かる．図 6.3 より，式 (5.2) の TD($\lambda^{(EV)}$) で定義される特徴量による話者空間表現がうまく実現されていないと示唆される．一方で，参照話者を用いない場合，参照話者と事前収録話者間での動的なアライメントを行わないため，特徴量空間そのもののモデル化はより精緻になることが期待される．よって、式 (5.2) より，$\mathbf{B}_m$ の構成に関しては参照話者の有無の影響は小さいと考えられ，むしろ各混合への事後確率を表す $\gamma_{m,t}$ への影響が示唆される．ここで，参照話者の有無の $\gamma_{m,t}$ への影響を確認するため，EVGMM の学習の過程で構築した話者非依存 GMM の全混合の平均ベクトルに関して分散を計算した．結果を図 6.5 に示す．図 6.5 から，全ての特徴量次元において，参照話者が存在しない場合に分散の値が大きくなることが分かる．このことから，参照話者が存在しない場合，話者非依存 GMM の各混合要素が表す情報が，より「拡大」していると考えられる．参照話者を使用した結合確率 EVGMM，例として一対多 EVGMM では，入力側の特徴量空間が常に同一の話者で学習されるために，GMM の各混合が話者情報以外の情報に対応するように学習されると期待される．パラレルデータフリーの学習ではそのような制約が発生しないために，各混合の表す情報に話者の違いも表出してしまうと考えられ，結果的にモデルそのものの推定精度が低下していると考えられる．図 6.5 の結果から開発データに対するバイアス成分に関しても，参照話者ありの場合に 1.646[dB]，参照話者なしの場合に 4.518[dB] となり，メルケプストラム歪みの差が非常に大きくなっていることが分かる．この原因としてバイアス項においては，ハイパーパラメータ $\mathbf{A}_m$ を単位行列としたことによって，変換式の残差項の値が大きくなりすぎてしまっていることが精度の低下の大きな要因になっていると考えられる．TD($\lambda^{(EV)}$) での議論と同様に事後確率の推定精度も低下しているといえるため，他の基底成分以上に精度の低下が激しくなっていることも考えられる．

これらの結果から，単位行列によるハイパーパラメータ $\mathbf{A}_m$ の置き換えでは変換精度は著しく低くなってしまうため，より適した値を推定する必要があると考えられる．

6.3　欠損EMアルゴリズムを用いたパラレルデータフリー共分散推定

前節での実験結果から，より適したハイパーパラメータ $\mathbf{A}_m$ の推定を目的として，欠損EMアルゴリズムを用いた共分散推定の導入を行う．

大谷らによって提案された分布共有モデルの応用手法として，欠損値を含む特徴量を用いた結合GMMを学習するEMアルゴリズムが内田らによって提案されている [39, 40]．分布共有モデルは2つの特徴量空間を介した変換を行う際に，経由する部分空間が共通のモデルによって表現されていると仮定することで，推定するパラメータを少なくすることができるというものである．内田らはこの分布共有モデルの学習において，パラレルデータの存在しない (欠損値の存在する) 特徴量を2乗誤差基準による特徴量変換によって補いながら学習を行うEMアルゴリズムを提案している．ここではこの欠損EMアルゴリズムを用いてパラレルデータ無しに共分散行列の推定を行うことを試みる．

単一特徴量空間で学習したEVGMMが式 (6.1) で表される場合，平均話者をアンカー話者とした多対一EVGMMは以下のように表される．

$$
\begin{aligned}
&P(\boldsymbol{y}_t^{(s)}, \mathbf{M}_t \mid \lambda^{(EV)}) \\
&= \sum_{m=1}^{M} \alpha_m \mathcal{N}([\boldsymbol{y}_t^{(s)\top}, \boldsymbol{m}_t^{\top}]^{\top}; \boldsymbol{\mu}_m^{(Z)}(\boldsymbol{w}^{(s)}), \boldsymbol{\Sigma}_m^{(Z)})
\end{aligned}
$$

$$
\begin{aligned}
\boldsymbol{\mu}_m^{(Z)}(\boldsymbol{w}^{(s)}) &= \begin{bmatrix} \mathbf{B}_m \mathbf{w}^{(s)} + \boldsymbol{b}_m^{(0)} \\ \boldsymbol{b}_m^{(0)} \end{bmatrix} \\
\boldsymbol{\Sigma}_m^{(Z)} &= \begin{bmatrix} \boldsymbol{\Sigma}_m^{(YY)} & \boldsymbol{\Sigma}_m^{(YM)} \\ \boldsymbol{\Sigma}_m^{(MY)} & \boldsymbol{\Sigma}_m^{(MM)} \end{bmatrix}
\end{aligned}
$$

$\boldsymbol{y}_t^{(s)}$ を事前収録話者 s の特徴量，$\boldsymbol{m_t}$ は平均話者の特徴量をそれぞれ表す．このとき，単一特徴量空間のEVGMMが学習済みであれば，共分散 $\boldsymbol{\Sigma}_m^{(YM)}$ 以外のパラメータはその値を用いることができる．

EMアルゴリズムによってパラメータの推定を行うことを考えると，結合特徴量系列 $[\boldsymbol{Y}^{(s)\top}, \boldsymbol{M}^{\top}]^{\top}$ が必要となるが，平均話者は事前収録話者から推定されたものであり，実際には特徴量 $\boldsymbol{m}_t$ は存在しない (欠損値)．欠損EMアルゴリズムでは，この存在しない特徴量を推定値によって補うことでパラメータの学習を行う．

推定結合特徴量 $\boldsymbol{z}_t^{(s)} = [\boldsymbol{y}_t^{(s)\top}, \boldsymbol{m}_t'^{\top}]^{\top}$ について，EMアルゴリズムによって以下の様に学習を行いパラメータを推定する．

Eステップ

$$
\gamma_{m,t}^{(s)} = \frac{\mathcal{N}(\boldsymbol{z}_t^{(s)}; \boldsymbol{\mu}_m^{(Z)}(\boldsymbol{w}^{(s)}), \boldsymbol{\Sigma}_m^{(Z)})}{\sum_{j=1}^{M} \alpha_j \mathcal{N}(\boldsymbol{z}_t^{(s)}; \boldsymbol{\mu}_j^{(Z)}(\boldsymbol{w}^{(s)}), \boldsymbol{\Sigma}_j^{(Z)})}
$$

Mステップ

$$\mathbf{\Sigma}_m^{(Z)} = \frac{1}{\gamma_m}\sum_{s=1}^{S}\sum_{t=1}^{T(s)}\left(\gamma_{m,t}^{(s)}(\boldsymbol{z}_t^{(s)} - \boldsymbol{\mu}_j^{(Z)})(\boldsymbol{z}_t^{(s)} - \boldsymbol{\mu}_j^{(Z)})^\top + \mathbf{D}_m\right)$$

ここで$T(s)$は各事前収録話者の特徴量フレーム数を，γ_mは全事前収録話者の全時系列に関する$\gamma_{m,t}^{(s)}$の和をそれぞれ表す．

Eステップにおける平均話者の推定特徴量$\boldsymbol{m}_t^{'}$はGMMの部分空間から以下の様に求める．

$$\boldsymbol{m}_t^{'} = \sum_{m=1}^{M} P(\boldsymbol{z}_t^{(s)} \mid \lambda^{(EV)}) E_{m,t}^{(\mathbf{M}|\mathbf{Y})}$$

$E_{m,t}^{(\mathbf{M}|\mathbf{Y})}$は事前収録話者の特徴量$\boldsymbol{y}_t^{(s)}$が与えられたときの平均話者の特徴量の期待値であり，以下の式で表される．

$$E_{m,t}^{(\mathbf{M}|\mathbf{Y})} = \boldsymbol{b}_m^{(0)} + \mathbf{\Sigma}_m^{(MY)}\mathbf{\Sigma}_m^{(YY)-1}(\boldsymbol{y}_t^{(s)} - \boldsymbol{\mu}_m^{(s)})$$

また，Mステップにおける$\mathbf{D}_m$も同様に以下の式で表される行列となる．

$$\mathbf{D}_m = \begin{bmatrix} \mathbf{0}^{(d,d)} & \mathbf{0}^{(d,d)} \\ \mathbf{0}^{(d,d)} & \mathbf{D}_m^{(\mathbf{M}|\mathbf{Y})} \end{bmatrix}$$

$$\mathbf{D}_m^{(\mathbf{M}|\mathbf{Y})} = \mathbf{\Sigma}_m^{(MM)} - \mathbf{\Sigma}_m^{(MY)}\mathbf{\Sigma}_m^{(YY)-1}\mathbf{\Sigma}_m^{(YM)}$$

dは特徴量の次元である．これは，現在のステップにおけるGMMのパラメータによって事前収録話者から平均話者への変換を行っていることに相当する．また，Mステップにおいては分散共分散行列の更新のみを行う．これらの平均話者特徴量の更新，Eステップ，Mステップを収束するまで繰り返すことによってGMMの学習を行う．

6.4　パラレルデータフリー共分散を用いた提案手法に関する実験

ハイパーパラメータ$\mathbf{A}_m$の改善を行った提案手法によるパラレルデータフリー声質変換の評価として，従来のGMMによる教師あり声質変換手法との客観及び主観評価指標による比較実験を行った．提案手法においては事前収録話者96名の非パラレル学習データを用いてEVGMMおよびDNNの学習を行った．また，提案手法では入力話者に関しては適応データを一切使用せず，目的話者に関してのみ適応を行った．提案手法で用いるEVGMMの混合数は256，DNNは隠れ層の数を5，隠れ層のノード数を512とした．従来手法(GMM)に関しては入力話者と目的話者のパラレルデータを用いて学習を行った．従来手法の混合数は2混合から128混合まで変化させ，最も結果がよかったものを採用した．テストデータには入力話者として男女各一名の計2名，目的話者として男女各2名の計4名を使用し，データ量は各話者21文とした．客観評価における変換精度の比較にはこれまでの実験と同様にメルケプストラム歪みを用いた．

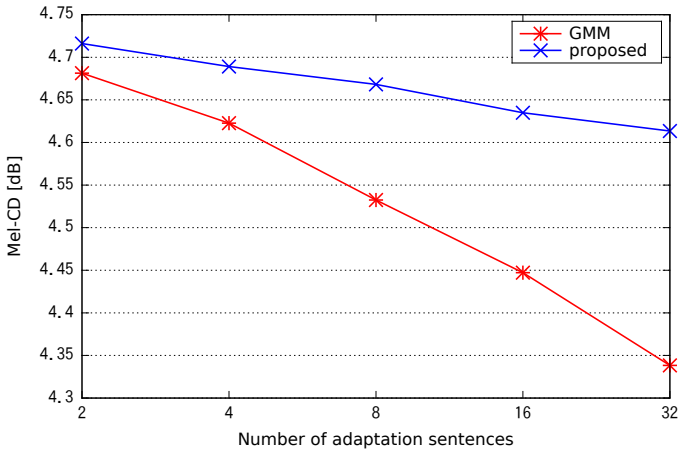

図 6.6: 4 話者への変換におけるメルケプストラム歪みによる客観評価結果

6.4.1　客観評価実験と結果

2 名の入力話者から 4 名の目的話者への変換について客観評価指標による評価を行った．提案手法における適応データおよび従来手法におけるパラレルデータを 2 文から 32 文に変化させた際の結果を図 6.6 に示す．図から，適応データが少ない場合には提案手法と従来手法が近い精度となっているものの，全体的な精度は教師有り学習を行った GMM を大きく下回っており，学習データが少ない場合も僅かに下回っていることが分かる．

6.4.2　主観評価実験と結果

提案手法と教師有り学習を行った GMM に関して，変換音声の自然性と個人性についてそれぞれ 5 名の日本人学生を対象に聴取実験を行った．実験条件は 5 章における客観評価実験と同様に自然性を一対比較法，個人性を RAB 法によってそれぞれ 5 段階で評価した．実験結果を図 6.7 に示す．青が提案手法をより自然，または話者性を再現していると判断した被験者の割合を，オレンジ色が GMM を選択した被験者の割合をそれぞれ表す．図中のエラーバーは 95%信頼区間を表す．実験では学習・適応データに 2 文を使用した場合と 32 文を使用した場合の比較をそれぞれ行った．結果から，2 文で学習を行った場合には，提案手法が自然性と話者性の両方において GMM を僅かに上回っている．また，32 文で学習を行った場合には，GMM が自然性と話者性の両方において提案手法を上回っている．しかし，32 文を学習に用いた場合の自然性は，GMM が提案手法を大きく上回っている一方で，話者性においては 32 文であっても大きな差は無く，ほぼ同等であると言える．この結果から，学習データ量が多い場合の自然性を除いた条件において，パラレルデータフリー学習を行った提案手法が教師あり学習を行った GMM と同等の変換を実現していると言える．

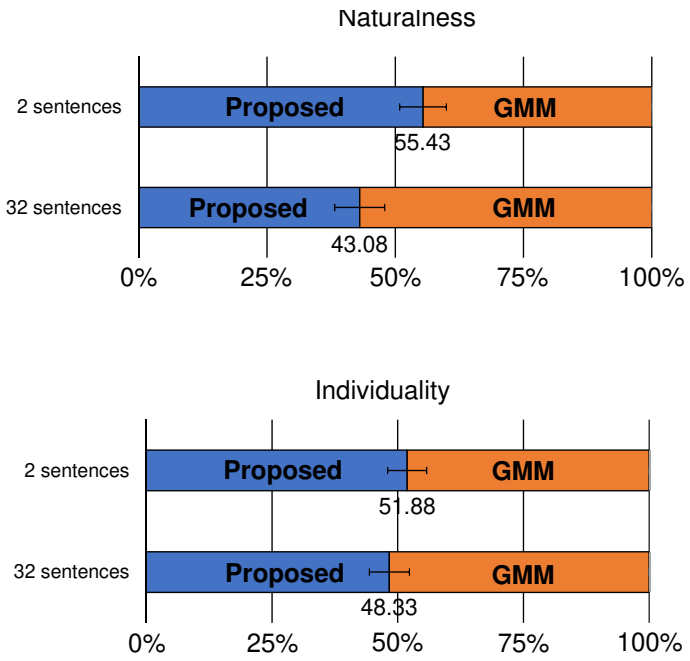

図 6.7: 提案手法と GMM の主観評価結果の比較

第7章

提案手法：FHLを用いた声質変換

7.1　DNN 声質変換手法におけるパラメータ適応

5 章において比較に用いた Average Voice Model と話者表現を用いた声質変換 [29] では，入力層に直接入出力話者の情報を与えることで変換が適応されることを期待したものであった．しかし，より高度な適応を考えた場合，DNN の各層のパラメータに対して適応を行うべきであると考えられる．また，各層における入出力話者による適応を可視化することができれば，各層がどちらの話者の情報をより必要としているかを分析することが可能となる．これによってマルチ出力サブネットワークを用いた手法において用いた，「DNN の浅い層では話者性の除去，深い層では目的話者の話者性の付加が行われている」という仮定の妥当性を高めることも可能となると考えられる．そこで本章では，DNN の各層に対して入出力話者による適応を行う手法として，音声認識などで用いられている手法である Factorized Hidden Layers (FHL) を用いた声質変換を提案する．

7.2　提案手法：FHL を用いた多対多声質変換

ここでは前述した目的から，FHL を用いた多対多声質変換手法を提案する．基本的な枠組みは 2 章で示した通常の FHL そのままであり，隠れ層に与える話者表現ベクトルとして入力話者の話者表現ベクトル $\boldsymbol{d}_i, \boldsymbol{v}_i$ と，出力話者の話者表現ベクトル $\boldsymbol{d}_o, \boldsymbol{v}_o$ の 2 種類を用いることで，話者非依存 DNN を入力話者から出力話者へ変換を行う DNN に適応する．ここでは，適応に用いる話者表現ベクトルとしては EVGMM より得られる固有声重みを使用した．

具体的な手法の概要は以下の様になる．

1. 大量の話者からなるパラレルコーパスを用いて 1 つの DNN の学習を行い，話者非依存 DNN を構築
2. 全学習データから EVGMM を構築し，各話者に対応する話者表現ベクトルを抽出
3. 話者非依存 DNN のパラメータを固定した状態で，再び全学習データによって入力話者及び出力話者の話者表現ベクトルを隠れ層に与えながら学習を行い，$\boldsymbol{\Gamma}^l$，$\boldsymbol{\Psi}^l$，$\mathbf{U}^l$ を推定
4. テストデータの入出力話者に関して，適応データから話者表現ベクトルを推定し，それらを与えた DNN によって最終的な変換を実行

2 章で述べた話者表現ベクトルの再学習と同様に，話者非依存 DNN に関しても，$\boldsymbol{\Gamma}^l$，$\boldsymbol{\Psi}^l$，$\mathbf{U}^l$ の推定が終わった後に，話者依存なパラメータと FHL のパラメータを全て固定した状態で再学習を行うことが可能である．しかし，音声認識における先行研究では最終的な精度に大きな変化が見られなかったという結果が得られていたため，本稿ではこれを行っていない [41]．

7.3　実験

提案手法の性能の確認のため，学習データ外の未知入力話者から未知出力話者への変換に関して，既存手法との客観評価指標による比較実験を行った．比較手法としては，提案手法と同様に話者表現ベクトルを用いている変換手法として AVM，及び 5 章で提案した EVGMM に基づく話者空間基底を用いた声質変換手法 (以下 EVDNN) を用いた．

ここでは 5 章における一対多声質変換実験と同様の条件を用いている．提案手法，AVM，EVDNN，及び話者表現ベクトル推定用の EVGMM の学習コーパスとしては，JNAS から 96 話者の音素バランス文 50 文を使用した [26, 31]．FHL の初期モデルとなる話者非依存 DNN の学習では，JNAS 中の 96 話者による同一話者を除いた $96 * (96 - 1)$ 通りの話者ペアのうち，同一のサブセットを持つ 1010 ペアのパラレルデータを使用した，テストデータの入力話者としては，学習に使用していない男性話者一名を使用し，出力の未知話者としては JNAS から男性話者 5 名と女性話者 5 名の合計 10 名を使用した．また，サブセット J を適応及び評価データとし，1 番から 32 番の 32 文を適応データ，33 番から 53 番の 21 文をテストデータとして使用した．特徴量としては，STRAIGHT 分析 [27] によって得られたメルケプストラム 24 次元とそのデルタ特徴量を使用した．比較の際には出力話者に関する適応データ，すなわち話者表現ベクトルの推定に用いるデータ量を 1 文から 32 文に変化させ，入力話者の話者表現ベクトルの推定に用いる適応データは 32 文で固定とした．提案手法で用いる DNN は EVDNN と同様に，隠れ層の構成を 5 層 256 ノードとした．実験結果の図中に示す提案手法のアルファベット列 (ex. “SNNNT”) は DNN の構造を表しており，S は入力話者の話者表現ベクトルによって適応した FHL，T は出力話者の話者表現ベクトルによって適応した FHL，N は適応を行っていない話者非依存 DNN の層である．ここでは，考えられる FHL の組み合わせ (3^5 通り) のうち，入力に近い層に入力話者の FHL が，出力に近い層に出力話者の FHL が配置されており，出力層を出力話者の FHL で適応しているものを簡易的に選択した．加えて，全ての層を入出力話者の結合話者表現ベクトルによって適応したものを “BBBBB” とした．また，適応は DNN 中の重み行列に対してのみ行い，バイアス項に対しては行っていない．変換精度の比較には以下の式で表されるメルケプストラム歪み (Mel cepstral distortion:MCD) をとり，その平均によって評価を行った．

$$\mathrm{MCD[dB]} = \frac{10}{\ln 10}\sqrt{2\sum_{d=1}^{24}(mc_d - \bar{mc}_d)^2} \tag{7.1}$$

ここで，mc_d と $\bar{mc}_d$ はターゲット話者の正解データの特徴量ベクトルと，ソース話者の特徴量ベクトルから変換された特徴量ベクトルをそれぞれ表す．EVGMM の混合数は 256 とし，分散と共分散は対角成分以外 0 として学習を行った．EVGMM より得られる固有声重みは話者数 −1 である 95 次元とした．提案手法及び既存手法で用いる DNN は隠れ層の数を 5 とし，隠れ層のノード数を 512 とした．活性化関数は Rectified Linear Unit を使用した [19]．

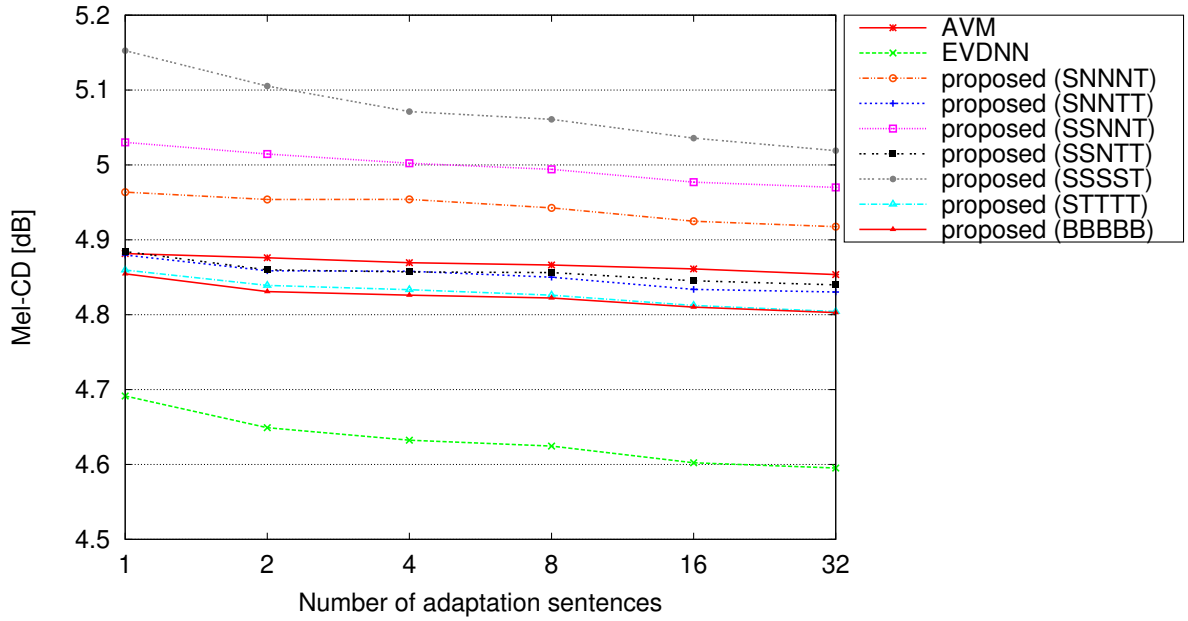

図 7.1: FHL の入出力話者適応による変換精度の比較

7.3.1　比較実験

提案手法，AVM の実験結果を図 7.1 に示す．図 7.1 はそれぞれの手法に関して，出力話者の話者表現ベクトルの推定に用いる適応データを 1 文から 32 文に変化させた際の変換精度を表す．

初めに，図 7.1 中の提案手法の結果を比較すると，“SSSST”，“SSNNT”，“SNNNT” といった出力話者によって適応を行った層が少ない場合の結果が悪く，“SSNTT”，“SNNTT”，“STTTT”，“BBBBB” といった出力話者による適応層が多い場合の結果がそれらを上回っている．このことから，DNN による声質変換では出力話者による適応が重要であり，より多くの層を出力話者に適応することが変換精度の改善に繋がると考えられる．

AVM と提案手法を比較した場合も，出力話者による適応層が多い “SSNTT”，“SNNTT”，“STTTT”，“BBBBB” といった適切な適応を行った提案手法が，AVM，すなわち入力に直接話者表現ベクトルを与えた場合よりも変換精度が良くなっており，提案手法による各層の適応が有効に働いていると考えられる．一方で，5 章における EVGMM と DNN を用いた手法 (ELDNN) と比較した場合，FHL による変換精度はこれを大きく下回っている．このことから，話者基底への分解を行うことの有効性がより示された形となった．

7.3.2　入出力話者による各層の適応の可視化

7.3.1 の実験結果から，全層に対して入出力話者両方の情報を与えた場合である “BBBBB” の変換結果が最も良いものの，条件 “STTTT” と非常に近い結果となっている．このことから “BBBBB” では，各層で入力話者または出力話者のどちらの適応をより重視するかに

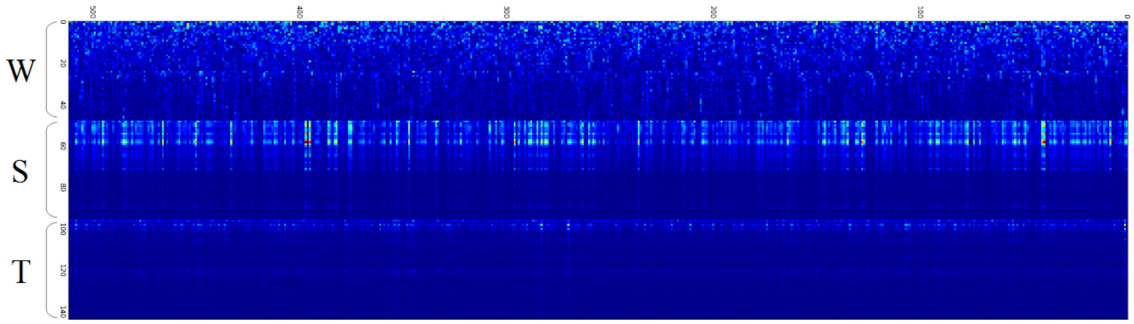

図 7.2: DNN の 1 層における話者非依存パラメータ (W) 及び FHL による入出力話者適応パラメータ (S, T)

偏りが生じていることが推測される．そこで，条件 “BBBBB” において FHL による適応パラメータを，入力話者によるものと出力話者によるものに分離することで，各層における適応がどちらの話者に比重を置いているかの確認を行った．l 層における話者非依存 DNN のパラメータ $\mathbf{W}^l$ に対する適応パラメータ $\mathbf{\Gamma}^l\mathbf{D}_s^l\mathbf{\Psi}^l$ を考えたとき，$\mathbf{D}_s^l$ は入力話者と出力話者の結合話者表現ベクトルを対角成分にもつ対角行列であるため，各話者表現ベクトルによる適応パラメータへの分離は簡単に行うことができる．ここでは各行列のパラメータの絶対値を取ったものを可視化に用いた． 可視化の結果を図 7.2，図 7.3，図 7.4 に示す．図 7.2 から，1 層目においては入力話者による適応パラメータが出力話者のものよりも大きくなっていることがわかる．2 層から 4 層では，1 層と比較して大きな差は無いものの，出力層に近づくにつれて出力話者による適応パラメータが大きく，入力話者による適応パラメータが小さくなっていることが分かる．5 層では 1 層とは対称的に出力話者による適応パラメータが大きくなっており，入力話者による適応がほとんど行われていないことが分かる．この結果は本研究で用いてきた「入力層付近では話者性の除去，出力層付近では出力話者の話者性の付与が行われている」という仮定に沿うものとなっている．

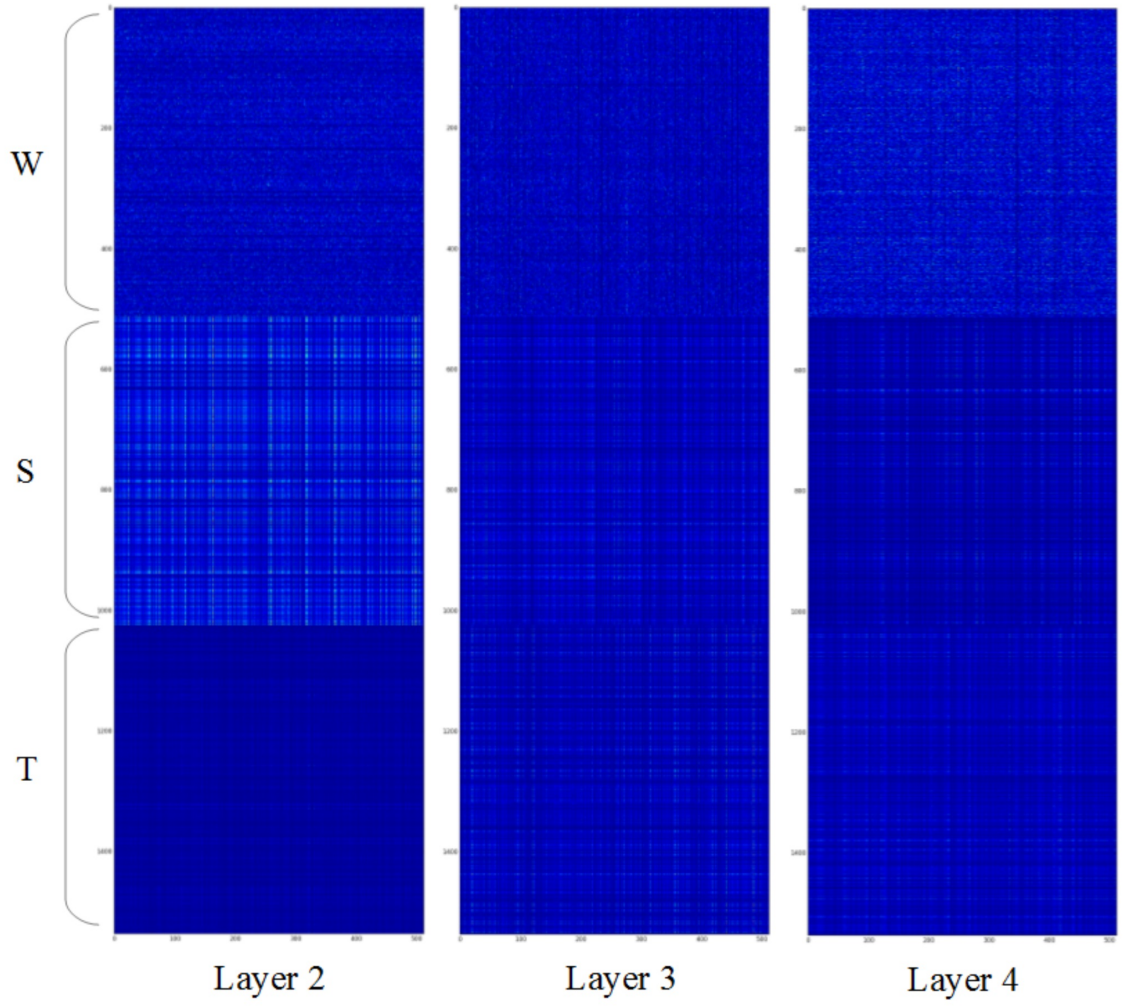

図 7.3: DNN の 2-4 層における話者非依存パラメータ (W) 及び FHL による入出力話者適応パラメータ (S, T)

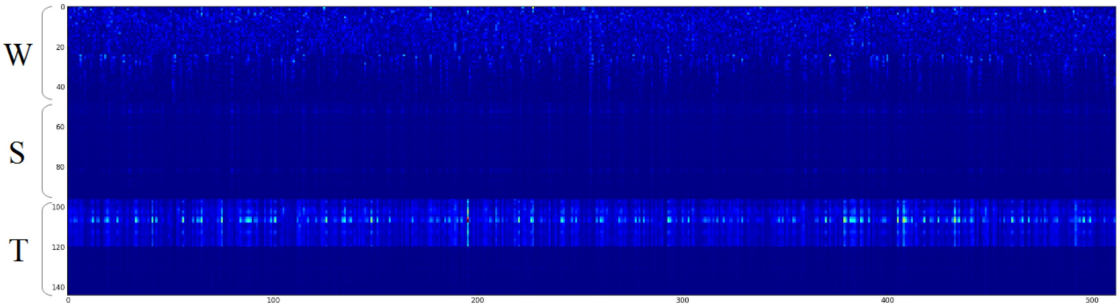

図 7.4: DNN の 5 層における話者非依存パラメータ (W) 及び FHL による入出力話者適応パラメータ (S, T)

第8章

結論

8.1　本論文のまとめ

本研究では，話者性の柔軟な制御を目的とした DNN による声質変換手法を提案した．話者性の柔軟な制御を目的とした手法としては，既存のモデルからパラメータを適応するというものが一般的であり，声質変換においては GMM による実装が多く取られている．この GMM による MAP 適応などの話者適応型声質変換手法を例に挙げ，複数話者コーパスを用いた学習を行うことで話者に依存していない，音声に共通な特徴量を抽出することが有用であると考えた．

また，DNN を用いた声質変換手法として，Deep Belief Nets による低次元空間表現を用いた声質変換という手法を挙げ，そこで用いられている，「深い階層を持つ DBN では各層のノード数で入力特徴量を表現するため，層の数が増えるほど入力特徴量が基底集合に近くなる」という考えを参考とした．この考えを仮定すると，DNN の pre-training において入力されるデータを複数の話者からなる音声コーパスとすることで，特徴量のフレーム毎に異なる情報，すなわち目的とする話者性への変換が RBM の深い層に集約されると考えた．

そこで，多言語音素認識タスクに用いられている手法を参考に，1 つの話者非依存サブネットワークと複数の話者依存サブネットワークからなる DNN による声質変換の枠組みを提案した．この手法では pre-training を複数の話者によって行うことで，前述したように話者性を深い層に集約し，その上で fine-tuning を出力話者毎の話者依存サブネットワークと話者非依存サブネットワークを導入して行うことで，話者非依存サブネットワークと話者依存サブネットワークの分岐点に集約した話者性を正規化するように学習を行う．これにより，学習に用いていない未知話者入力に対しても話者性が正規化されるために，柔軟な変換が可能となると考えられる．また，ANN において問題であった入力層に近い浅い層の学習をより効率的に行うことが可能となると考えられる．

実験の結果，pre-training に複数話者を用いることによる変換精度の向上と，提案手法による学習データ中の話者と未知話者の両方の入力に対する変換精度の向上を確認し，提案手法が入力話者に対して柔軟な変換が可能であることを示した．一方で，ここで提案したサブネットワークを用いた変換手法は，入力に関しては適応を行うことなく未知話者を用いた変換が可能であるものの，出力に関しては未知話者に関して適応を行うことができないという問題があった．

そこで，より柔軟かつ高精度な多対多声質変換を目的として，EVGMM と DNN を組み合わせた話者空間基底を用いた多対多声質変換手法を提案した．この手法では，EVGMM において構築される話者空間基底と平均話者を表すパラメータを基に，複数の DNN によって入力特徴量を基底成分と平均成分に分解し，分解した特徴量を再構成することで多対多声質変換を実現した．他の多対多声質変換手法との比較実験から，客観評価においては最も高い性能が得られ，手法の有用性が示された．主観評価においては一部他の多対多変換手法を僅かに下回る結果が見られたものの，それらはほぼ同程度の結果であり，学習データ量が多い場合の自然性においては最も良い結果が得られた．

加えて，提案手法の拡張として，パラレルデータフリーな共分散推定手法を導入することで全学習過程においてパラレルデータフリーな多対多声質変換手法を提案した．教師あ

り学習を行ったＧＭＭとの比較を行った結果，客観評価においては学習データ量が少ない場合でも提案手法がＧＭＭを僅かに下回っていると言う結果が得られた．その一方で，主観評価においては，学習データ量が少ない場合には自然性と個人性の両尺度において提案手法が GMM を上回っていた．

また，研究全体において前提としてきた DNN 声質変換における浅い層と深い層における変換の仮定について調べるため，FHL を用いた多対多声質変換を提案するとともに，その各層の適応パラメータの可視化を行った．その結果から，浅い層においては入力話者の，深い層においては出力話者の情報を用いた適応が強く行われており，仮定を裏付けるデータが得られた．

今後の方向性としては，主観評価において完全パラレルデータフリーの提案手法が個人性評価では教師あり学習に近い結果となっていたことから，自然性を補うフィルタリング手法の導入によって精度の向上を試みる，などの方策が考えられる．

謝辞

本研究を進めるにあたって, 峯松信明教授，斎藤大輔助教授には研究についての助言から，発表の場の斡旋，論文の添削など様々な面で大変お世話になりました．

柏木陽佑さん，内田秀継さんには週次ミーティングやグループ勉強会において研究についての様々な意見を頂いたり，発表原稿や論文の添削などもしていただきました．

研究室の先輩方，並びに後輩の方々にもいつもお世話になり，大変感謝しております．この場を借りて研究室の皆様方に心からお礼申し上げます．

参考文献

[1] A. Kain and M. W. Macon, “Spectral voice conversion for text-to-speech synthesis,” ICASSP, vol. 1, pp. 285–288, 1998.

[2] L. Deng, A. Acero, L. Jiang, J. Droppo, and X. Huang, “High-performance robust speech recognition using stereo training data,” ICASSP, pp. 301–304, 2001.

[3] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, “Voice conversion through vector quantization,” Proc. ICASSP, vol. 1, pp. 655–658, 1988.

[4] Y. Stylianou, O. Cppe, and E. Moulines, “Continuous probabilistic transform for voice conversion,” IEEE Trans. on Speech, and Audio Processing, vol. 6, no. 2, pp. 131–142, 1998.

[5] S. Desai, E.V. Raghavendra, B. Yegnanarayana, A.W. Black, and K. Prahallad,“Voice conversion using artificial neural networks,” ICASSP, pp. 3893–3896, 2009.

[6] A. Mouchtaris, J.V. der Spiegel, and P. Mueller, “Nonparallel training for voice conversion based on a parameter adaptation approach,” IEEE Trans. on Audio, Speech, and Language Processing, vol. 14, no. 3, pp. 952–963, 2006.

[7] C.H. Lee, and C.H. Wu, “Map-Based Adaptation for Speech Conversion Using Adaptation Data Selection and Non-Parallel Training,” INTERSPEECH, pp. 2254–2257, 2006.

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

[9] G.E. Hinton, L. Deng, D. Yu, G.E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, and B. Kingsbury, “Deep neural networks for acoustic modeling in speech recognition,” IEEE Signal Processing Magazine, pp. 82–97, 2012.

[10] 中鹿亘, 他, “Deep Belief Netsによる低次元空間表現を用いた声質変換の検討”, 日本音響学会春季研究発表会講演論文集, 3–P–46b, pp. 517–520, 2013. bibitemtensorD. Saito, K. Yamamoto, N. Minematsu, and K. Hirose, “One-to-Many Voice Conversion Based on Tensor Representation of Speaker Space,” INTERSPEECH, pp. 653–656, 2011.

[11] T. Toda, Y. Ohtani, H. Saruwatari, and K. Shikano, “Adaptive training for voice conversion based on eigenvoices,” IEICE TRANS. INF. &SYST., VOL.E93-D, NO.6, pp. 1589–1598, 2010.

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

[13] S. Matsuda, X. Lu, and H. Kashioka, “Automatic localization of a language-independent sub-network on deep neural networks trained by multi-lingual speech,” ICASSP, pp. 7359–7362, 2013.

[14] 中村哲, 鹿野清宏, “ファジィベクトル量子化を用いたスペクトログラムの正規化,” 日本音響学会学会誌, pp. 107–114, 1989.

[15] A. P. Dempster, N. M. Laird and D. B. Rubin, “Maximum Likelihood from Incomplete Data via the EM Algorithm,” Journal of the royal statistical society. Series B (methodological), pp. 1–38, 1977.

[16] Y. Freund and D. Haussler, “Unsupervised learning of distributions on binary vectors using two layer networks,” Computer Research Laboratory, 1994.

[17] G.E. Hinton, S. Osindero and Y. W. Teh, “A fast learning algorithm for deep belief nets.” Neural computation, pp. 1527–1554, 2006.

[18] P. Vincent, H. Larochelle, Y. Bengio, and P. A. Manzagol, “ Extracting and composing robust features with denoising autoencoders,” Proc. ICML, pp. 1096–1103, 2008.

[19] Vinod Nair, and Geoffrey E. Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines,” Proc. ICML. 2010.

[20] Maas, Andrew L., Awni Y. Hannun, and Andrew Y. Ng. ”Rectifier nonlinearities improve neural network acoustic models.” Proc. ICML. Vol. 30. No. 1. 2013.

[21] M. A. Ranzato, C. Poultney, S. Chopra and Y. LeCun, “Efficient Learning of Sparse Representations with an Energy-Based Model,” Advances in neural information processing systems, 2007.

[22] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, “Greedy layer-wise training of deep networks,” Advances in neural information processing systems, pp. 153–160, 2007

[23] G. E. Hinton, S. Osindero, and Y. W. Teh, “A fast learning algorithm for deep belief net,” Neural Computation, vol. 18, no. 7, pp. 1527–1554, 2006.

[24] L.J. Liu, L.H. Chen, Z.H. Ling, and L.R. Dai, “Spectral Conversion Using Deep Neural Netwotks Trained With Multi-Source Speakers,” ICASSP, pp. 4849–4853, 2015.

[25] 橋本 哲弥, 柏木 陽佑, 齋藤 大輔, 広瀬 啓吉, 峯松 信明, “話者依存サブネットワークを用いた深層学習による多対一声質変換, ” 日本音響学会秋季講演論文集, 2-Q-38, pp. 353–356, 2014.

[26] A. Kurematsu, K. Takeda, Y. Sagisaka, S. Katagiri, H. Kuwabara, and K. Shikano, “ATR Japanese speechdatabase as a tool of speech recognition and synthesis,” Speech Communication, vol. 9, pp. 357–363, 1990.

[27] H. Kawahara, I. Masuda-Katsuse, and A.de Cheveigne, “Restructuring speech representations using a pitch-adaptive time frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds,” Speech Communication, vol. 27, pp. 187–207, 1999.

[28] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Frontend factor analysis for speaker verification,” IEEE Trans. on Audio, Speech, and Language Processing, 19, 4, pp. 788–798, 2011.

[29] Wu. J, Wu. Z, and Xie. L, “On the Use of I-vectors and Average Voice Model for Voice Conversion without Parallel Data,” APSIPA, 2016.

[30] T. Hashimoto, D. Saito, and N. Minematsu, “Arbitrary speaker conversion based on speaker space bases constructed by deep neural networks,” APSIPA, 2016.

[31] “Jnas: Japanese newspaper article sentences,”
http://www.milab.is.tsukuba.ac.jp/jnas/instruct.html

[32] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” arXiv preprint arXiv:1207.0580, 2012.

[33] D. Erro, A. Moreno, and A. Bonafonte, “INCA algorithm for training voice conversion systems from nonparallel corpora,” IEEE Trans. on Audio, Speech, and Language Processing., vol. 18, no. 5, pp. 944-953, 2010.

[34] H. Silen, J. Nurminen, E. Helander, and M. Gabbouj, “Voice conversion for non-parallel datasets using dynamic kernel partial least squares regression,” INTERSPEECH, pp.373–377, 2013.

[35] T. Nakashika, T. Takiguchi and Y. Ariki, “PARALLEL-DATA-FREE, MANY-TO-MANY VOICE CONVERSION USING AN ADAPTIVE RESTRICTED BOLTZMANN MACHINE,” MLSLP, 2015.

[36] G. E. Hinton, and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks.,” Science 313, 5786, p. 504–507, 2006..

[37] K. Cho, A. Ilin and T. Raiko, “Improved learning of gaussian-bernoulli restricted boltzmann machines,” ICNN, Springer, pp. 10–17, 2011.

[38] T. Hashimoto, H. Uchida, D. Saito, and N. Minematsu, “Parallel-data-free many-to-many voice conversion based on DNN integrated with eigenspace using a non-parallel speech corpus,” INTERSPEECH, pp.1278–1282, 2017.

[39] Y. Ohtani, T. Toda, H. Saruwatari, and K. Shikano, “Manyto-many eigenvoice conversion with reference voice,” INTERSPEECH, 1623–1626, 2009.

[40] U. Hidetusgu, D. Saito, N. Minematsu, K. Hirose, “Statistical Acoustic-to-Articulatory Mapping Unified with Speaker Normalization Based on Voice Conversion,” INTERSPEECH, pp. 588–592, 2015.

[41] L. Samarakoon and Sim, K. C. “Factorized hidden layer adaptation for deep neural network based acoustic modeling,” IEEE Trans. on Audio, Speech, and Language Processing, vol. 24, no.12, pp. 2241–2250, 2016.

発表文献

国内研究会・全国大会

[1] 橋本 哲弥, 柏木 陽佑, 齋藤 大輔, 広瀬 啓吉, 峯松 信明, “話者依存サブネットワークを用いた深層学習による多対一声質変換, ” 日本音響学会秋季講演論文集, 2-Q-38, pp. 353–356, 2014.

[2] 橋本 哲弥, 柏木 陽佑, 齋藤 大輔, 広瀬 啓吉, 峯松 信明, “複数出力サブネットワークを有するディープニューラルネットに基づく声質変換, ” 信学技報, vol. 114, no. 365, SP2014–117, pp. 99–104, 2014.

[3] 橋本哲弥, 柏木陽佑, 齋藤大輔, 峯松信明, “Deep Neural Network を用いた話者空間基底への射影による声質変換,” 信学技報, vol. 115, no. 346, SP2015–70, pp. 1–6, 2015.

[4] 橋本 哲弥, 柏木 陽佑, 齋藤 大輔, 峯松 信明, “話者空間の基底成分を用いたディープニューラルネットワーク任意話者声質変換, ” 日本音響学会春季講演論文集, pp. 329–332, 2016.

[5] 橋本 哲弥, 齋藤 大輔, 峯松 信明, “話者基底成分への特徴量分解に基づくパラレルデータフリー声質変換の検討, ” 日本音響学会秋季講演論文集, ROMBUNNO.3-Q-35, 2016.

国際学会

[6] T. Hashimoto, D. Saito, and N. Minematsu, “Arbitrary speaker conversion based on speaker space bases constructed by deep neural networks,” APSIPA, 2016.

[7] T. Hashimoto, H. Uchida, D. Saito, and N. Minematsu, “Parallel-data-free many-to-many voice conversion based on DNN integrated with eigenspace using a non-parallel speech corpus,” INTERSPEECH, pp.1278–1282, 2017.

雑誌論文

[8] T. Hashimoto, D. Saito, and N. Minematsu, “Many-to-many and Completely Parallel-data-free Voice Conversion Based on Eigenspace DNN,” IEEE Trans. on Audio, Speech, and Language Processing, 2018, (conditionally accepted).