修　　士　　論　　文

畳み込み非負値行列分解を用いた音声の基底の教師無し学習と音素分類

Unsupervised Learning of Speech Bases and Phone Classification Using Convolutive Non-negative Matrix Factorization

指導教員　　**近山　隆　教授**

近山研究室
工学系研究科　電気系工学専攻

氏 名　**37-106488　針谷　航**

提 出 日　　**平成25年2月6日(水)**

概 要

知覚情報に含まれる基底を教師無しで学習させると, その結果が人間の脳内での処理と類似することが明らかにされている. このために教師無し学習は, 知覚情報処理の分野でとても注目されている. 音情報処理においては, 教師無し学習の一つである非負値行列分解 (NMF) が楽器音の抽出や音声の話者分離の研究で成果を上げている. 本稿では NMF を改良したモデルの一つである畳み込み非負値行列分解 (CNMF) を用いて, 音声の基底を抽出する手法を提案する. CNMF により抽出した音声の基底を用いた音素分類実験の結果, 音声の中に含まれる音韻情報を学習出来ることを確かめた. また CNMF のパラメータを最適化する手法として, 従来の補助関数法を用いた手法よりも Coordinate Descent を用いた手法の方がより速く最適解が求まることを示した.

目 次

図 目 次

表 目 次

第1章　序論

1.1　研究の背景と目的

近年計算機の性能が向上し大量のデータを処理できるようになったため, それに伴い教師無し学習を用いた研究が盛んに行われるようになった. この教師無し学習とは, 単に与えられたデータ内からデータを構成する基底を抽出する学習のことであるが, その処理は人間が脳内で行う知覚情報処理と似ていることが知られている. 例えば, Lee らは Deep Belief Networks (DBNs) を用いて様々な画像情報を学習させることにより, 人間の第 1 視覚野と第 2 視覚野で行われる処理を再現している [1]. 教師無し学習のこのような側面によりに, 教師無し学習は知覚情報処理において非常に注目され, 現在様々な研究がなされている.

知覚情報処理の一つである音声処理の分野では, 特徴量としてメル周波数ケプストラム係数 (MFCC) を, 音響モデルとして隠れマルコフモデル (HMM) を用いた手法が以前より成果を挙げている [2]. 現在の HMM を用いた音響モデルでは音声の特徴量から直接音素系列を推定しているが, 音素を構成する基底を教師無しで抽出することで音声認識の精度を高めようとする研究もなされている [3].

音声の基底を抽出する研究では, Convolutional Deep Belief Networks (CDBNs) などが成果を上げている. これはニューラルネットの一種であり, 原理は複雑であるが学習にかかる計算時間が短く, また Deep Learning というより複雑な学習も行える利点を持つ [4, 5]. 音声の基底を抽出する研究では, 抽出した基底が音声認識に役立てられるかを確認するために音素分類実験を行うことが多い [6].

しかしながら, CDBNs で提案されている高速な最適化法は全ての問題で収束が保証されていないなど問題点もいくつか指摘されている [4]. そのため, 同様の教師無し学習が行える他の手法により音声の基底を抽出することで, 更に音素分類実験の精度を高めることが出来ると推測される. そのような教師無し学習の一つとして, 畳み込み非負値行列分解 (CNMF) [7] が挙げられる.

CNMF は, 非負値行列分解 (NMF) [8] という教師無し学習の手法を時間的に遷移する基底も抽出出来るように改良したものである. NMF 自体も音情報処理で用いら

れており [9, 10], 音声の基底の抽出に適用出来ると考えられる. また NMF は原理が分かりやすく実装も手軽に行えるため, 様々な分野の研究で使われており, 他に改良が加えやすいという点も持っているため, NMF を応用した様々な手法が現在までに提案されている. そのため CNMF を用いた音素分類実験で成果を上げることが出来れば, これまでの知見から更なる改良が簡単に加えられると考えられる.

そこで本研究では, CNMF によって抽出した音声の基底を音素分類実験に用いて, 教師無し学習を用いた音素分類の精度を上げることを目的とする. 更に, 従来の CNMF を最適化する手法では計算にかなりの時間を要していたが, 近年 NMF で提案された Coordinate Descent (CD) を用いた最適化の手法が, 従来の手法に比べ格段に計算時間を短くなることが報告されている. [11]. そこて本研究では, CNMF の最適化に CD を導入することを提案し, 実際に計算時間が短くなることを示す.

1.2　本研究の位置づけ

本研究の最終的な目標は, 音声認識の精度を向上させるために音響モデルを再構築することである. この枠組みでは次の二つが重要な要素となる.

- 用いる音響特徴量
- 用いる音声のパターン列

本論文ではこの中の音声パターン列の構造に主眼をおき, CNMF を用いて音素列をさらに細かく表現する音声の基底列を抽出することを提案している. また, 抽出した基底が実際に音声認識に役立てられるかを音素分類実験を用いて検証している.

1.3　本論文の構成

本論文では, まず第 1 章で本研究の背景について述べた. これ以降, 第 2 章で CNMF による音声の基底の抽出を行う前に必要な処理, CNMF の原理や最適化の方法, 更に音素分類を行う SVM について説明する. 第 3 章では提案手法について説明し, 第 4 章では提案手法を評価するための実験の説明とその結果を示し, 考察を行った. 最後に第 5 章でまとめ, 今後の課題について述べる.

第2章　関連研究

本章では, まず初めに研究で使われた音声の特徴量について説明する. 次に音声の特徴量から音声の基底を教師無しで抽出する畳み込み非負値行列分解に付いて説明する. 最後に抽出した音声の基底を使って音素分類を行う際に用いたサポートベクターマシーンについて説明する.

2.1　音響特徴量の抽出

音響信号や音声信号を解析・処理する際は, 一般に音を構成している信号の周波数とその振幅値の情報が使われている. しかし離散音声信号は初め各時刻におけるサンプル値 (音圧) の情報しか与えられていないために, 音圧のサンプル値の情報から何かしらの処理を行い信号を周波数領域に変換する必要がある. 以後はサンプル値のみ持つ連続音声信号を, どのようにして周波数領域の情報に変換するかを説明する.

2.1.1　離散フーリエ変換

信号のサンプル値から信号に含まれる周波数の振幅値が必要な場合は, 離散フーリエ変換 (DFT) を用いて求めるのが一般的である. ここで時刻 t における音声信号のサンプル値を x_t, 信号に含まれる N 個の周波数のうち小さい方から第 n 番目のものの振幅値を A_n, 初期位相を ϕ_n とすると, 離散フーリエ変換は式2.1 のように書くことが出来る.

$$A_n \exp(j\phi_n) = \sum_{t=0}^{N-1} x_t \exp\left(-j\frac{2\pi}{N}nt\right) \tag{2.1}$$

これとは反対に, 信号に含まれる各周波数の振幅値と位相が求まっている場合, 次の逆フーリエ変換 (IDFT) により元の音声信号へ復元することも出来る.

$$x_t = \sum_{n=0}^{N-1} A_n \exp(j\phi_n) \exp\left(j\frac{2\pi}{N}nt\right) \tag{2.2}$$

離散フーリエ変換を用いて周波数領域に変換する場合, 音声信号に含まれる周波数成分の情報は不変であると仮定して計算される. しかし実際の音声では時間に応じてその信号に含まれる周波数成分の情報は変わってしまう. そこで連続音声を周波数領域に変換する際は, 音声信号を短時間のフレームごとに区切りそれぞれのフレーム内でフーリエ変換を施して各時刻における音声信号の周波数情報を得る. このような形で音声信号にフーリエ変換を施すことを短時間フーリエ変換 (STFT) と呼ぶ.

連続音声信号をフレームで区切る場合, 隣接するフレームはある程度重なる形で区切られることが多い. 一般にフレーム一つの中に含まれるサンプルの数は”フレームサイズ”, 隣接するフレーム間のサンプルのずれは”フレームシフト”と呼ばれている. フレームシフトの値をどのように設定すれば良いかは行いたい処理に応じて決める必要があるが, フレームサイズの半分とする場合が多い.

音声信号処理においては各フレームにおける周波数情報のうち, 各周波数の振幅値のみ用いられることが多く, 位相情報はあまり用いられない. STFT を施した後は, 各周波数の振幅値の情報だけが残されて, その後様々な処理に使われる. フーリエ変換を施した後に横軸を周波数, 縦軸を各周波数の振幅値として得られた情報を並べたものは, スペクトラムと呼ばれる. 更に, 音声信号処理 (音響信号処理) においては各フレームにおける各周波数の振幅値を行列の形で並べたもの, つまり各フレームのスペクトラムを時間ごとに並べたものをスペクトログラム (図.2.1) と呼んでおり, 様々な音響 · 音声処理の分野で用いられている.

スペクトログラムは音声情報を表す行列であるが行列のまま見ると分かりづらいため, 図.2.2 に示すような色付きの図で表すことが多い. スペクトログラムを色が付いた図で表す場合は, 図の横軸を時刻, 縦軸を周波数とし, それぞれの場所における振幅値の値を色の濃淡で表す. 図.2.2 においては, 行列の値が大きいところは暖色で表し, 値が小さいところは寒色で表している.

2.1.2　窓関数

フーリエ変換では, フレーム内の信号が無限に繰り返されると仮定して周波数領域に変換している. そのため連続音声信号に対して単純に短時間フーリエ変換を施すと, 各フレームの両端のサンプル値が大きく異なる場合には不連続な形で信号が繰り返されてしまい, フーリエ変換後の情報に不要な雑音が多く入ってしまう. そこでフーリエ変換を各フレームに施す前に, フレーム内のサンプル値に窓関数 (図.2.3) を掛け不要な雑音が入らないようにする必要がある. この作業は窓掛けと呼ばれて

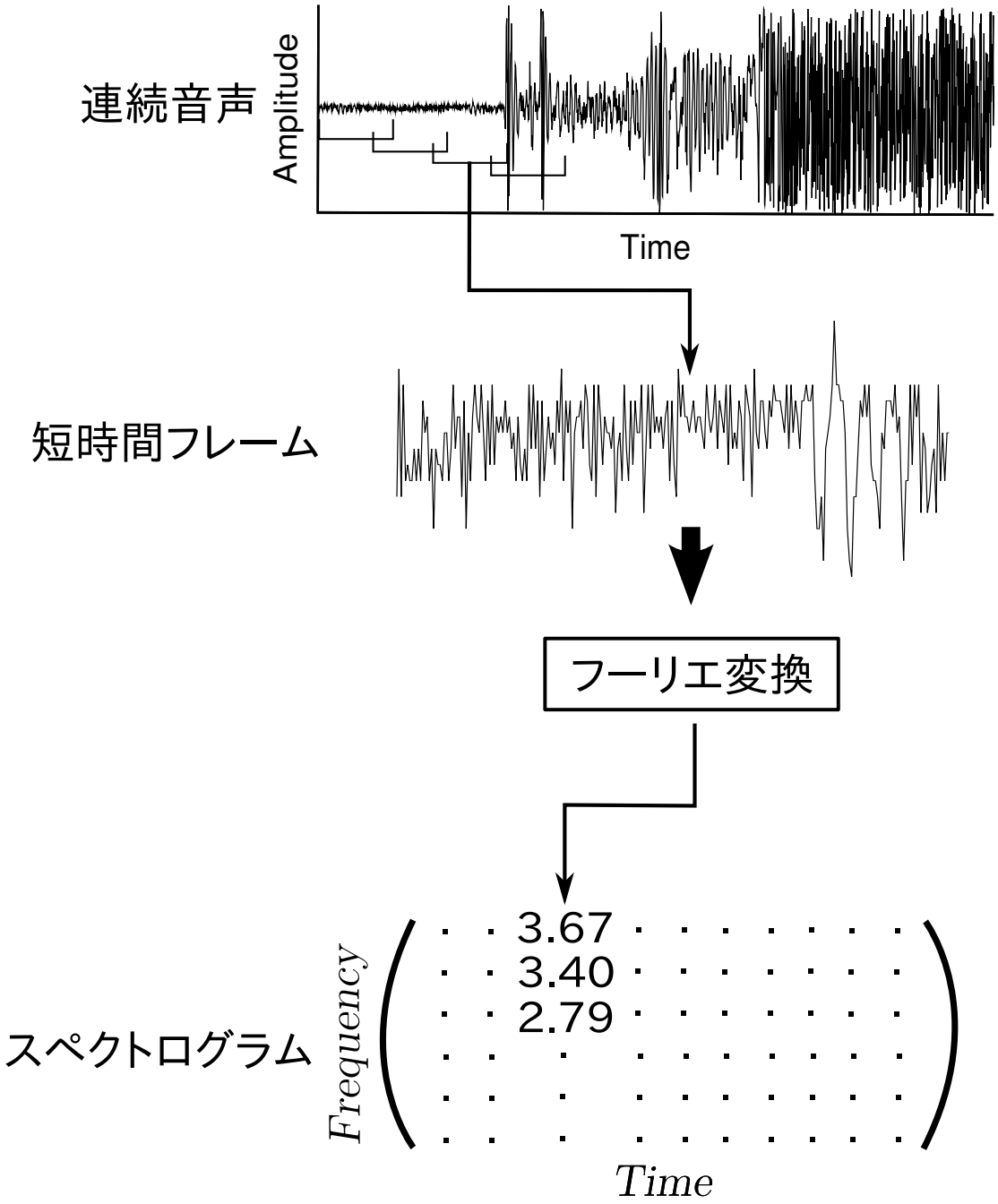

図 2.1: スペクトログラム変換の手順

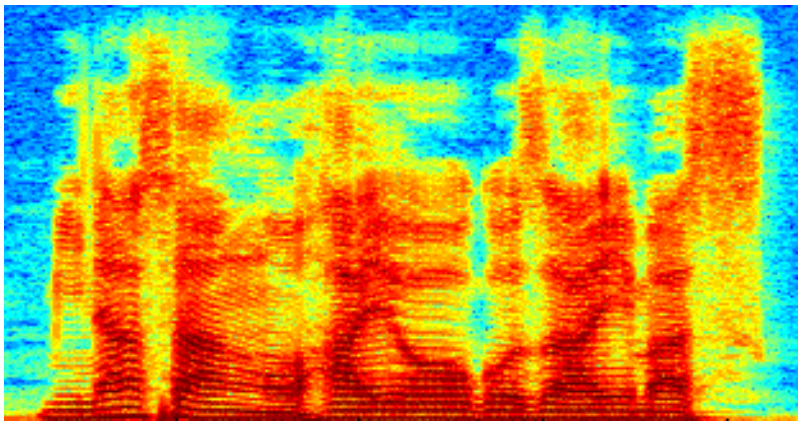

図 2.2: 色つきのスペクトログラム

いる.

窓関数には次の二つのことが要求される.

- 周波数分解能の良さ.
- ダイナミックレンジの広さ.

ここで挙げた二つの要件はトレードオフの関係にあるため, どちらも満たす理想の窓関数は存在しておらず, 用途に応じて適切な窓関数を用いる必要がある. 現在までに様々な窓関数が提案されているが, ここではその窓関数のうちで特に使われているものをいくつか紹介する. なお N はフレーム内のサンプル数を表し, t は各サンプルの番号を表している.

$$w(t) = 1 \quad (\text{方形窓}) \tag{2.3}$$

$$w(t) = 0.5 - 0.5\cos\left(2\pi\frac{t}{N}\right) \quad (\text{ハニング窓}) \tag{2.4}$$

$$w(t) = 0.52 - 0.48\cos\left(2\pi\frac{t}{N}\right) \quad (\text{ハミング窓}) \tag{2.5}$$

$$w(t) = 0.42 - 0.5\cos\left(2\pi\frac{t}{N}\right) + 0.08\cos\left(4\pi\frac{t}{N}\right) \quad (\text{ブラックマン窓}) \tag{2.6}$$

どの窓関数を用いるかは問題に応じて決める必要があるが, 本稿ではその中でも特に広く使われているハミング窓 (式.2.5) を使用することとした.

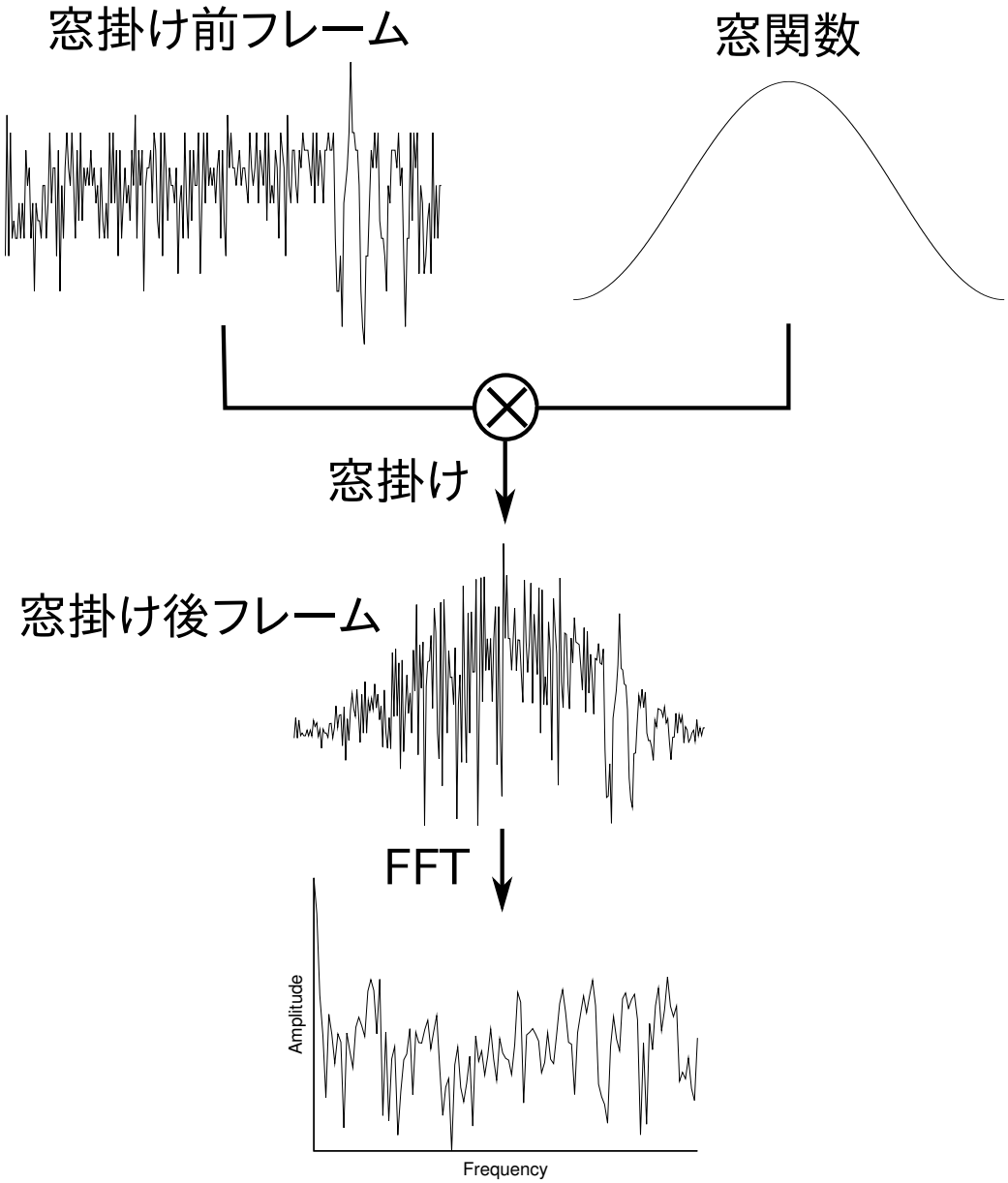

図 2.3: 窓掛けの例

2.1.3　メルスペクトログラム

人間の音高の知覚特性に関して, 低周波においては分解能が高く高周波においては低いことが知られており, 更に詳しく説明すると人の周波数分解能は対数関数を用

いて近似することが出来る. この知覚特性を反映した尺度はメル尺度と言われ, 幾つかのものが提案されている. メル尺度により周波数 f[Hz] をメル周波数 f_{mel} に変換する式の一つを以下に示す.

$$f_{mel}(f) \quad = \quad 2595 \log_{10} \left(1 + \frac{f}{700} \right) \tag{2.7}$$

本研究では式.2.7 を用いて, 周波数をメル周波数へと変換している. このメル周波数軸の上においては, 人間の周波数分解能はどの場所においても等しくなっている.

先で述べたスペクトログラムは人間の分解能とは異なる形で音響的特徴量を抽出していることになるので, 人間の知覚特性を反映したメル周波数を取り入れた形でスペクトログラムを作り直す必要がある. そのため, スペクトログラムの各フレームにおけるスペクトラムにメルフィルタバンクを掛けてメルスペクトラムに変換する作業を本研究では行った (図.2.4).

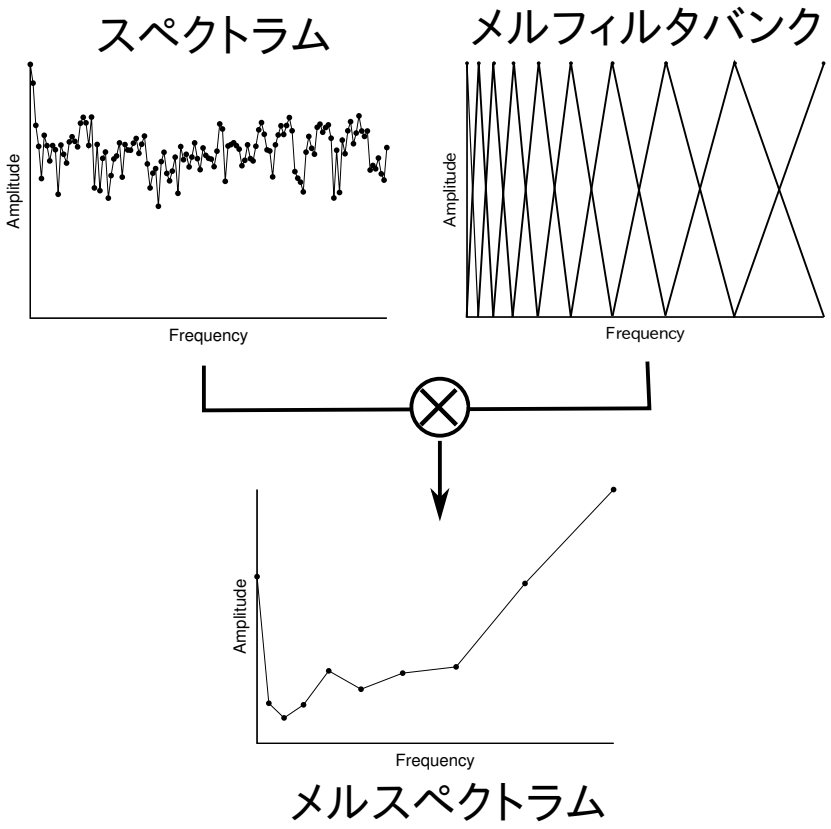

図 2.4: メルスペクトラムへの変換

メルフィルタバンクとはメル周波数上で等間隔に配置された三角窓の集合を指しており, フーリエ変換で得た周波数情報は各三角窓と掛け合わされてその内積が計算され, 最終的に内積の値が各三角窓から出力される. この作業により変換されたスペクトログラムはメルスペクトログラムと呼ばれている. 本研究ではここで説明したメルスペクトログラムを音声の特徴量として連続音声から音声の基底を抽出し, 音素分類実験を行っている.

2.2　音声の教師無し学習

2.2.1　非負値行列分解

非負値行列分解 (NMF) は, 任意の非負値を要素に持つ入力行列 $\boldsymbol{V}$ を二つの非負値を要素に持つ行列 $\boldsymbol{W}$ と $\boldsymbol{H}$ の積で近似し, 入力行列に潜在する基底を抽出する手法である (式.2.8) [8].

$$\begin{aligned} \boldsymbol{V} &\approx \boldsymbol{W}\boldsymbol{H} \\ &= \boldsymbol{A} \end{aligned} \tag{2.8}$$

このとき行列 $\boldsymbol{W}$ は入力行列に潜在する基底を表し, 行列 $\boldsymbol{H}$ は入力行列の各列が抽出した基底のいずれと一致するかを示す重みを表している. 以後, 行列 $\boldsymbol{W}$ は基底行列, 行列 $\boldsymbol{H}$ は重み行列, 基底行列と重み行列の積 $\boldsymbol{W}\boldsymbol{H}$ を近似行列 $\boldsymbol{A}$ と呼ぶこととする. NMF による分解の例を図.2.5 に示す.

NMF は原理が非常に単純であり, それによりに実装も容易に行えるために, 様々な情報処理の分野で研究が盛んに行われている. また NMF に様々な改良を加えたアルゴリズムも提案されており [12, 13], このように改良を加えやすいところも広く使われる一因となっている.

音情報処理に限って見ると, NMF は音楽情報処理や音声の話者情報に関する処理の研究では多く使われているものの [9, 10], 音声の音韻情報に関する処理では余り研究されていない. そこで本稿では, NMF が他の音情報処理と同様に, 音声の音韻情報に関する処理でも有用であるか確かめるために様々な実験を行った.

NMF を音楽情報や音声情報に適用する際には, 入力行列にスペクトログラムが使われるのが一般的である. しかし, 本研究では人間の知覚により近い形で音声の基底を抽出し音声認識などに役立てたい意図があるため, 入力行列にはメルスペクトログラムを用いることにした.

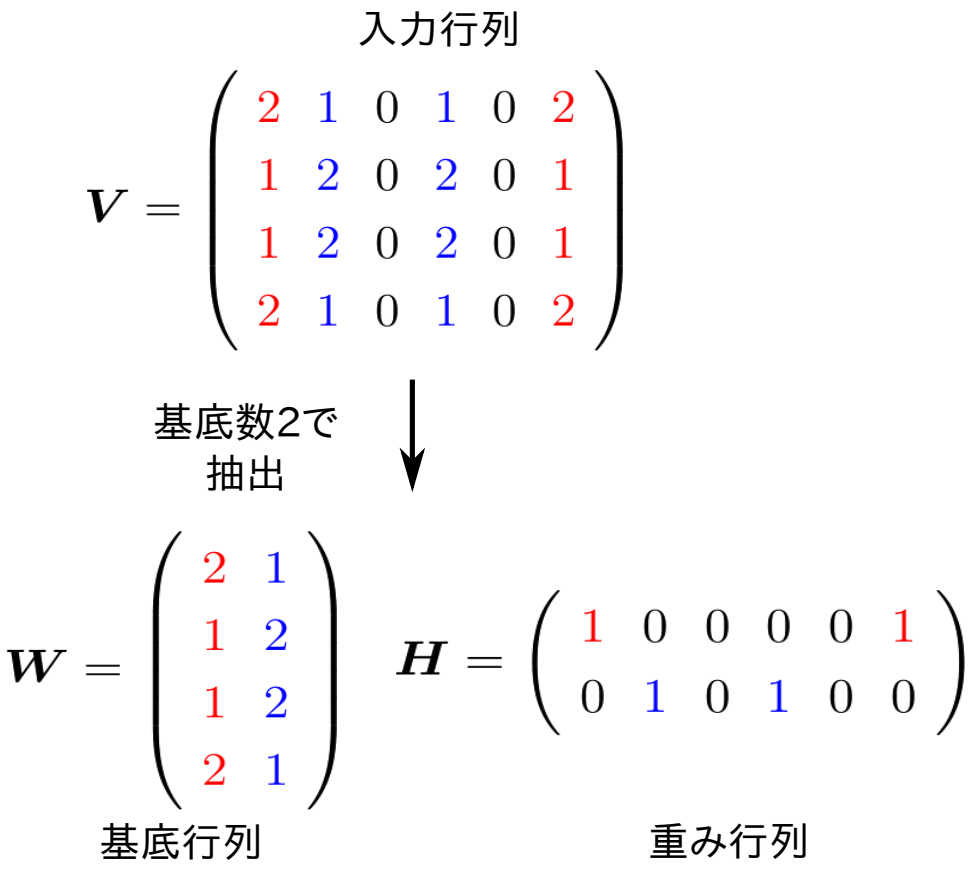

図 2.5: NMF の例

2.2.2　目的関数

NMF を用いて入力行列を分解するとき, 必ずしも完璧に分解出来るとは限らない. そこで NMF では目的関数を設定し, その目的関数を最小とする基底行列と重み行列の組を解とする. 目的関数には様々なものが提案されているが, 主にフロベニウス距離 (式.2.9) と KL-Divergence (式.2.10) の二つが用いられる. ここで行列 $\boldsymbol{B}$ の第 m 行 · 第 n 列の要素を b_{mn} と表すこととする.

$$\begin{aligned} D(\boldsymbol{V}|\boldsymbol{A}) &= \sum_{i,k} (v_{ik} - a_{ik})^2 \\ &= \sum_{i,k} \left(v_{ik} - \sum_{j} w_{ij} h_{jk} \right)^2 \end{aligned} \tag{2.9}$$

$$
\begin{aligned}
D(\boldsymbol{V}|\boldsymbol{A}) &= \sum_{i,k}\left\{v_{ik}\ln\left(\frac{v_{ik}}{a_{ik}}\right) - v_{ik} + a_{ik}\right\} \\
&= \sum_{i,k}\left\{v_{ik}\ln\left(\frac{v_{ik}}{\sum_j w_{ij}h_{jk}}\right) - v_{ik} + \sum_j w_{ij}h_{jk}\right\} \qquad (2.10)
\end{aligned}
$$

1 次元のフロベニウス距離, KL-Divergence の発散値 $D(y|x)$ を図.2.6 に示す. 図.2.6 よりフロベニウス距離は負の値を取ることが許されているが, KL-Divergence は取ることが許されていないことが確認できる. NMF では要素の値を全て非負値という制約を設けているため, KL-Divergence はフロベニウス距離よりも使われている傾向がある.

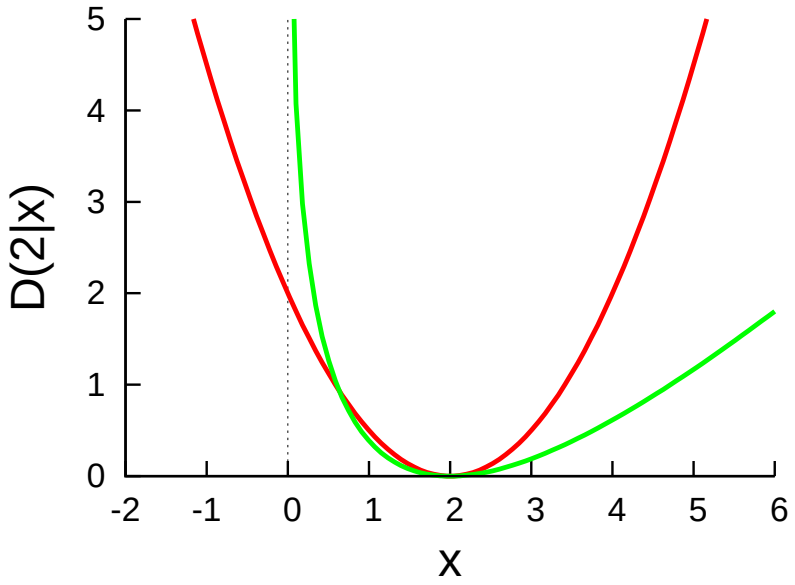

図 2.6: フロベニウス距離 (赤) と KL-Divergence(緑) ($y = 2$)

フロベニウス距離においては, 入力行列 $\boldsymbol{V}$ の各要素は近似行列 $\boldsymbol{A}$ の各要素を平均値とする正規分布に従うと仮定している. 同様にして KL-Divergence においては, 入力行列 $\boldsymbol{V}$ の各要素は近似行列 $\boldsymbol{A}$ の各要素を平均値とするポアソン分布に従うと仮定している. どの目的関数を用いるかは解く問題に応じて決める必要があるが, 本研究においてはフロベニウス距離を用いることにした.

2.2.3　更新式

設定した目的関数を最小とする基底行列と重み行列の組を求めるとき, 目的関数を解析的に解くことは現実的に不可能である. そこで解を求める際には, あらかじめ基底行列と重み行列の各要素に任意の初期値を与え, 各要素の値を目的関数の値が小さくなる方向に何度も更新し最終的に得られた値を解とするのが一般的である.

更新式の導出方法には様々な手法が提案されているが, 現在 NMF において主流であるものは以下の二つである.

1. 補助関数法を用いた手法
2. Coordinate Descent(CD) を用いた手法

NMF が始めて提案された際には 1 つ目の手法が使われており, その後現在に至るまで速く最適解に収束するアルゴリズムとして広く使われてきた. 2 つ目の手法は近年新しく提案された手法で, 基底行列 (または重み行列) の要素を一つだけ動かし, その要素に関して目的関数を最小とする作業を繰り返して最適解を求める手法である. CD は以前より使われていた 1 つ目の手法に比べ速く収束するという報告もあり, とても注目されている手法である.

CNMF においては CD を用いて最適化を行った研究がまだなされていないが, CNMF においても速く収束することが推定される. そこで本稿においてはどちらの手法も扱うこととし, それらの優劣に関して実験を行い調べ議論することにした.

2.2.4　補助関数法

補助関数法 (AF 法) とは, 目的関数 $D(\theta)$ を最小化するパラメータ θ を求める際に式.2.11, 式.2.12 を満たす補助関数 $C(\theta, \phi)$ を設定し, 補助関数を最小化するパラメータ ϕ と θ を交互に求めて目的関数を最小化する手法である [14].

$$D(\theta) \leq C(\theta, \phi) \tag{2.11}$$

$$D(\theta) = \min_{\phi} C(\theta, \phi) \tag{2.12}$$

具体的には次の二つのステップを繰り返し目的関数を最小化するパラメータ θ を求める. ここで t 回更新した後のパラメータ θ, ϕ の値をそれぞれ $\theta^{(t)}$, $\phi^{(t)}$ とする.

1. 補助関数 $C(\theta, \phi)$ をパラメータ $\theta = \theta^{(t)}$ に固定した状態で, 補助関数を最小化するパラメータ ϕ を求める.

$$\phi^{(t+1)} = \min_{\phi} C(\theta^{(t)}, \phi) \tag{2.13}$$

2. 補助関数 $C(\theta, \phi)$ をパラメータ $\phi = \phi^{(t+1)}$ に固定した状態で, 補助関数を最小化するパラメータ θ を求める.

$$\theta^{(t+1)} = \min_{\theta} C(\theta, \phi^{(t+1)}) \tag{2.14}$$

上に示した 2 段階の更新を行うとき, $D(\theta^{(t)})$ と $D(\theta^{(t+1)})$ の間には次の不等式が成り立つ.

$$\begin{aligned} D(\theta^{(t)}) &= C(\theta^{(t)}, \phi^{(t+1)}) \\ &\geq C(\theta^{(t+1)}, \phi^{(t+1)}) \\ &\geq C(\theta^{(t+1)}, \phi^{(t+2)}) \\ &= D(\theta^{(t+1)}) \end{aligned} \tag{2.15}$$

これにより, 補助関数法を用いた更新では目的関数が必ず減少することが保証されている. ここで補助関数法によるパラメータの更新の流れを図.2.7 に示す. 図で確認できるように, パラメータ θ と ϕ を交互に更新することで, 目的関数の値が徐々に小さくなっていく.

2.2.5　イェンゼンの不等式

関数 $f(x)$ が凸関数で有るとき, 式.2.16 が成り立つ.

$$\sum_i p_i f(x_i) \geq f\left(\sum_i p_i x_i\right) \tag{2.16}$$

ここで実数 p_i は式.2.17 を満たすものとする.

$$\sum_i p_i = 1 \tag{2.17}$$

イェンゼンの不等式は, 補助関数法を用いて NMF の更新式を求める際に必要となる. 具体的には提案手法の更新式の導出部分で説明する.

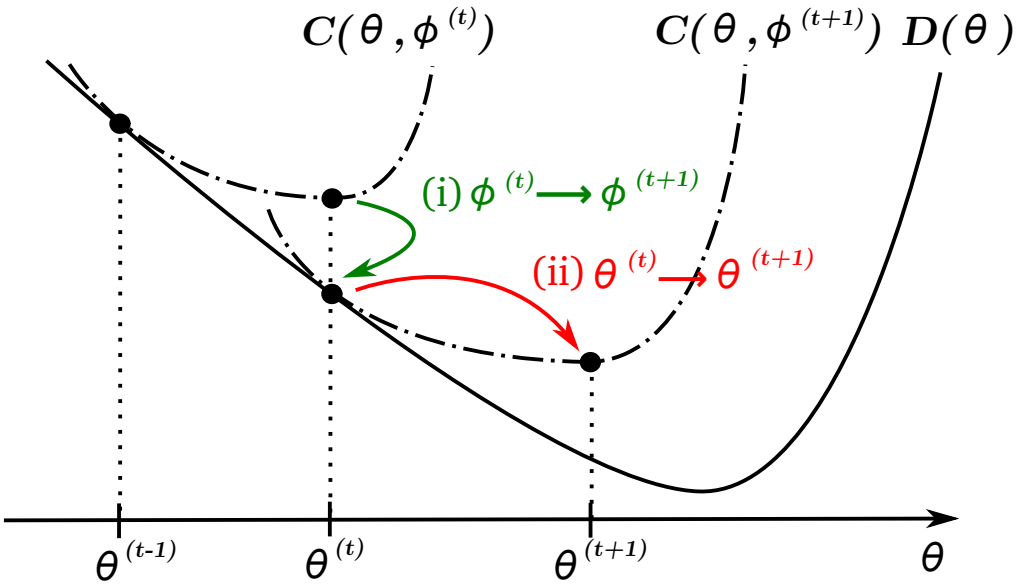

図 2.7: 補助関数法によるパラメータの更新

2.2.6　Coordinate Descent

Coordinate Descent (CD) は目的関数 $D(\theta)$ のパラメータ θ を $\theta + \epsilon$ に更新して目的関数を最小化する手法である [11]. このときパラメータ θ を更新した目的関数は式.2.18 のように書くことが出来る.

$$D(\theta + \epsilon) \quad = \quad D_1(\theta) + D_2(\theta, \epsilon) \tag{2.18}$$

ここで目的関数 $D(\theta + \epsilon)$ をパラメータ ϵ で偏微分することでを, 目的関数を最小化するための最適な更新量 $\hat{\epsilon}$ を求めることが出来る. $\hat{\epsilon}$ は式.2.19 を満たす.

$$\left.\frac{\partial D_2(\theta, \epsilon)}{\partial \epsilon}\right|_{\epsilon=\hat{\epsilon}} \quad = \quad 0 \tag{2.19}$$

パラメータが 2 次元以上のベクトルで表される場合は, ベクトルの各要素ごとにこの更新を行う. 各更新においては必ず目的関数が減少する方向に更新されるため, 更新を繰り返すことで最適解に収束することが保証されている. Coordinate Descent を用いたパラメータの更新の模式図を図.2.8 に示す. 模式図では, 目的関数の値が小さい点は暖色で表されており, 値が大きい点は寒色で表されている.

Coordinate Descent は元々別の分野の最適化問題で用いられていた手法であったが, 近年 NMF でも用いられその収束の速さより非常に注目されている.

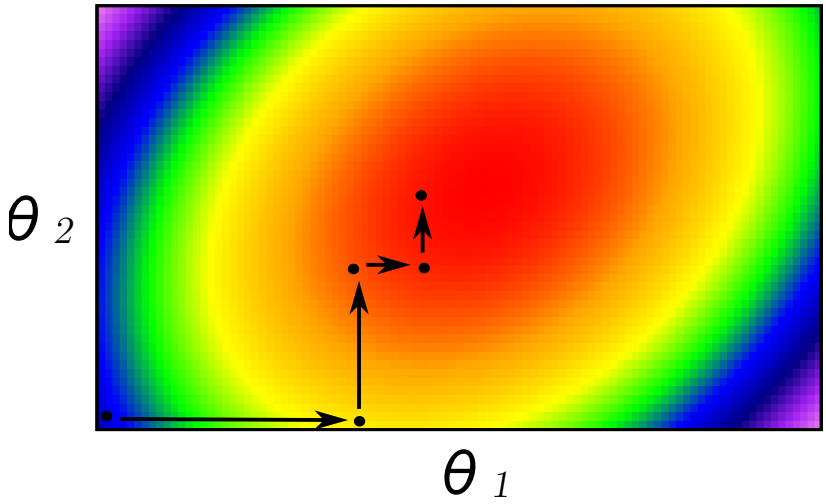

図 2.8: Coordinate Descent によるパラメータの更新

2.2.7　畳み込み非負値行列分解

NMF を用いて音声の基底を抽出するとき, 定常的な基底しか抽出することが出来ない. そのために時間的に遷移する音声の基底は抽出することが出来ない. しかし全ての音声の基底が定常的に表されるとは考えにくい. そこで本研究では, NMF を時間的に遷移する基底も抽出出来るように改良した, 畳み込み非負値行列分解 (CNMF) [7] を使うこととした. CNMF は基本的に NMF と同じ手法であるが, 基底行列が NMF の場合と異なり, 抽出する基底の長さと同じ個数用意されている. CNMF は数式で表すと, 式.2.20 の形となる.

$$\begin{aligned} \boldsymbol{V} &\approx \sum_t \boldsymbol{W}_t \left(\boldsymbol{H}\right)^{t\rightarrow} \\ &= \boldsymbol{A} \end{aligned} \tag{2.20}$$

$\boldsymbol{W}_t$ は時刻 t における各基底の特徴量を表している. このように基底行列を複数用意することで, 時間的に遷移する基底を表現することが可能となった. ちなみに, 式.2.20 に現れる演算子 $(\boldsymbol{B})^{t\rightarrow}$ は, 行列 $\boldsymbol{B}$ の各列を t 個矢印の方向へ動かす作用を持つ (表.2.1). この移動により値が無くなった列の要素は, 全て 0 で埋めるものとする.

CNMF は遷移する基底を抽出することが出来るため, 主に音情報処理の分野で利用されており, 楽器音の抽出や話者分離の分野では成果をあげている [15, 16]. CNMF

表 2.1: 演算子 $(\cdot)^{t\rightarrow}$ の例

$$\boldsymbol{B} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{pmatrix}$$

$$(\boldsymbol{B})^{1\rightarrow} = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 0 & 5 & 6 & 7 \end{pmatrix}, (\boldsymbol{B})^{\leftarrow 2} = \begin{pmatrix} 3 & 4 & 0 & 0 \\ 7 & 8 & 0 & 0 \end{pmatrix}$$

による入力行列の分解例を図.2.9 に示す.

2.3 Support Vector Machine

Support Vector Machine (SVM) は 2 クラス分類器の一つであり, 与えられたデータを分類する超平面を求める手法のことである. 以下では, まず線形カーネル SVM による分類について説明する. 次に, 分類器の性能をより実世界のデータに近づけるための手法として, ソフトマージン SVM について説明する. 更に, 線形分離では行えない複雑な分類が可能である非線形カーネル SVM について説明し, 最後に 2 クラス分類器を多クラス分類器に拡張する One-versus-All 法について説明する.

2.3.1 線形カーネル SVM

線形カーネル SVM とは, 次に示す識別関数 $f(\boldsymbol{x})$ の正負に応じて与えられたサンプルを 2 クラスに分類する分類器のことである [17]. ここで入力ベクトル $\boldsymbol{x}$ と重みベクトル $\boldsymbol{w}$, 定数 b を用いて識別関数 $f(\boldsymbol{x})$ は次のように表すことが出来る.

$$f(\boldsymbol{x}) \quad = \quad \boldsymbol{w}^T\boldsymbol{x} + b \tag{2.21}$$

重みベクトル $\boldsymbol{w}$ と定数 b は与えられた訓練データを分類する超平面の法線ベクトルを表しており, 訓練データが線形分離可能である場合には重みベクトルと定数を求めることが出来る. 訓練データが線形分離可能な場合には重みベクトルの解として様々なものが考えられるために, SVM では式.2.22 に示す拘束条件を設け解を冗長性を減らしている.

$$\min_i \left| \boldsymbol{w}^T\boldsymbol{x}_i + b \right| \quad = \quad 1 \tag{2.22}$$

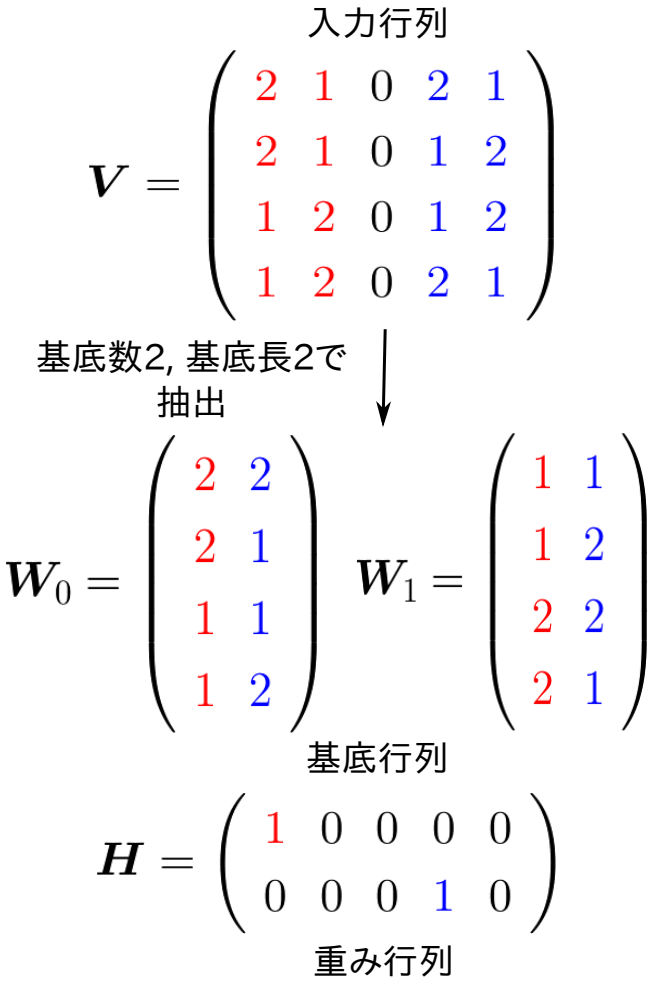

図 2.9: CNMF の例

この制約によって, 分離超平面までの距離が最も小さいサンプルとその超平面との距離 $d_{\min}$ は式.2.23 で表すことが出来る.

$$\begin{aligned} d_{\min} &= \min_i \frac{\left|\boldsymbol{w}^T \boldsymbol{x}_i + b\right|}{\|\boldsymbol{w}\|} \\ &= \frac{1}{\|\boldsymbol{w}\|} \end{aligned} \tag{2.23}$$

この距離 $d_{\min}$ はマージンと呼ばれており, この値が大きくなるほど訓練データではないデータに対する分類性能 (汎化性能) が良くなることが知られている. そこで,

SVM では距離 $d_{\min}$ を最大化する重みベクトル $\boldsymbol{w}$ を解とする．このことはマージン最大化と呼ばれ, SVM の特徴の一つとしてあげられる．このとき解くべき問題について考えると,

$$\begin{aligned} & \max_{\boldsymbol{w}} d_{\min} \\ \Leftrightarrow & \max_{\boldsymbol{w}} \frac{1}{\|\boldsymbol{w}\|} \\ \Leftrightarrow & \min_{\boldsymbol{w}} \|\boldsymbol{w}\| \end{aligned} \tag{2.24}$$

となるので, 距離 $d_{\min}$ を最大化する条件は $\|\boldsymbol{w}\|$ を最小化する条件に書き換えることが出来る．

ここで入力サンプル $\boldsymbol{x}_i$ のクラスラベル y_i を 1 か -1 の 2 値とすると, 分離可能な超平面を求める問題は次の問題として表現できる．

$$\begin{aligned} & \min_{\boldsymbol{w}} \|\boldsymbol{w}\| \\ & \text{subject to } \forall_i y_i \left(\boldsymbol{w}^T \boldsymbol{x} + b \right) \geq 1 \end{aligned} \tag{2.25}$$

この問題はラグランジュの未定乗数法を用いて, 次の形に書き直すことが出来る．

$$\min_{\boldsymbol{\alpha}} \ -\sum_i \alpha_i + \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j \tag{2.26}$$

$$\text{subject to } \sum_i \alpha_i y_i = 0, \ \forall_i \alpha_i \geq 0 \tag{2.27}$$

この時, Karush-Kuhn-Tucker 条件から

$$\forall_i \alpha_i \left(y_i \boldsymbol{w}^T \boldsymbol{x} - 1 \right) \ = \ 0 \tag{2.28}$$

が満たされる．この式より, 分離超平面から最も近い距離にあるサンプルに関しては $\left(y_i \boldsymbol{w}^T \boldsymbol{x} - 1 \right) = 0$ となり, それ以外のサンプルに関しては $\left(y_i \boldsymbol{w}^T \boldsymbol{x} - 1 \right) \neq 0$, すなわち $\alpha_i = 0$ となる．これは, SVM において境界面に最も近いサンプルの情報だけが重みベクトルの計算に用いられ, その他のサンプルの情報は全く用いられないこと意味している．SVM では, この特性によって分離超平面を求めるために必要な計算量を減らすことが出来, 高速な分離超平面の学習を可能としている．

分離超平面を求めるために使われるサンプル, つまり分離超平面に最も近いサンプル達はサポートベクターと呼ばれていて, これが手法の名称の由来となっている．

サポートベクターの集合を SV と表すとき, 式.2.26 から求められる最適解 $\boldsymbol{\alpha}^*$ によって重みベクトルの最適解 $\boldsymbol{w}^*$ は次のように表される.

$$\boldsymbol{w}^* = \sum_{\boldsymbol{x}_i \in SV} \alpha_i^* y_i \boldsymbol{x}_i \tag{2.29}$$

$$b = y_k - (\boldsymbol{w}^*)^T \boldsymbol{x}_k \quad (\boldsymbol{x}_k \in SV) \tag{2.30}$$

以上より 2 値クラス分類に用いられる識別関数 $f(\boldsymbol{x})$ は次のように表される.

$$\begin{aligned} f(\boldsymbol{x}) &= (\boldsymbol{w}^*)^T \boldsymbol{x} + b \\ &= \sum_{\boldsymbol{x}_i \in SV} \alpha_i^* y_i \boldsymbol{x}_i^T \boldsymbol{x} + b \end{aligned} \tag{2.31}$$

また, 求める分離超平面の方程式は次のように表される.

$$\sum_{\boldsymbol{x}_i \in SV} \alpha_i^* y_i \boldsymbol{x}_i^T \boldsymbol{x} + b = 0 \tag{2.32}$$

線形カーネル SVM の模式図を図.2.10 に示す. 図において中が塗りつぶされているサンプルはサポートベクターを, 赤線は分離境界面を表している.

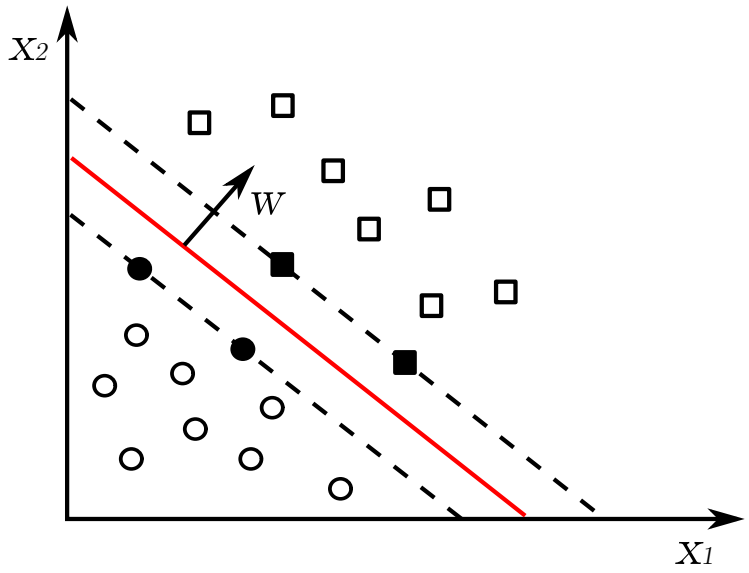

図 2.10: 線形カーネル SVM の模式図

2.3.2　ソフトマージン SVM

先ほど説明した線形カーネル SVM は, 与えられたデータが完全に線形分離可能で有ることを前提としており, このような SVM はハードマージン SVM と呼ばれている. しかし, 現実に存在する多くの問題は線形分離不可能で有り, そのような場合には, ハードマージン SVM を用いて最適な分離超平面を求めることは不可能である.

そこでこの問題を解決するために提案された手法が, ソフトマージン SVM である. これはデータを分離する超平面を求める際に, 多少の識別誤りを許しながらもある程度の分離精度を残すように調節した SVM のことである. 識別誤りが許されることにより, 求められなかった分離行平面が求まるようになる. このソフトマージン SVM には様々な手法が提案されているが, その中でも特に広く用いられている C-SVM [18] を本研究では使用することにした. 以後は C-SVM について説明する.

C-SVM では満たすべき制約条件を緩和するために, 次に示すスラック変数を最適化する式に導入する.

$$\forall_i \xi_i \geq 0 \tag{2.33}$$

次に最適化の目的関数と最適化を行う際の制約条件をスラック変数と定数 C を用いて次式のように変更する.

$$\frac{1}{2} \min_{\boldsymbol{w}} \|\boldsymbol{w}\| + C \sum_i \xi_i$$

$$\text{subject to } \forall_i y_i \left(\boldsymbol{w}^T \boldsymbol{x} + b \right) \geq 1 - \xi_i \tag{2.34}$$

ここで使われるパラメータ C の値は分類の誤りに対するペナルティを表しており, C の値が小さいほど誤りが許され, 逆に C の値が大きくなるほど誤りは許されず, ハードマージン SVM に近づく. このパラメータ C の値を最適化する手法は現在のところ確立されておらず, 経験的に決定されることが多い.

以上のようにして最適化問題に対してスラック変数を導入することで, ラグランジュの未定乗数法により書き換えた最適化問題の式は次のように変更される.

$$\min_{\boldsymbol{\alpha}} \ -\sum_i \alpha_i + \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j \tag{2.35}$$

$$\text{subject to } \sum_i \alpha_i y_i = 0, \ \forall_i, 0 \leq \alpha_i \leq C \tag{2.36}$$

ソフトマージン SVM においては, $\alpha_i > 0$ を満たす全てのサンプル $\boldsymbol{x}_i$ がサポートベクターと呼ばれている. サポートベクターが求まった後は, ハードマージン SVM の

場合と同様にして式.2.31, 2.32 から重みベクトル $\boldsymbol{w}$ と定数 b の値が決定される．ソフトマージン SVM の模式図を図.2.11 に示す.

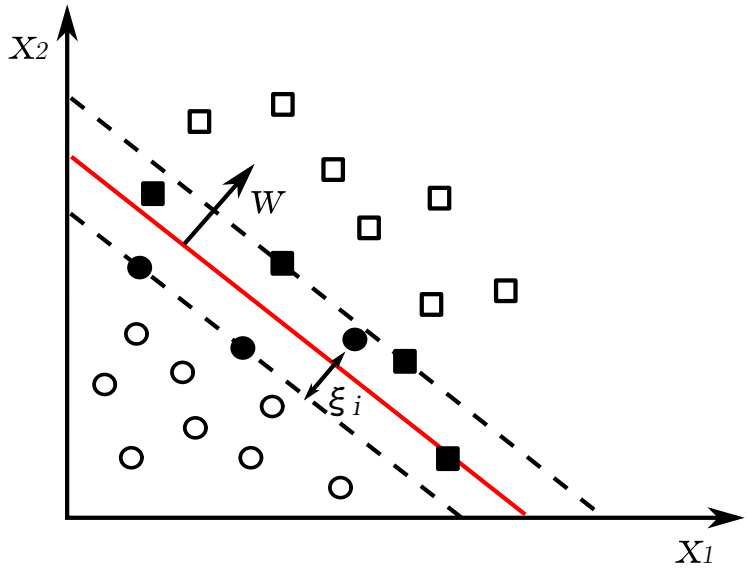

図 2.11: ソフトマージン SVM の模式図

2.3.3　非線形カーネル SVM

ソフトマージン SVM を用いることで解ける問題の数はある程度増えるが, 現実問題の中には線形分離だけでは解決出来ない問題も存在する．そのような問題を解くためには, より表現力のある非線形分類が必要となる．SVM においては非線形分類を行う際に使われる手法として, 非線形カーネル SVM が提案されている [19]．この非線形カーネル SVM ではカーネルトリック [20] を用いることで, 計算量を抑えつつ非線形分類を可能にしている.

非線形カーネル SVM では, 線形分類が不可能なデータ群を非線形写像を用いてより高次元の, データ群が線形分離可能となる特徴空間に写像する．非線形写像を行った後の新しい特徴空間においては, 線形分類が可能となっており, そのため写像後の特徴空間においてこれまでに説明してきた線形分類用の SVM をそのまま適用することが出来る．このように非線形写像を行った後に線形分類を行うことで, 実質的に元の特徴空間で非線形分類を行った場合と同等の分類を行うことが可能となる.

ここで M 次元の特徴ベクトル $\boldsymbol{x}$ を N 次元の特徴ベクトルに写す写像を $\phi(\cdot)$ と表

すこととする. このとき, 写像を行う前と行った後の特徴ベクトルの要素は, それぞれ次のように表すことが出来る.

$$\boldsymbol{x} = \left(x^{(1)}, x^{(2)}, ..., x^{(M)}\right) \tag{2.37}$$

$$\phi(\boldsymbol{x}) = \left(\phi^{(1)}(\boldsymbol{x}), \phi^{(2)}(\boldsymbol{x}), ..., \phi^{(N)}(\boldsymbol{x})\right) \tag{2.38}$$

更に, 二つの M 次元のデータ $\boldsymbol{x}$, $\boldsymbol{z}$ の写像の内積を $K(\boldsymbol{x}, \boldsymbol{z})$ とおく.

$$K(\boldsymbol{x}, \boldsymbol{z}) = \phi(\boldsymbol{x})^T \phi(\boldsymbol{z}) \tag{2.39}$$

SVM においては, 二つのデータの写像の内積 $K(\boldsymbol{x}, \boldsymbol{z})$ のことをカーネル関数と呼んでいる. 高次元空間における線形カーネル SVM はこれまで説明した式を用いて表現することが可能であるため, 非線形カーネル SVM における分離超平面の方程式はカーネル関数を用いて, 次のように書くことが出来る.

$$\sum_{\boldsymbol{x}_i \in SV} \alpha_i^* y_i \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}) + b = 0 \tag{2.40}$$

$$\Leftrightarrow \sum_{\boldsymbol{x}_i \in SV} \alpha_i^* y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b = 0 \tag{2.41}$$

この式から分かる通り, カーネル関数を用いることによって実際にデータ $\boldsymbol{x}$ の非線形写像 $\phi(\boldsymbol{x})$ を求めることなく, 非線形写像を行った分類が可能となる. そのため非線形カーネル SVM では, 写像を求めるための計算量を削減することが出来, 高速な非線形分類を可能としている.

このようにして計算量を抑える仕組みはカーネルトリックと呼ばれていおり, SVM ではカーネルトリックを用いることが出来るため, 今日では様々な分類問題に対して SVM が広く使われている. 非線形カーネル SVM の模式図を図.2.12 に示す.

どのような非線形写像を行うかはカーネル関数に委ねられており, カーネル関数を調節することで様々な非線形分類を行うことが出来る. ここではその中でも広く使われているカーネル関数をいくつか紹介する.

- 多項式カーネル

$$K(\boldsymbol{x}, \boldsymbol{z}) = \left(\gamma \boldsymbol{x}^T \boldsymbol{z} + r\right)^d \quad (\gamma > 0) \tag{2.42}$$

- Radial Basis Function (RBF) カーネル

$$K(\boldsymbol{x}, \boldsymbol{z}) = \left(-\gamma \left\|\boldsymbol{x} - \boldsymbol{z}\right\|^2\right) \quad (\gamma > 0) \tag{2.43}$$

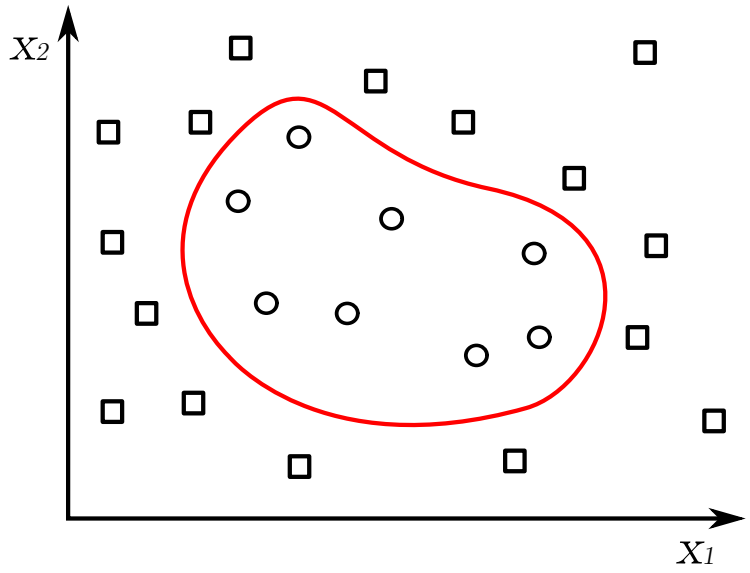

図 2.12: 非線形カーネル SVM の模式図

- シグモイドカーネル

$$K(\boldsymbol{x}, \boldsymbol{z}) \quad = \quad \tanh\left(\gamma \boldsymbol{x}^T \boldsymbol{z} + r\right)^d \tag{2.44}$$

上述のカーネル関数の中に現れるパラメータ γ, r, d は, ソフトマージン SVM で述べたパラメータ C と同様に経験的に値を決定するものであるが, 多くの論文では SVM のパラメータを交差検定を用いたグリッドサーチで求めている. SVM のパラメータは分類の精度に対して, カーネル関数と同じ程度の影響力を持っており, 正しく設定しなければ良い特徴量を使った分類を行っても精度が出ないことが知られている. 本稿では SVM のパラメータに関しては, 経験的にに決定することとした.

2.3.4　One-against-All 法

SVM は 2 クラス分類を行う分類器であり, 多クラスの分類には対応していない. そこで, SVM を多クラスの分類にも対応出来るようにした手法がいくつか提案されている. 本研究では, その中でも比較的原理・実装が簡単な, One-against-All 法 [21] を用いた.

これは, 分類したいクラスの数だけ分類器を用意し, ある一つのクラスに属するか属さないかを各クラスごとに分類器に学習させるものである. このとき, あるクラス

に属する場合は 1, そうでない場合は -1 のラベルを取ることにする. この方法により学習させた分類器を用いてあるデータを分類したとき, ある一つのクラスに属する結果が出れば問題は無いが, 複数のクラスに属したり, 逆にどのクラスにも属さない可能性も考えられる. そのような場合にも対応するために, 各クラスの識別関数の値が最小となるクラスを, そのデータが属するクラスとする.

ここで j 番目のクラスに関する分類器のパラメータを $\boldsymbol{w}_j$, b_j とするとき, データ $\boldsymbol{x}$ が属するクラス $\hat{j}$ は次の式より求まる.

$$\hat{j} \quad = \quad \underset{j}{\mathrm{argmax}} \ \boldsymbol{w}_j^T \boldsymbol{x} + b_j \tag{2.45}$$

第3章　提案手法

本章では提案手法である, 音声の基底の抽出に適した制約を付けたCNMFの目的関数, 目的関数を元にAFとCDを使って導出したCNMFのパラメータの更新式, 抽出した音声の基底を用いてSVMによる音素分類を行う際に必要となる重み行列の変換方法について説明する.

3.1　制約付き目的関数

NMFを用いてパターン抽出を行う際には, 目的関数としてフロベニウス距離かKL-Divergenceのどちらかが使われることが多い. 本研究においては更新式の導出の際にCoordinae-Descentを用いており, この手法はフロベニウス距離の目的関数から導出することが多いため, 本研究においても目的関数としてフロベニウス距離(式.2.9)を用いることとした.

しかし, この目的関数を単純に最小化するだけでは最適解が一意に定まらない. そのために何かしらの制約を目的関数に付加し, 最適解を一意的に定める必要がある. これ以後では, 音声の基底を抽出する目的に適した制約について論じることとする.

単一話者の連続音声中には, 各時刻に存在する音声の基底の数は高々数個と考えられ, 多くの基底が同一時刻において存在するとは考えにくい. そこで, スパースな形で連続音声から基底を抽出するために, CNMFの重み行列にL-1正則化をかける. L-1正則化とは, その要素の総和を目的関数に付加することを指し, これによって重み行列は大半の要素が0となる行列に分解される. 重み行列は各時刻における基底の重み(存在の度合い)を示しており, 大半の値が0となるとき同時刻に存在する基底の数は少ないことを意味している. このような, 多くの要素が0の値を取る行列はスパース行列と呼ばれている. 重み行列のL-1正則化を付加した目的関数を式.3.1に示す.

$$D(\boldsymbol{V}|\boldsymbol{A}) \quad = \quad \frac{1}{2}\sum_{i,k}\left(v_{ik}-a_{ik}\right)^2+\lambda_h\sum_{j,k}h_{jk} \tag{3.1}$$

ここで λ_h は正則化のパラメータであり問題に合わせて決める必要があるが, このパラメータも通常は経験的に決定される.

重み行列に L-1 正則化をつけることにより, スパースな形で連続音声から基底を抽出することが出来るが, この目的関数を単純に最小化すると重み行列の要素は全て 0 に近づき, 基底行列の要素は無限大に発散することが考えられる. そこで基底行列には L-2 正則化を掛け, 基底行列の要素をある程度の値に抑えるようにする. L-2 正則化とは, その要素の二乗和を目的関数に付加することを指し, これによって基底行列の大きさはあるところで収束することになる. L-2 正則化を基底行列にかけた目的関数を式.3.2 に示す. ここで w_{ijt} は t 番目の基底行列 $\boldsymbol{W}_t$ の第 i 行 · 第 j 列の成分を表す.

$$D(\boldsymbol{V}|\boldsymbol{A}) = \frac{1}{2}\sum_{i,k}\left(v_{ik}-a_{ik}\right)^2 + \lambda_h \sum_{j,k} h_{jk} + \frac{1}{2}\lambda_w \sum_{i,j,t} w_{ijt}^2 \tag{3.2}$$

この式に現れるパラメータ λ_w もまた問題に合わせて決定する必要がある. パラメータ λ_w と λ_h の値については, 本研究では議論されておらず経験的に決定しているが, これに関する解析も必要であると考える.

3.2　補助関数法による更新式

ここでは, 補助関数法 (AF 法) とイェンゼンの不等式を用いて CNMF の目的関数から導出する方法を説明する. 以後, この更新式のことは AF 更新式と呼ぶことにする.

まず提案した目的関数 $D(\boldsymbol{V}|\boldsymbol{A})$ に新しくパラメータ δ_{ijkt} を導入する. ただし δ_{ijkt} は次の式を満たすものとする.

$$\forall i,\ \forall k,\ \sum_{j,t}\delta_{ijkt} = 1 \tag{3.3}$$

$$\forall i,\ \forall j,\ \forall k, \forall t,\ 0 \leq \delta_{ijkt} \leq 1 \tag{3.4}$$

また要素 $(h_{jk})^{t\rightarrow}$ は行列 $(\boldsymbol{H})^{t\rightarrow}$ の第 j 行, 第 k 列の要素を表すこととする. この時目的関数は次のように変形する事が出来る.

$$\begin{aligned} D(\boldsymbol{V}|\boldsymbol{A}) &= \frac{1}{2}\sum_{i,k}\left(v_{ik}-a_{ik}\right)^2 + \lambda_h \sum_{j,k} h_{jk} + \frac{1}{2}\lambda_w \sum_{i,j,t} w_{ijt}^2 \\ &= \frac{1}{2}\sum_{i,k}\left\{ v_{ik}^2 - 2v_{ik}a_{ik} + \left(\sum_{j,t} w_{ijt}\left(h_{jk}\right)^{t\rightarrow}\right)^2 \right\} \end{aligned}$$

$$
\begin{aligned}
&+\lambda_h \sum_{j,k} h_{jk} + \frac{1}{2}\lambda_w \sum_{i,j,t} w_{ijt}^2 \\
= \ & \frac{1}{2}\sum_{i,k}\left\{ v_{ik}^2 - 2v_{ik}a_{ik} + \left(\sum_{j,t} \delta_{ijkt} \frac{w_{ijt}\left(h_{jk}\right)^{t\rightarrow}}{\delta_{ijkt}} \right)^2 \right\} \\
&+\lambda_h \sum_{j,k} h_{jk} + \frac{1}{2}\lambda_w \sum_{i,j,t} w_{ijt}^2 \qquad (3.5)
\end{aligned}
$$

イェンゼンの不等式により, 式.3.5 中に存在する項 $\left(\sum_{j,t} \delta_{ijkt} \frac{w_{ijt}\left(h_{jk}\right)^{t\rightarrow}}{\delta_{ijkt}} \right)^2$ は次の不等式を満たす.

$$
\begin{aligned}
\left(\sum_{j,t} \delta_{ijkt} \frac{w_{ijt}\left(h_{jk}\right)^{t\rightarrow}}{\delta_{ijkt}} \right)^2 \ &\leq \ \sum_{j,t} \delta_{ijkt} \left(\frac{w_{ijt}\left(h_{jk}\right)^{t\rightarrow}}{\delta_{ijkt}} \right)^2 \\
&= \ \sum_{j,t} \frac{w_{ijt}^2 \left\{ \left(h_{jk}\right)^{t\rightarrow} \right\}^2}{\delta_{ijkt}} \qquad (3.6)
\end{aligned}
$$

この不等式は次の条件を満たすときに等号が成立する.

$$
\begin{aligned}
\delta_{ijkt} \ &= \ \frac{w_{ijt}\left(h_{jk}\right)^{t\rightarrow}}{\sum_{j',t'} w_{ij't'}\left(h_{j'k}\right)^{t'\rightarrow}} \\
&= \ \frac{w_{ijt}\left(h_{jk}\right)^{t\rightarrow}}{a_{ik}} \qquad (3.7)
\end{aligned}
$$

次に式.3.6 を式.3.5 に代入して,

$$
\begin{aligned}
D(\boldsymbol{V}|\boldsymbol{A}) \ \leq \ & \frac{1}{2}\sum_{i,k}\left(v_{ik}^2 - 2v_{ik}a_{ik} + \sum_{j,t} \frac{w_{ijt}^2 \left\{ \left(h_{jk}\right)^{t\rightarrow} \right\}^2}{\delta_{ijkt}} \right) \\
&+\lambda_h \sum_{j,k} h_{jk} + \frac{1}{2}\lambda_w \sum_{i,j,t} w_{ijt}^2 \qquad (3.8)
\end{aligned}
$$

と変形できる. ここで式.3.8 の右辺を $D(\boldsymbol{V}|\boldsymbol{A})$ の補助関数 $C(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{\Delta})$ とおく. ただし $\boldsymbol{\Delta}$ は要素 δ_{ijkt} の集合を表す.

次に補助関数法を用いて行列 $\boldsymbol{W}$ と $\boldsymbol{H}$ を更新するため, $C(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{\Delta})$ を要素 $w_{ijt},\ h_{jk}$ で偏微分した式を考える.

$$\frac{\partial C(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{\Delta})}{\partial w_{ijt}} = -\sum_{k'} v_{ik'} \left(h_{jk'}\right)^{t\rightarrow} + w_{ijt} \sum_{k'} \frac{\left\{\left(h_{jk'}\right)^{t\rightarrow}\right\}^2}{\delta_{ijk't}} + \lambda_w w_{ijt} \tag{3.9}$$

$$\frac{\partial C(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{\Delta})}{\partial h_{jk}} = -\sum_{i',t'} w_{i'jt'} \left(v_{i'k}\right)^{\leftarrow t'} + h_{jk} \sum_{i',t'} \frac{w_{i'jt'}^2}{\left(\delta_{i'jkt'}\right)^{\leftarrow t'}} + \lambda_h \tag{3.10}$$

ここで $\left(\delta_{i'jkt'}\right)^{\leftarrow t'}$ は次式のものとする.

$$\left(\delta_{i'jkt'}\right)^{\leftarrow t'} = \frac{w_{i'jt'} h_{jk}}{\left(a_{i'k}\right)^{\leftarrow t'}} \tag{3.11}$$

更に, 上で求めた偏微分がそれぞれ 0 となるときの $w_{ijt},\ h_{jk}$ を求める.

$$\begin{aligned} \frac{\partial C(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{\Delta})}{\partial w_{ijt}} &= 0 \\ \Leftrightarrow\ w_{ijt} &= \frac{\displaystyle\sum_{k'} v_{ik'} \left(h_{jk'}\right)^{t\rightarrow}}{\displaystyle\sum_{k'} \frac{\left\{\left(h_{jk'}\right)^{t\rightarrow}\right\}^2}{\delta_{ijk't}} + \lambda_w} \end{aligned} \tag{3.12}$$

$$\begin{aligned} \frac{\partial C(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{\Delta})}{\partial h_{jk}} &= 0 \\ \Leftrightarrow\ h_{jk} &= \frac{\displaystyle\sum_{i',t'} w_{i'jt'} \left(v_{i'k}\right)^{\leftarrow t'} - \lambda_h}{\displaystyle\sum_{i',t'} \frac{w_{i'jt'}^2}{\left(\delta_{i'jkt'}\right)^{\leftarrow t'}}} \end{aligned} \tag{3.13}$$

最後に, 式.3.7 を式.3.12 と式.3.13 に代入する事で, $w_{ijt},\ j_{jk}$ の更新式を求める事が出来る.

$$w_{ijt} \leftarrow w_{ijt} \frac{\displaystyle\sum_{k'} v_{ik'} \left(h_{jk'}\right)^{t\rightarrow}}{\displaystyle\sum_{k'} a_{ik'} \left(h_{jk'}\right)^{t\rightarrow} + \lambda_w w_{ijt}} \tag{3.14}$$

$$h_{jk} \leftarrow h_{jk} \frac{\sum_{i',t'} w_{i'jt'} \left(v_{i'k}\right)^{\leftarrow t'} - \lambda_h}{\sum_{i',t'} w_{i'jt'} \left(a_{i'k}\right)^{\leftarrow t'}} \tag{3.15}$$

非負値行列分解の制約として, 全ての要素は必ず正になる必要がある. しかし, 式.3.16 を用いた更新では h_{jk} が負の値に更新される可能性がある. そこで, 式.3.16 を次の式に書き換える.

$$h_{jk} \leftarrow h_{jk} \frac{\left[\sum_{i',t'} w_{i'jt'} \left(v_{i'k}\right)^{\leftarrow t'} - \lambda_h\right]_+}{\sum_{i',t'} w_{i'jt'} \left(a_{i'k}\right)^{\leftarrow t'}} \tag{3.16}$$

ここで $[\cdot]$ は, 0 より大きい実数はそのまま出力し, 0 以下の実数は 0 に近い非常に小さなある正数に変換する非線形変換を表す. これにより, h_{jk} の値は負の値に更新される前に止められる. 補助関数 $C(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{\Delta})$ は h_{jk} に関して二次関数となっており, そのために凸関数であるので, このように更新される要素の値を途中で止めても必ず目的関数が小さくなる方向に更新される.

以上の更新式を収束するまで繰り返す事で, 目的関数を最適化する行列 $\boldsymbol{W}$ と $\boldsymbol{H}$ を求める事が出来る.

3.3　Coordinate Descent による更新式

ここでは Coordinate Descent を用いて CNMF の目的関数から導出する更新式について説明する. 以後, この更新式は CD 更新式と呼ぶことにする.

まず重み行列のある一つの要素 w_{abc} を $\hat{w}_{abc} = w_{abc} + \alpha_{abc}$ に更新することを考える. この時更新後の重み行列を $\hat{\boldsymbol{W}}$, 近似行列を $\hat{\boldsymbol{A}}$ とすると目的関数の次式のように書ける.

$$\begin{aligned} D(\boldsymbol{V}|\hat{\boldsymbol{A}}) &= \frac{1}{2}\sum_{i,k}\left(v_{ik} - \hat{a}_{ik}\right)^2 + \lambda_h \sum_{j,k} h_{jk} + \frac{1}{2}\lambda_w \sum_{i,j,t} \hat{w}_{ijt}^2 \\ &= \frac{1}{2}\sum_{i,k}\left\{v_{ik} - \sum_{j,t} \hat{w}_{ijt} \left(h_{jk}\right)^{t\rightarrow}\right\}^2 + \lambda_h \sum_{j,k} h_{jk} + \frac{1}{2}\lambda_w \sum_{i,j,t} \hat{w}_{ijt}^2 \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2}\sum_{i,k}\left\{v_{ik} - \sum_{j,t} w_{ijt}\left(h_{jk}\right)^{t\rightarrow}\right\}^2 + \lambda_h \sum_{j,k} h_{jk} + \frac{1}{2}\lambda_w \sum_{i,j,t} w_{ijt}^2 \\
&\quad + \sum_{k} \alpha_{abc}\left(a_{ak} - v_{ak}\right)\left(h_{bk}\right)^{c\rightarrow} + \sum_{k} \frac{1}{2}\alpha_{abc}^2 \left\{\left(h_{bk}\right)^{c\rightarrow}\right\}^2 \\
&\quad + \lambda_w \alpha_{abc} w_{abc} + \frac{1}{2}\lambda_w \alpha_{abc}^2
\end{aligned} \tag{3.17}
$$

この目的関数を最小化するパラメータ α_{abc} は次の等式を満たす.

$$
\frac{\partial D(\boldsymbol{V}|\hat{\boldsymbol{A}})}{\partial \alpha_{abc}} = 0 \tag{3.18}
$$

式.3.18 を解いて最適化する α_{abc} を求める.

$$
\begin{aligned}
\frac{\partial D(\boldsymbol{V}|\hat{\boldsymbol{A}})}{\partial \alpha_{abc}} &= \sum_{k}\left(a_{ak} - v_{ak}\right)\left(h_{bk}\right)^{c\rightarrow} + \sum_{k}\alpha_{abc}\left\{\left(h_{bk}\right)^{c\rightarrow}\right\}^2 + \lambda_w w_{abc} + \lambda_w \alpha_{abc} \\
&= 0 \\
\Leftrightarrow\ \alpha_{abc} &= -\frac{\sum_{k}\left(a_{ak} - v_{ak}\right)\left(h_{bk}\right)^{c\rightarrow} + \lambda_w w_{abc}}{\sum_{k}\left\{\left(h_{bk}\right)^{c\rightarrow}\right\}^2 + \lambda_w}
\end{aligned} \tag{3.19}
$$

求まった α_{abc} で w_{abc} を更新する際に更新後の値が負になる可能性がある. しかし CNMF の制約で全ての行列の要素は非負値に制限されている. そこで求まった α_{abc} を使って次の式により更新する.

$$
\hat{w}_{abc} = \max\left(w_{abc} + \alpha_{abc}, 0\right) \tag{3.20}
$$

式.3.17 は α_{abc} に関して二次関数になっているため, 負値になる前に 0 で止めても目的関数は必ず小さくなる方向に更新される. よって式.3.20 で重み行列を更新する際に必ず目的関数が減少することが保証されている. (図.3.1)

次に重み行列のある一つの要素 h_{de} を $\hat{h}_{de} = h_{de} + \beta_{de}$ に更新することを考える. この時更新後の重み行列を $\hat{\boldsymbol{W}}$, 近似行列を $\hat{\boldsymbol{A}}$ とすると目的関数の次式のように書ける.

$$
\begin{aligned}
D(\boldsymbol{V}|\hat{\boldsymbol{A}}) &= \frac{1}{2}\sum_{i,k}\left(v_{ik} - \hat{a}_{ik}\right)^2 + \lambda_h \sum_{j,k}\hat{h}_{jk} + \frac{1}{2}\lambda_w \sum_{i,j,t} w_{ijt}^2 \\
&= \frac{1}{2}\sum_{i,k}\left\{v_{ik} - \sum_{j,t} w_{ijt}\left(\hat{h}_{jk}\right)^{t\rightarrow}\right\}^2 + \lambda_h \sum_{j,k}\hat{h}_{jk} + \frac{1}{2}\lambda_w \sum_{i,j,t} w_{ijt}^2
\end{aligned}
$$

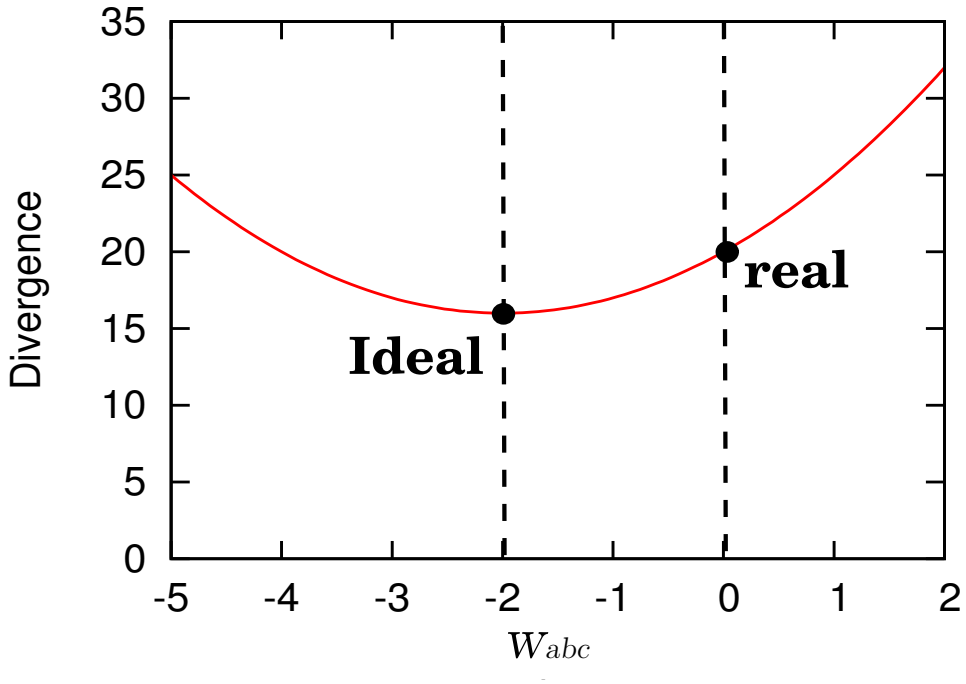

図 3.1: CD による負値への更新

$$
\begin{aligned}
= \quad & \frac{1}{2}\sum_{i,k}\left\{v_{ik} - \sum_{j,t} w_{ijt}\left(h_{jk}\right)^{t\rightarrow}\right\}^2 + \lambda_h \sum_{j,k} h_{jk} + \frac{1}{2}\lambda_w \sum_{i,j,t} w_{ijt}^2 \\
& + \sum_{i,t} \beta_{de} w_{idt}\left\{\left(a_{ie}\right)^{\leftarrow t} - \left(v_{ie}\right)^{\leftarrow t}\right\} \\
& + \sum_{i,t} \frac{1}{2}\beta_{de}^2 w_{idt}^2 \left(\mathbf{1}\right)^{\leftarrow t} + \lambda_h \beta_{de}
\end{aligned} \tag{3.21}
$$

ここで行列 $\mathbf{1}$ は入力行列と同等の行数, 列数で全ての要素が 1 である行列とする.

この目的関数を最小化するパラメータ β_{de} は次の等式を満たす.

$$
\frac{\partial D(\boldsymbol{V}|\hat{\boldsymbol{A}})}{\partial \beta_{de}} = 0 \tag{3.22}
$$

式.3.22 を解いて最適化する β_{de} を求める.

$$\begin{aligned}\frac{\partial D(\boldsymbol{V}|\hat{\boldsymbol{A}})}{\partial \beta_{de}} &= \sum_{i,t} \beta_{de} w_{idt} \left\{ (a_{ie})^{\leftarrow t} - (v_{ie})^{\leftarrow t} \right\} + \sum_{i,t} \frac{1}{2} \beta_{de}^2 w_{idt}^2 (\mathbf{1})^{\leftarrow t} + \lambda_h \beta_{de} \\ &= 0 \\ \Leftrightarrow \ \beta_{de} &= -\frac{\sum_{i,t} w_{idt} \left\{ (a_{ie})^{\leftarrow t} - (v_{ie})^{\leftarrow t} \right\} + \lambda_h}{\sum_{i,t} w_{idt}^2 (\mathbf{1})^{\leftarrow t}} \end{aligned} \tag{3.23}$$

求まった β_{de} で h_{de} を更新する際, パターン行列の時と同様に更新後の値が負になる可能性がある. そこで求まった β_{de} を使って次の式により更新する.

$$\hat{h}_{de} = \max(h_{de} + \beta_{de}, 0) \tag{3.24}$$

3.4　音素分類のための重み行列の変換

CNMF によって抽出した音声パターンを用いて SVM による音素分類を行う際, 音声の各時刻における特徴量は各パターンへの重みとすることが望ましいと考えられる. そのため重み行列を SVM の入力行列とすると音素分類が可能になる. しかし CNMF において重み行列が表す重みは, パターンが生起する時刻を表しておりパターンが存在する時刻は表していない. 音素分類を行う他の手法においては, Max-pooling [22] が使われており, 本研究においてもそれを参考にした変換を提案する.

重み行列 $\boldsymbol{H}$ を変換した後の行列を $\boldsymbol{H}^{\mathrm{svm}}$ とする. このとき, 変換後の行列の要素を次のようにする. ただし抽出した音声パターンの長さを T とする.

$$h_{jk}^{\mathrm{svm}} = \max\left\{ h_{jk}, (h_{jk})^{1\rightarrow}, \ \ldots, (h_{jk})^{(T-1)\rightarrow} \right\} \tag{3.25}$$

この重み行列の変換の例を図.3.2 に示す.

$$\boldsymbol{V} = \begin{pmatrix} 2 & 1 & 0 & 2 & 1 \\ 2 & 1 & 0 & 1 & 2 \\ 1 & 2 & 0 & 1 & 2 \\ 1 & 2 & 0 & 2 & 1 \end{pmatrix}$$

$$\boldsymbol{H} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \longrightarrow \boldsymbol{H}^{\mathrm{svm}} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

図 3.2: 重み行列の変換

第4章　実験と評価

4.1　音素分類実験の概要

本稿では連続音声を用いた音素分類実験により, 提案手法の評価を行った. これ以降では実験に使われる TIMIT Corpus, メルスペクトログラムへの変換方法, 実験の流れをそれぞれ説明する.

4.1.1　TIMIT Acoustic-Phonetic Continuous Speech Corpus

TIMIT Acoustic-Phonetic Continuous Speech Corpus (TIMIT Corpus) [23] は音素分類実験に広く使われているコーパスである. このコーパスには複数の話者の様々な発話が収録されており, 各発話には音素 (Phone) のラベルが付けられている. 音素分類を行う際は [6] に示されている慣例に従うものとする.

まず決められた連続音声 3696 発話を用いて音声パターンを教師無し学習する. 次に core_test と言われている音声データ 192 発話を訓練データと評価データの二つに分け, SVM による音素分類実験を行う. 音声パターンの教師無し学習においては計算量の問題により, 実験に応じて 3696 発話全て使わずにその内のある程度の発話を使うものとする.

TIMIT Corpus に元々付けられている音素ラベルは 64 種類存在するが, その音響的類似によってそれらを 39 種類に置き換えて音素分類の実験を行う[1]. 音素ラベルの対応関係 4.1 を表に示す.

4.1.2　音素分類実験の流れ

実験の大まかな流れを図.4.1 に示す. 初めに連続音声を FFT で周波数情報に変換し, 更にメルフィルタバンクを用いてメルスペクトラムまで変換する. 次に式.3.20, 式.3.24 を用いて音声パターン抽出用の連続音声からパターン行列 $\boldsymbol{W}$ を求める. その後求めたパターン行列で固定したまま, SVM 分類用の訓練データ, 評価データに

[1]慣例によりラベル”q”となる部分は音素分類を行う際に取り除く.

表 4.1: 各音素ラベルの振り分け

音素ラベル	番号	音素ラベル	番号
iy	1	eng	21
ih	2	dx	22
ix	2	jh	23
eh	3	ch	24
ae	4	z	25
ah	5	s	26
ax	5	sh	27
ax-h	5	zh	27
uw	6	hh	28
ux	6	hv	28
uh	7	v	29
aa	8	f	30
ao	8	dh	31
ey	9	th	32
ay	10	b	33
oy	11	p	34
aw	12	d	35
ow	13	t	36
er	14	g	37
axr	14	k	38
l	15	bcl	39
el	15	pcl	39
r	16	dcl	39
w	17	tcl	39
y	18	gcl	39
m	19	kcl	39
em	19	epi	39
n	20	pau	39
en	20	h#	39
nx	20	CL	39
ng	21	q	-

対して式.3.24 を用いて重み行列 $\boldsymbol{H}$ を求める．最後に重み行列を SVM で変換して $\boldsymbol{H}^{\mathrm{svm}}$ を得た後, SVM により訓練 · 評価を行いその精度を求める．SVM のカーネル

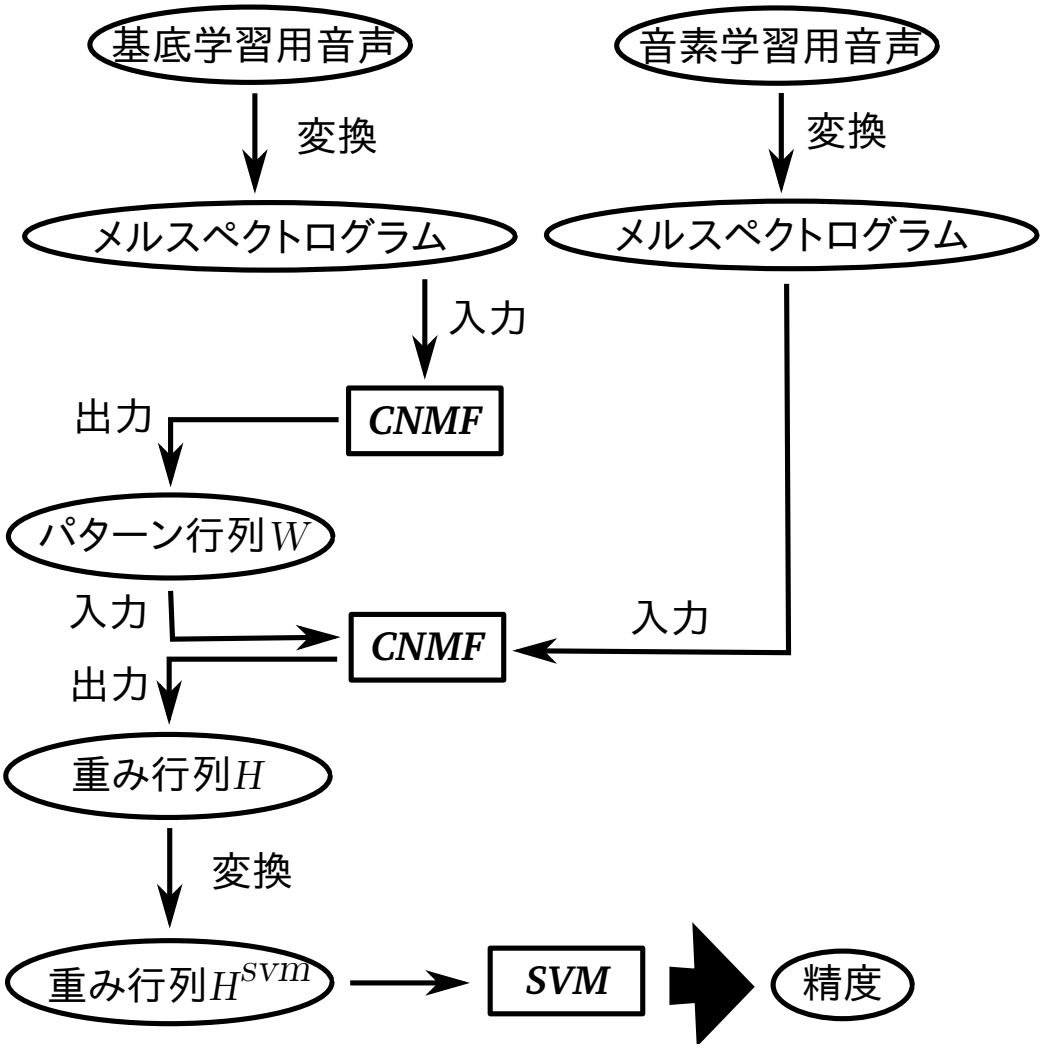

図 4.1: 音素分類実験の流れ

関数では RBF カーネルを使用し, SVM の精度はは 6 分割交差検定で行った.

本実験において最終的に得られた SVM の精度が良い場合, CNMF により抽出した音声パターンが音韻的情報を抽出していると考えられる. よって本実験では SVM の精度を向上させることを目標に様々なパラメータを決定する.

表 4.2: メルスペクトラムのパラメータ

サンプリング周波数	16 [kHz]
プリエンファシス	0.97
窓関数	ハミング窓
フレームサイズ	16 [msec]
フレームシフト	8 [msec]
メルフィルタバンクの次数	40

4.1.3　連続音声の前処理

連続音声を振幅値情報からメルスペクトラムに変換する際に使用したパラメータを表.4.2 に示す. 今回の実験では, まず連続音声に対してプリエンファシスという前処理を行った. 具体的には n 番目の元のサンプル値を x_n, プリエンファシス後のサンプル値を y_n とするとき, 次式の変換を行うことである.

$$y_n \quad = \quad x_n - px_{n-1} \tag{4.1}$$

本実験では他の音声の研究でも行われているように, $p = 0.97$ としてプリエンファシスを行った. その後連続音声をフレームごとに切り出し, ハミング窓を掛けた後に FFT により周波数領域に変換した. このとき FFT のフレームサイズを 256 点 (= 16[msec]) とし, フレームシフトは 128 点 (= 8[msec]) とした. 最後に, 得られた FFT の値を 40 次のメルフィルタバンクによりメル尺度に変換して, メルスペクトログラムを得た.

4.2　AF 更新式と CD 更新式の比較

まず, 最適な更新式を決定するために提案した二つの更新式の発散値の推移を比較した. この実験においてはより速くより小さな発散値に収束する更新式を良い更新式とする.

4.2.1　実験方法

更新式の優劣を調べるため, まず基底学習用の音声 1 発話を用いてパターン行列と重み行列の両方を各更新式を用いて更新した. 更に音素学習用の音声 1 発話を用いてパターン行列を固定したまま重み行列を更新したときの発散値の推移を調べた.

本実験においては, 抽出する基底の数は 10 と 20 の 2 通り, また基底の長さは 1, または 4 とした. 更に正則化項の値は, $(\lambda_w,\ \lambda_h) = (0,\ 0),\ (10^{-10},\ 10^{-10}),\ (1,\ 1)$ の 3 つの場合で実験を行った. またどちらの手法においても更新回数は, 10000 更新まで行った. ここで 1 更新とは, 全ての行列の要素が 1 回更新された状態の事を指す.

4.2.2　実験結果

基底学習用の音声の実験結果を図.4.2 に, 音素学習用の音声の実験結果を図.4.3 に示す. 図において PTN は抽出する基底の数を表し, TIM は基底の長さを表している.

まず基底学習用の音声による実験結果から考察する. 実験結果をみると, パラメータがどの値を取っても必ず CD 更新式の方が速く収束していることが確認できる. このため, NMF の同様に CNMF においても CD を用いて導出した更新式は有用であると考える事が出来る.

更に, 抽出する基底の数とその長さが大きくなると, CD 更新式による収束が AF 更新式に比べてより速くなることが確認できる. つまり CNMF のモデルのパラメータの次元が大きくなるほど, CD による収束の速さが際立つということである. 現在は学習に用いている音声の数は少ないが, 以降の実験では更に増えてくるため, CD 更新式の効果は更に顕著になると考えられる.

このように CD 更新式の方が収束が速くなる理由としては, AF 更新式は積形で表されているのに対して, CD 更新式は和形で表されている事が挙げられる. AF 更新式では, 更新式が積形で表されるために値が 0 に更新されることが無いが, 全てのパターンが全ての時刻に存在しているとは考えにくい. そのために積形では遅くなると考えられる. 対して CD 更新式では, 更新式が和形で表されているために値が 0 にたやすく更新される. この点が収束性の違いを生む原因の一つであると考えられる.

またそれ以外にも, CD 更新式では各パラメータを更新する際, 必ずそのパラメータに関して目的関数を最小にするように更新をしている. しかし, AF 更新式では補助関数を用いて他のパラメータを経由してから更新されるために, そのパラメータに関して必ずしも目的関数を最小化するようには更新されていない. この点も収束性の違いを生む原因と考えられる.

次に音素学習用の音声による実験結果について考察する. こちらの実験結果も, 先の基底学習用の結果と同様で CD 更新式の方がよい性能を示しているが, 基底学習用ほどの差異は 2 つの更新式には見られない. これは, 音素学習用の音声の更新では重み行列しか更新されていないためであり, パターン行列は固定されているために変動が小さく, どちらの更新式でもすぐに収束したと考えられる.

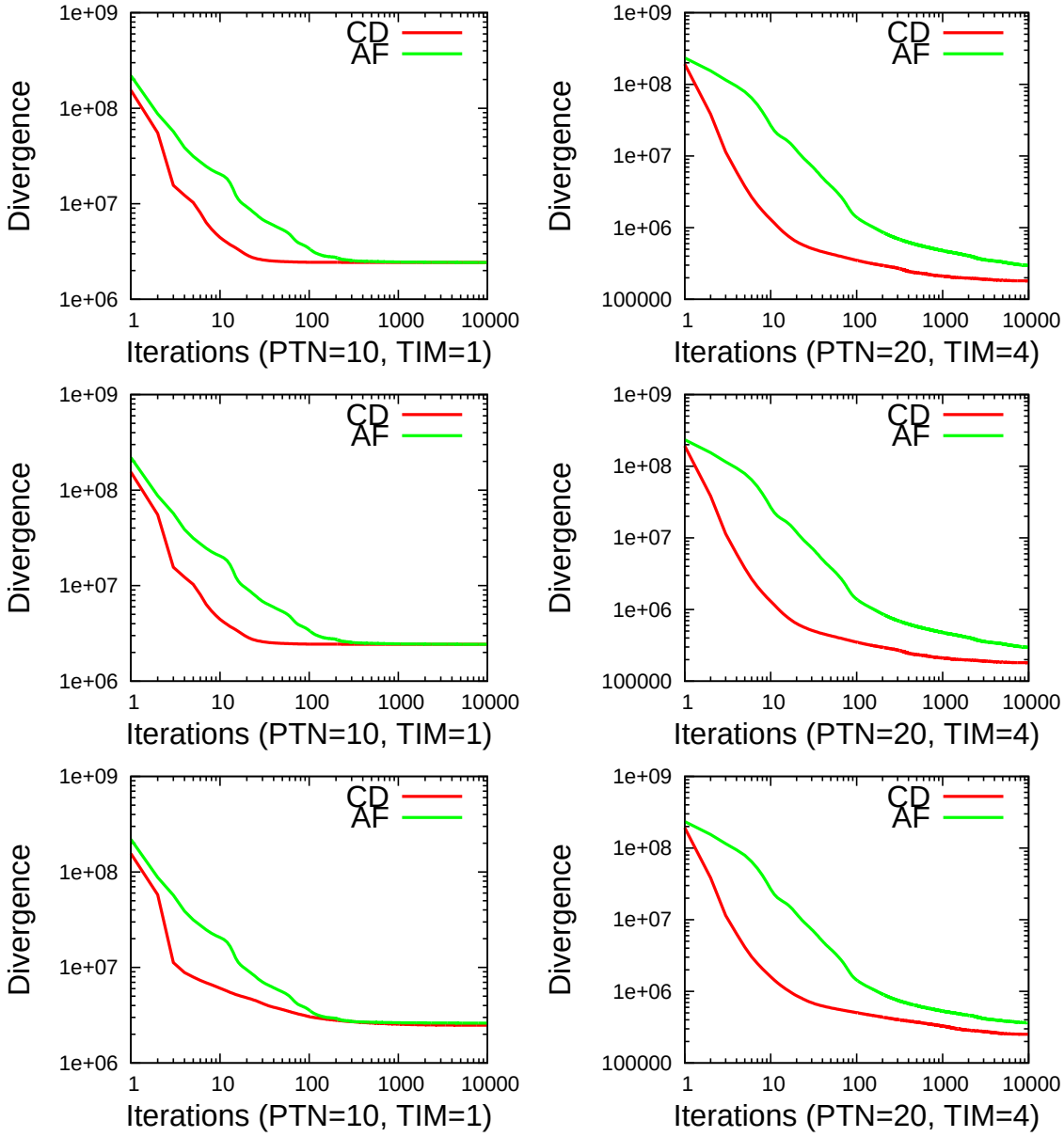

図 4.2: 基底学習用の発散値の推移 (上段:$\lambda_w = \lambda_h = 0$, 中段:$\lambda_w = \lambda_h = 10^{-10}$, 下段:$\lambda_w = \lambda_h = 1$)

またどちらの音声の実験においても, 正則化項の値によって収束の速さが大きく変わることはなかった. つまり, どちらの更新式においても正則化項の値で収束の速さが変化することは無いと確認する事が出来る.

以上より, 更新する行列の数や CNMF の各パラメータの値に依らず, 必ず CD 更

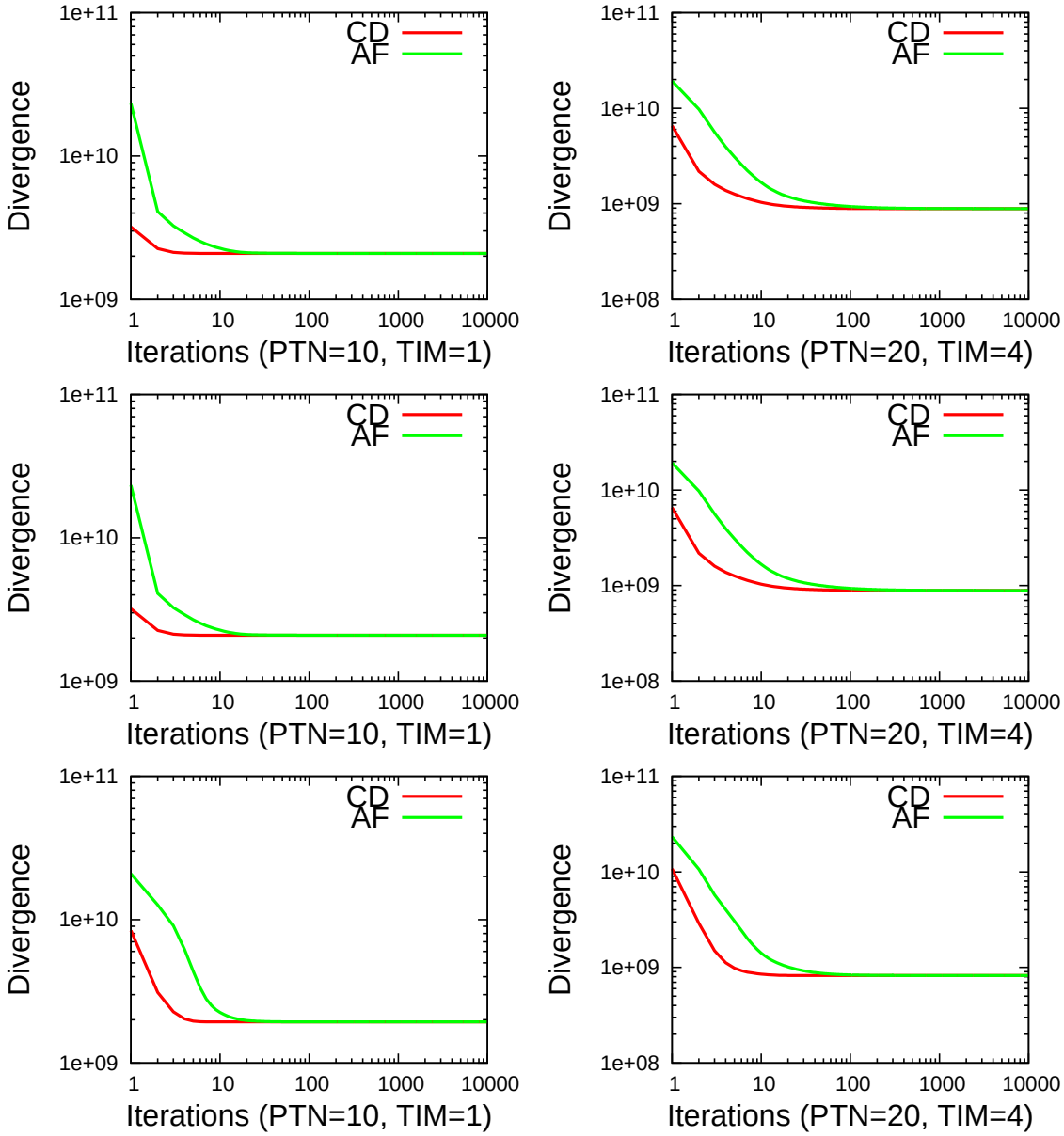

図 4.3: 音素学習用の発散値の推移 (上段:$\lambda_w = \lambda_h = 0$, 中段:$\lambda_w = \lambda_h = 10^{-10}$, 下段:$\lambda_w = \lambda_h = 1$)

新式の方が速く収束する事が示された.

4.3　パターン数とパターン長の設定

次に, CNMF のパターン数とパターン長を様々変えて, SVM の分類精度がどのように変化するかを実験で確かめた.

4.3.1　実験方法

SVM の分類精度の変化を調べるために, 基底学習用の音声 100 発話と音素学習用の音声 192 発話を使用した. 実験で使用した更新式は, 先の実験で性能が良かった CD 更新式を用いた. CNMF による基底の抽出数は 50, 100, 150, 200, 250, 300 の 6 通りで, 抽出する基底の長さは 1, 2, 3, 4, 5, 6 の 6 通りとした. また正則化項の値は $(\lambda_w,\ \lambda_h) = (0,\ 0),\ (1,\ 1)$ の 2 通りで行った. 更に SVM のパラメータを $C = 10^2,\ \gamma = (\text{パターン長}) \times 10^{-1}$ とした.

4.3.2　実験結果

実験結果を図.4.4 に示す. 横軸は基底の長さ, 縦軸は音素分類実験の精度を表す. また PTN は抽出する基底の数を表している. まず抽出する基底の数に注目して実験

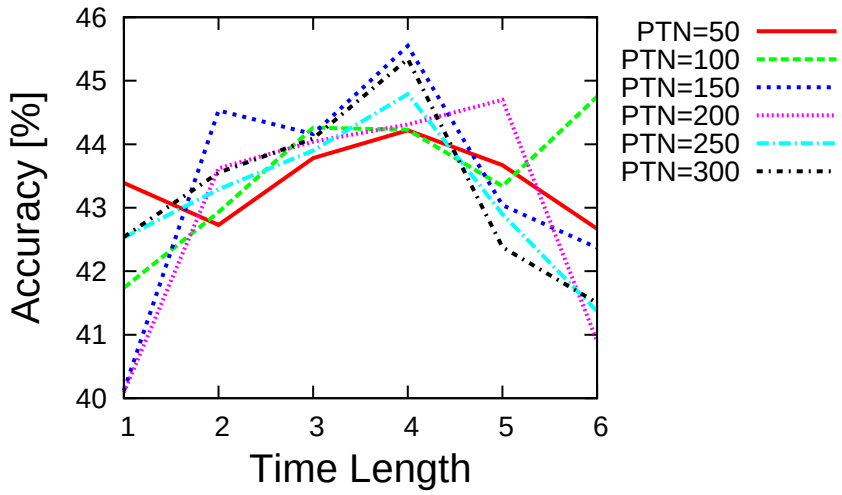

図 4.4: パターン数とパターン長の影響

結果を見る. 基底の数が 150 個になったところで精度が上昇していないことがグラフより確認できる. このため, 抽出するべき基底の数は 150 個であると推測される.

また基底の長さに注目すると, どの基底数においても, 長さが 4(= 32[msec]) となるときに精度が最大となる. これより, 抽出する基底の長さは 4 に設定する事が望ましいと考えられる.

この実験結果より, 音声に含まれている基底の数は高々150 個であることが推測される. 現在の音声認識では, 音素を表すトライフォンモデルが持つ状態の総数は 1000 個ほどと言われている. トライフォンモデルが持つ状態とは, CNMF の基底に相当するものと考えられれ, 本研究の結果よりトライフォンモデルが持つ状態数は更に削減する事が出来ると考えられる. 教師無し学習が持つ利点としては, このように知識ベースでは分類することが出来ないパターンも, 適切な形で分類出来るところである.

4.4　CNMF による音素分類実験

最後に, 先の実験で求めた適切なパターン数とパターン長を用いて音素分類実験を行い, 他の手法との比較を行った.

4.4.1　実験方法

提案した手法による SVM の音素分類の精度を調べるために, 基底学習用の音声 3696 発話と音素学習用の音声 192 発話を使用した. 実験では CD 更新式を使用し, CNMF により抽出する基底の数は 150 個, 基底の長さは 4 とした. また正則化項の値は $(\lambda_w,\ \lambda_h) = (1,\ 1)$ とし, SVM のパラメータをそれぞれ $C = 10^2,\ \gamma = 10^{-3}$ とした.

4.4.2　実験結果

実験で得られた精度を他の手法と共に表.4.3 に示す. メルスペクトログラムの精

表 4.3: 提案手法と他の手法との比較

メルスペクトログラム	37.5 %
提案手法	56.6%
CDBNs (H. Lee et al, 2009 [3])	64.4 %
MFCC (K. Shon & H. Lee, 2012 [24])	80.0 %
H-LMGMM (H. Chang & J. R. Glass, 2007 [25])	81.3 %

度は, 本実験で使用した音声を CNMF で変換せずに, メルスペクトログラムのまま SVM へ入力し音素分類実験を行った結果を示している.

結果の表より, メルスペクトログラムよりも提案手法により変換した方が精度が良いことが確認できる. これは, CNMF により抽出した基底の中には音韻情報が含まれているためと考えられ, それゆえに抽出した基底は音声認識に役立てることが出来ると考えられる. しかし提案手法を他の既存手法と比べると, 精度が劣っていることを確認する事が出来る.

提案手法が既存の手法より精度で劣る理由としては, まず収束の遅さが挙げられる. 本研究では CD 更新式が速く収束することを示したが, それでもかなりの計算時間を要する. それに対して, CDBNs では Contrastive Divergence という手法でパラメータの学習を行う [4]. これは, 連続音声から基底の長さだけデータをサンプルし学習するステップを繰り返す手法である. この手法により, CDBNs は CNMF に比べ格段に速くパラメータを学習できる. しかしながら, Contrastive Divergence を用いた手法では, 解の収束性が保証されていない問題も持っている. それに対して, CNMF の更新では解の収束性が保証されており, 計算資源が豊富な環境においては CNMF が優れていると言える.

提案手法が劣る理由は, 他にパターン行列の制約が挙げられる. CDBNs では CNMF のパターン行列に相当する要素が負の値を取る事が許されている. そのために, CDBNs は CNMF よりもより複雑な形でパターンを学習し, それが精度の差に繋がったとも考えられる.

以上より, 提案手法は既存手法よりは精度で劣っており, 更なる改良が必要であると考えられる.

第5章　結論

5.1　まとめ

本稿では, CNMF を用いて音声の基底を教師無しで学習する手法を提案し, 抽出した基底を用いた音素分類実験を行うことで, 抽出した基底が音声認識に利用できることを示した. 更に, CNMF の更新式を求める際に CD を用いることで, 従来の AF を用いた更新式よりも速く収束することを示すことが出来た.

CNMF を用いた音素分類の実験より, 音声に含まれている基底の数は 150 個, 基底の長さは大きくても $4(= 32[\mathrm{msec}])$ と考えられ, この知見は今後音声認識の研究で役立てる事が出来ると考えられる.

しかしながら, 今回提案した手法は音素分類実験において従来手法を越える精度を出す事が出来なかった. その理由として挙げられるのは, CNMF 自体の収束の遅さや, CNMF が持つ制約が挙げられる. 今後はこれらの課題を克服することで, 従来手法と同程度かそれ以上の精度が出せると考えられる.

5.2　今後の課題

本研究の提案手法は, 基底となる音声を学習するときに厳密に収束性が保証されており, その点は CDBNs に比べ優位にあるが, CDBNs に比べ学習に掛かる時間が長い側面も持つ. そのために, CNMF による基底の学習ではある程度発散値が小さくなったところで学習を終了させていた. しかしながらそれでは十分に収束していない可能性が存在する.

そこで基底学習時の計算時間を短くするために, 更新するとより発散値を下げるパラメータのみを選択的に更新するアルゴリズムの導入することが考えられる. これは CD を用いた NMF に応用したアルゴリズムの一つで, 発散値を計算するために必要な値を上手く更新 · 保存することで, 効率的に更新後の発散値を予測し, その結果に応じてパラメータを順次更新するものである [26]. これにより, 通常の CD に比べ約 4 分の 1 の計算時間で収束することが報告されている. このアルゴリズムはま

だ CNMF では提案されていないが, NMF の場合と同等の効果が得られると推測されるために, 計算時間を短くする有効な手段の一つと考えられる.

また本実験では CNMF のパラメータを, 様々な値に変え精度が良くなる値の組み合わせを探索したが, CNMF ではベイズ推定を用いたパラメータの推定法が提案されている [27]. この手法では CNMF に基底を学習させる前に, 先に入力行列から隠れているパターンの数を推定し. その後で実際に入力行列から基底を学習させる. この手法は, 適切な CNMF のパラメータが学習出来るだけではなく, 各基底ごとに基底長を設定することも出来る. これにより基底の数だけが設定できるだけでなく, 各基底が適切な基底長を持つことも出来る.

もし適切な基底数や基底長が求まるとすれば, CNMF による分解はより適切な形で行われ, さらに音素分類で用いるために行った重み行列の変換もより適切な形に行われ, 音素分類の精度は上昇すると考えられる.

また本研究で用いた手法は, 最終的には音声認識に役立てたいと考えている. そこで音素分類から, より音声認識に近い音素認識の実験も行い, この手法の評価を行いたいと考えている. 音素分類実験では, 重み行列をどのように変換し, 上手く基底の重みを各フレームに割り当てるかということが重要になる. それに比べると, 音素認識の実験では重み行列の値の遷移から音素列を推定するため, 必ずしもパターンが存在するフレームに上手く重みを割り当てる必要が小さくなる. そのため, 音素分類実験とはまた異なる結果が音素認識実験で出る可能性があり, こちらの実験による評価も提案手法には必要と考える.

参考文献

[1] Honglak Lee, Chaitanya Ekanadhan, and Andrew Y. Ng. Sparse deep belief net model for visual area V2. In *Advances in Neural Information Processing Systems (NIPS) 2008*, Vol. 20, 2008.

[2] Li Deng. A generalized hidden markov model with state-conditioned trend functions of timefor the speech signal. *Signal Processing*, Vol. 27, No. 1, pp. 65–78, 1992.

[3] Honglak Lee, Yan Largman, Peter Pham, and Andrew Y. Ng. Unsupervised feature learning for audio classfication using convolutional deep belief networks. In *Advances in Neural Information Processing Systems (NIPS) 2009*, Vol. 22, 2009.

[4] Miguel Á. Carreira-Perpi nán and Geoffrey E. Hinton. On contrastive divergence learning. In *International Workshop on Artificial Intelligence and Statistics (AISTATS) 2005*, pp. 59–66, 2005.

[5] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, Vol. 18, No. 7, pp. 1527–1554, 2006.

[6] Kai-Fu Lee and Hsiao-Wuen Hon. Speaker-independent phone recognition using hidden markov models. *IEEE Transactions on Acoustic, Speech and Signal Processing*, Vol. 37, No. 11, pp. 1641–1648, Nobember 1989.

[7] Paris Smaragdis. Non-negative matrix factor deconvolution; extraction of multiple sound sourses from monophonic inputs. In *Lecture Notes in Computer Science 3195 Springer*, pp. 494–499. Springer-Verlag Berlin Heidelberg, 2004.

[8] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, Vol. 401, pp. 788–791, 1999.

[9] Emmanouil Benetos, Margarita Kotti, and Constantine Kotropoulos. Musical instrument classification using non-negative matrix factorization algorithms. In *International Symposium on Circuits and Systems (ISCAS) 2006*, pp. 21–24, 2006.

[10] Bhiksha Raj, Rita Singh, and Paris Smaragdis. Recognizing speech from simultaneous speakers. In *Interspeech 2005*, 2005.

[11] Andrzej Cichocki and Anh-Huy Phan. Fast local algorithms for large scale nonnegative matrix and tensor factorizations. *IEICE Transaction on Fundamentals*, Vol. E92-A, No. 3, pp. 708–721, 2009.

[12] Patrik O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Machine Learning Research*, Vol. 5, pp. 1457–1469, 2004.

[13] Tuomas Virtanen and Anssi Klapuri. Analysis of polyphonic audio using source-filter model and non-negative matrix factorization. In *Advances in Models for Acoustic Processing, Neural Information Processing Systems Workshop*, 2006.

[14] Jan de Leeuw. Block-relaxation algorithms in statistics. In H. H. Bock, W. Lenski, and M. M. Richter, editors, *Information Systems and Data Analysis*, Berlin, 1994. Springer Verlag.

[15] Henry Lindsay-Smith, Skot McDonald, and Mark Sandler. Drumkit transcription via convolutive nmf. In *International Conference on Digital Audio Effects (DAFx-12)*, 2012.

[16] Paul D. O'Grady and Barak A. Pearlmutter. Discoveringspeech phones using convolutive non-negative matrix factorisation with a sparseness constraint. *Neurocomputing*, Vol. 72, No. 1-3, pp. 88–101, 2008.

[17] Vladimir N. Vapnik and A. J. Lerner. Pattern recognition using generalized portrait method. *Automation and Remote Control*, Vol. 24, pp. 774–780, 1963.

[18] Corinna Cortes and Vladimir N. Vapnik. Support-vector networks. *Machine Learning*, Vol. 20, No. 3, pp. 273–297, 1995.

[19] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In David Haussler, editor, *5th Annual ACM Conference on Learning Theory (COLT)*, pp. 144–152, Pittsburgh, PA, 1992. ACM Press.

[20] Mark A. Aizerman, Emmanuel M. Braverman, and Lev I. Rozonoér. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, Vol. 25, pp. 821–837, 1964.

[21] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, Vol. 13, No. 2, pp. 415–425, 2002.

[22] Dominik Scherer, Andreas Müller, and Sven Behnke. Evaluation of pooling operations in convolutional architectures for object recognition. In *International Conference on Artificial Neural Networks (ICANN) 2010*, pp. 92–101, 2010.

[23] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and Victor Zue. Timit acoustic-phonetic continuous speech corpus. LDC, Philadelphia, 1993.

[24] Kihyuk Sohn and Honglak Lee. Learning invariant representations with local transformations. In *International Conference on Machine Learning (ICML) 2012*, 2012.

[25] Hung-An Chang and James R. Glass. Hierarchical large-margin gaussian mixture models for phonetic classication. In *IEEE Workshop on Automatic Speech Recognition and Uunderstanding (ASRU) 2007*, pp. 272–277, 2007.

[26] Cho-Jui Hsieh and Inderjit S. Dhillon. Fast coordinate descent methods with variable selection for non-negative matrix factorization. In *the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1064–1072, 2011.

[27] Ron J. Weiss and Juan Pablo Bello. Identifying repeated patterns in music using sparseconvolutive non-negative matrix factorization.

謝辞

本研究を進める中で様々な方々にお世話になりました.

近山隆教授には, 私の研究の方向性や研究内容の示し方などをご教授頂き, それにより私は本研究を最後までやり遂げることが出来ました. またそれ以外にも私生活に関して様々なご配慮を頂き, それによって今日まで研究を続けることが出来ました. また鶴岡慶雅准教授には, 機械学習の観点から私の手法の改善点を指摘していただき, 今回の形にまとめることが出来ました. また近山研究室 OB の三輪誠さんには, 研究の不明瞭な部分を指摘して頂き, 資料の作り方や研究の進め方に関しても様々なアドバイスを頂きました.

その他にも, 近山 · 鶴岡研究室の方々に感謝しております. 博士 2 年の浦晃さんと現金田研究室修士 2 年の鈴木洋平君には, 機械学習の観点から私の研究に対する意見を頂き, それを参考にして研究の質を高めることが出来ました. また修士 2 年の古居敬大君や関栄二君には, 発表内容で不明瞭な部分を指摘して頂き, 研究内容の発表の質を高めることが出来ました.

ここには書ききれませんが, その他にも多くの方々に支えられこれまでの研究生活を送ることができました. この場をお借りして厚くお礼申し上げます.

最後に, このような素晴らしい環境で研究を行えるように育ててくださった私の両親に感謝いたします. 本当にありがとうございました.

平成 25 年 2 月 6 日