ARG WI2 No.10, 2017
WI2-2017-1

多層パーセプトロンを用いた効率的な共有潜在空間の推定

大山 まりほ[†,a]　　小林 一郎[††,b]

† お茶の水女子大学大学院　†† お茶の水女子大学基幹研究院

a) *g1220509@is.ocha.ac.jp*　b) *koba@is.ocha.ac.jp*

概要　複数の異なる次元の時系列データの対応を考える際，それらのデータを共有する空間上で同じ次元に揃えて比較可能にする必要がある．課題を達成するために，本研究では，ガウス過程に基づく高次元の時系列データの圧縮手法である Gaussian Process Latent Variable Models(GPLVM) を次元圧縮に用い，潜在空間を推定する．GPLVM には EM アルゴリズムが用いられており，計算量が膨大で実用化は難しい．そのため，EM アルゴリズムの代わりに多層パーセプトロン (MLP:Multilayer Perceptron) を組み込み，非線形識別を可能にし，且つ計算量の削減を実現する．実用を目指したより効率的な潜在空間識別モデルの構築を行う．本提案手法を GPLVM，Shared Gaussian Process Latent Variable Models(sharedGPLVM)，Gaussian Process Dynamical Models(GPDM)，Shared Gaussian Process Dynamical Models(sharedGPDM) に導入し，効果を実証する．また，本提案手法を用いて 2 つの時系列データを共有する潜在空間へ圧縮を行い，対応関係の学習を行う．取得した潜在空間から観測空間である時系列データの復元についても取り組む．

キーワード　ガウス過程，次元圧縮，多層パーセプトロン

1　はじめに

唇の動き, 音声発話, しぐさなど複数かつ高次元の時系列データの対応関係を捉えることは重要である. それらの対応を考える際，異なる次元を持つ各データを共有する空間上に射影し次元を揃えて比較可能にする必要がある．本研究ではガウス過程に従う高次元の時系列データを対象にし，まず Gaussian Process Latent Variable Models(GPLVM) を用い潜在空間上での次元圧縮を考察し，従来の線形識別に対し精度の向上のため，非線形識別可能な Multilayer perceptron(MLP) を用いる手法を導入した. 次にそれを用いて複数の時系列データの対応関係を捉えることが出来る Shared Gaussian Process Latent Variable Models(sharedGPLVM) を構築した．しかし sharedGPLVM では，次元圧縮を行う際に潜在空間における時系列情報を考慮していないため，観測空間内におけるデータの系列を精度良く潜在空間内に圧縮することが難しいことが判明した．このことから本研究では，潜在空間内のデータにおける関係もガウス過程でとらえる Gaussian Process Dynamical Models(GPDM) に MLP の最適化手法を適用した手法の提案を行う.

2　GPLVM

本研究では，GPLVM[1] を用いてガウス過程で表現される高次元の時系列データを潜在空間に圧縮する．GPLVM とは，確率的主成分分析にガウス過程を導入したものである．GPLVM を使用した次元圧縮のアルゴリズムを Algorithm 1 に示す.

Algorithm 1 GPLVM
Require: $\boldsymbol{X}$,$\boldsymbol{Y}$,$\boldsymbol{K}$,T
　Initialize $\boldsymbol{X}$ through PCA.
　for T iterations **do**
　　Optimise Eq.(1) with respect to the parameters of $\boldsymbol{K}$ using scaled conjugate gradients.
　　Select a new $\boldsymbol{X}$ using the parameters of $\boldsymbol{K}$.
　end for

観測データ全体を $\boldsymbol{Y}$，圧縮先の潜在空間上のデータ全体を $\boldsymbol{X}$ とする．まず Principal component analysis(PCA) を用いて $\boldsymbol{Y}$ を圧縮し，$\boldsymbol{X}$ の初期値を設定する．そして $\boldsymbol{X}$，$\boldsymbol{Y}$ それぞれをガウス過程に従っていると仮定した正規化を行う．その後，以下の式 (1) で表される $p(\boldsymbol{Y}|\boldsymbol{X},\alpha,\beta,\gamma)$ の対数尤度が最大となるよう式 (2) で表されるカーネル関数のパラメータ α，β，γ を更新する．更新したパラメータを当てはめ，$\boldsymbol{X}$ を再設定する．

$$\begin{aligned} p(\boldsymbol{Y}|\boldsymbol{X},\alpha,\beta,\gamma) &= \frac{1}{(2\ \pi\ )^{DN/2}|\boldsymbol{K}|^{D/2}} \\ &\times \exp(-\frac{1}{2}\mathrm{tr}(\boldsymbol{K}^{-1}\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}})) \quad (1) \end{aligned}$$

$$\begin{aligned} k(\boldsymbol{x}_n,\boldsymbol{x}_m) &= \alpha \exp(-\frac{\gamma}{2}(\boldsymbol{x}_n-\boldsymbol{x}_m)^{\mathrm{T}}(\boldsymbol{x}_n-\boldsymbol{x}_m)) \\ &+ \delta_{nm}\beta^{-1} \quad (2) \end{aligned}$$

カーネルのパラメータの更新を T 回繰り返すことで $\boldsymbol{X}$ が更新され，高次元の $\boldsymbol{Y}$ に対して低次元の潜在空間 $\boldsymbol{X}$ が求まる．しかし，このアルゴリズムでは予め定めた回数 T 回 $\boldsymbol{X}$ の更新を繰り返しており，収束判定を行っていない．そのため最適な $\boldsymbol{X}$ が求められた保証がされていない．そのため収束判定を行うために MLP を組み込

み，収束判定をするアルゴリズムを作成した．そのアルゴリズムを Algorithm 2 に示す.

Algorithm 2 GPLVM with MLP

Require: $\boldsymbol{X}$,$\boldsymbol{Y}$,$\boldsymbol{K}$

  Initialize $\boldsymbol{X}$ through PCA.

  while Gradient $\log p(\boldsymbol{Y}, \boldsymbol{X}, \alpha, \beta, \gamma, \boldsymbol{M}) \leq \varepsilon$ **do**

    Select a new $\boldsymbol{X}$ through feed forward MLP.

    Optimise $\log p(\boldsymbol{Y}, \boldsymbol{X}, \alpha, \beta, \gamma, \boldsymbol{M})$ with respect to the parameters of $\boldsymbol{K}$ and the synaptic weights of MLP using scaled conjugate gradients.

  end while

まず GPLVM のみの場合と同様に PCA を用いて $\boldsymbol{Y}$ を圧縮し，$\boldsymbol{X}$ の初期値を設定する．そして $\boldsymbol{X}$，$\boldsymbol{Y}$ それぞれをガウス過程に従っていると仮定した正規化を行う．次に MLP を用いる．$\boldsymbol{Y}$ を入力，$\boldsymbol{X}$ を出力として結合荷重を求め，$\boldsymbol{X}$ と $\boldsymbol{Y}$ の対応関係を学習させておく．その後，式 (3) で表される $\log p(\boldsymbol{Y}, \boldsymbol{X}; \alpha, \beta, \gamma)$ が最大となるよう，式 (2) で表されるカーネル関数のパラメータ α，β，γ と MLP の結合荷重 $\boldsymbol{w}$ の更新を行い，$\boldsymbol{X}$ を再設定する．結合荷重の更新とカーネルのパラメータの更新を式 (3) の勾配が収束するまで繰り返すことで $\boldsymbol{X}$ が更新され，高次元の $\boldsymbol{Y}$ に対して低次元の潜在空間 $\boldsymbol{X}$ が求まる．

$$\log p(\boldsymbol{Y}, \boldsymbol{X}|\alpha, \beta, \gamma) = \log p(\boldsymbol{Y}|\boldsymbol{X}, \alpha, \beta, \gamma) + \frac{1}{2}\sum_i {\boldsymbol{X}_i}^2 + \frac{1}{2}\sum_j {\boldsymbol{w}_j}^2 \quad (3)$$

3 実験：GPLVM

構築したアルゴリズムの正当性を検証するために，Ek[6] において sharedGPLVM の検証用に使用されていたのと同じサンプルプログラムを作成し，動作結果を比較した．式 (4)〜(6) の 3 式を基底としてガウス分布に従う重みをかけ足し合わせ，ノイズを付加し 20 次元の観測データ $\boldsymbol{Y}$ を作成する．20 次元の観測データ $\boldsymbol{Y}$ を 3 次元の潜在空間 $\boldsymbol{X}$ に圧縮する．作成した $\boldsymbol{Y}$ は図 1 の左図のようになった．圧縮した結果，基底である上記 3 式の曲線 (図 1:右) に近い形が出力されるのが理想である．

$$x_i^1 = \cos(\pi t_i) \quad (4)$$

$$x_i^2 = \sin(\pi t_i) \quad (5)$$

$$x_i^3 = \cos(\sqrt{5}\pi t_i + 2) \quad (6)$$

3.1 実験結果

GPLVM により得られた結果を図 2 に示す．左は GPLVM，右は GPLVM に MLP を組み込んだものである． また，

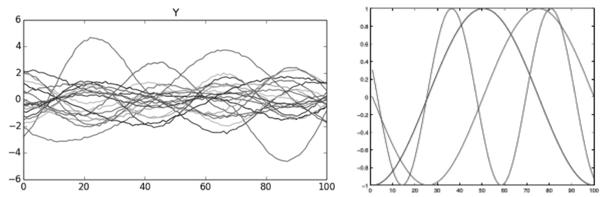

図 1 左：観測データ $\boldsymbol{Y}$　右：$\boldsymbol{Y}$ の基底 (式 (4)〜(6))

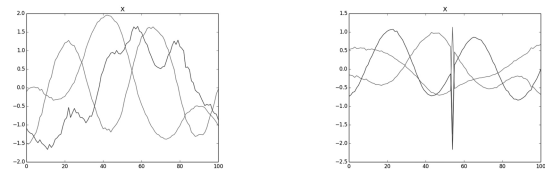

図 2 左：MLP 有　右：MLP 無

処理時間はそれぞれ以下の表のようになった．

表 1 elapsed time

original GPLVM	GPLVM with MLP
5872.89[sec]	25.27[sec]

3.2 考察

Lawrence[1] によって提案された元々のアルゴリズムでは，カーネル関数 $\boldsymbol{K}$ のパラメータを求める際の $\boldsymbol{X}$ の初期値および繰り返しの回数によって得られる $\boldsymbol{X}$ がガウス過程に従っているか定かではないケースが多く観測された．そこで本研究では，GitHub 上で公開されているソースコード [1] を参考に，対数尤度の微分値の収束を観測できる MLP を採用し，高次元の動作データを低次元に圧縮し良い結果を得ることができた．表 1 より，MLP を導入することで圧倒的に実行時間が短縮されたことがわかる．MLP の導入によって，精度の向上と計算量の削減を実現した．提案するアルゴリズムは精度の向上と実行時間削減を実現する．

4 sharedGPLVM

対応関係の取得を行うために 2 つの観測空間で 1 つの潜在空間を共有することで，2 つの観測空間の関係を学習することの出来る sharedGPLVM[6] を用いる．複数の時系列データを共有の潜在空間に圧縮させ，新たないずれかの時系列データ入力された際対応する双方での時系列データへの復元を行う．sharedGPLVM の概要図を図 3 に示す．sharedGPLVM は大別して学習 (共有する潜在空間の構築) とマッピング (取得された潜在空間を通じた 2 つの時系列データの対応) の 2 つの段階に分けられる．

---

[1] http://github.com/jameshensman

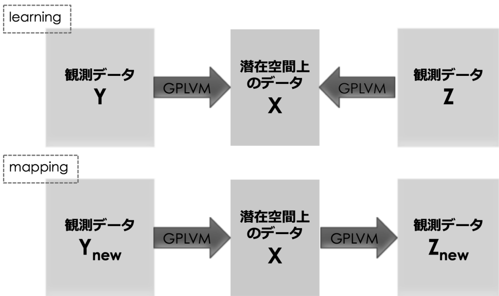

図 3 sharedGPLVM の概要図

4.1 学習

sharedGPLVM を用いて 2 つの観測空間から共有する潜在空間へと次元圧縮を行う．前節の実験結果より，GPLVM と同様に MLP を組み込む．学習アルゴリズムを Algorithm3 に示す．

Algorithm 3 sharedGPLVM with MLP:learning

Require: $\boldsymbol{X}$,$\boldsymbol{X_Y}$,$\boldsymbol{X_Z}$,$\boldsymbol{Y}$,$\boldsymbol{Z}$,$\boldsymbol{K_Y}$,$\boldsymbol{K_Z}$

Initialize $\boldsymbol{X_Y}$,$\boldsymbol{X_Z}$ through PCA.
Initialize $\boldsymbol{\phi_Y}$,$\boldsymbol{\phi_Z}$.
Calculate $\boldsymbol{X}$ using $\boldsymbol{X_Y}$ and $\boldsymbol{X_Z}$.
Calculate $p(\boldsymbol{Y},\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\phi_Y},\boldsymbol{\phi_Z})$.
while Gradient $\log p(\boldsymbol{Y},\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\phi_Y},\boldsymbol{\phi_Z}) \leq \varepsilon$ **do**
  Select a new $\boldsymbol{X'_Y}$,$\boldsymbol{X'_Z}$ using the parameter of $\boldsymbol{K_Y}$,$\boldsymbol{K_Z}$ through feed forward MLP.
  Optimise $\log p(\boldsymbol{Y},\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\phi_Y},\boldsymbol{\phi_Z})$ with respect to the parameters of $\boldsymbol{K_Y}$,$\boldsymbol{K_Z}$ and the synaptic weights of MLP using scaled conjugate gradients.
  Calculate $\boldsymbol{X'}$ using $\boldsymbol{X'_Y}$ and $\boldsymbol{X'_Z}$.
end while

まず PCA を用いて観測空間 $\boldsymbol{Y}$ と $\boldsymbol{Z}$ を圧縮し，$\boldsymbol{X_Y}$ と $\boldsymbol{X_Z}$ をそれぞれ求める．それらの平均を取り $\boldsymbol{X}$ の初期値を設定する．そして $\boldsymbol{X}$，$\boldsymbol{Y}$，$\boldsymbol{Z}$ それぞれをガウス過程に従っていると仮定した正規化を行う．次に MLP を用いる．$\boldsymbol{Y}$ と $\boldsymbol{Z}$ を入力，$\boldsymbol{X}$ を出力としてそれぞれ結合荷重を求め，$\boldsymbol{X}$ と $\boldsymbol{Y}$，$\boldsymbol{X}$ と $\boldsymbol{Z}$ の対応関係を学習させておく．その後，以下の式 (7) で表される $\log p(\boldsymbol{Y},\boldsymbol{Z},\boldsymbol{X};\boldsymbol{\phi_Y},\boldsymbol{\phi_Z})$ が最大となるよう，$\boldsymbol{Y}$，$\boldsymbol{Z}$ それぞれのカーネル関数のパラメータ $\boldsymbol{\phi_Y}$，$\boldsymbol{\phi_Z}$ と MLP の結合荷重 $\boldsymbol{w_Y}$,$\boldsymbol{w_Z}$ の更新を行い，$\boldsymbol{X_Y}$ と $\boldsymbol{X_Z}$ を再設定し，再びそれらの平均をとることで $\boldsymbol{X}$ を再設定する．勾配が収束するまで繰り返すことで $\boldsymbol{X}$ が更新され，高次元の $\boldsymbol{Y}$，$\boldsymbol{Z}$ に対して低次元の共有の潜在空間 $\boldsymbol{X}$ が求まる．

$$\log p(\boldsymbol{Y},\boldsymbol{Z},\boldsymbol{X};\boldsymbol{\phi_Y},\boldsymbol{\phi_Z}) = \log p(\boldsymbol{Y},\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\phi_Y},\boldsymbol{\phi_Z}) + \frac{1}{2}\sum_i {\boldsymbol{X}_i}^2 + \frac{1}{2}\sum_j {\boldsymbol{w_Y}_j}^2 + \frac{1}{2}\sum_k {\boldsymbol{w_Z}_k}^2 \quad (7)$$

$$p(\boldsymbol{Y},\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\phi_Y},\boldsymbol{\phi_Z}) = P(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\phi_Y})P(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\phi_Z}) \times P(\boldsymbol{\phi_Y})P(\boldsymbol{\phi_Z})P(\boldsymbol{X})$$

$$\begin{aligned} P(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\phi_Y}) &= \frac{1}{\sqrt{(2\ \pi\ )^{ND_{\boldsymbol{Y}}}|\boldsymbol{K_Y}|^{D_{\boldsymbol{Y}}}}} \\ &\times\ \exp(-\frac{1}{2}\sum_{k=1}^{D_{\boldsymbol{Y}}} {\boldsymbol{y}_k}^T {\boldsymbol{K_Y}}^{-1}\boldsymbol{y}_k) \end{aligned}$$

$$\begin{aligned} P(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\phi_Z}) &= \frac{1}{\sqrt{(2\ \pi\ )^{ND_{\boldsymbol{Z}}}|\boldsymbol{K_Z}|^{D_{\boldsymbol{Z}}}}} \\ &\times\ \exp(-\frac{1}{2}\sum_{k=1}^{D_{\boldsymbol{Z}}} {\boldsymbol{z}_k}^T {\boldsymbol{K_Z}}^{-1}\boldsymbol{z}_k) \end{aligned}$$

4.2 マッピング

新たな高次元の観測データ $\boldsymbol{Y}_{new}$ が入力された際，潜在空間 $\boldsymbol{X}$ を通して他の高次元の $\boldsymbol{Z}_{new}$ を出力する．$\boldsymbol{Y}_{new}$ の $\boldsymbol{y}_{new}$ に対応する潜在空間 $\boldsymbol{X}$ 上の点 $\boldsymbol{x}_{new}$ から，以下の式 (8) で表される $\boldsymbol{z}_{new}$ の確率分布が得られる．$k(\boldsymbol{x}_{new})$ は i 行目が $k_i(\boldsymbol{x}_{new}) = k(\boldsymbol{x}_{new},\boldsymbol{x}_i)$ となるベクトルを表している．マッピングのアルゴリズムを Algorithm4 に示す．

$$\begin{aligned} \overline{z}(\boldsymbol{x}_{new}) &= \boldsymbol{\mu_Z} + \boldsymbol{Z}^T{\boldsymbol{K_Z}}^{-1}k(\boldsymbol{x}_{new}) \\ \boldsymbol{\sigma}^2_{\boldsymbol{Z}}(\boldsymbol{x}_{new}) &= k(\boldsymbol{x}_{new},\boldsymbol{x}_{new}) \\ &-\ k(\boldsymbol{x}_{new})^T{\boldsymbol{K_Z}}^{-1}k(\boldsymbol{x}_{new}) \end{aligned} \quad (8)$$

Algorithm 4 SharedGPLVM with MLP:mapping

Require: $\boldsymbol{X}_{new}$,$\boldsymbol{Z}$,$\boldsymbol{K_Z}$

for $\boldsymbol{X}_{new}$ **do**
  Calculate Eq.(8) using $\boldsymbol{Z}$,$\boldsymbol{K_Z}$.
end for

5 実験：sharedGPLVM

5.1 学習

3 節で作成した 20 次元の観測データ $\boldsymbol{Y}$ に加え，同様に 30 次元の観測データ $\boldsymbol{Z}$ を作成する (図 4)．観測データ $\boldsymbol{Y}$ と $\boldsymbol{Z}$ を共有する 3 次元の潜在空間 $\boldsymbol{X}$ に圧縮する．結果は図 5 のようになった．

5.2 マッピング

学習で生成された $\boldsymbol{X}$ を $\boldsymbol{X}_{new}$ とし，マッピングを行った．結果として $\boldsymbol{Z}$ が復元されるのが理想である．結果は図 6 の右図のようになった．結果を比較するため $\boldsymbol{Z}$ と復元結果の $\boldsymbol{Z}$ の一部 (10 次元) を図 7 に示す．

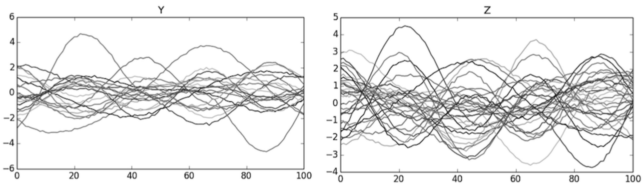

図 4 観測データ (左：$\boldsymbol{Y}$　右：$\boldsymbol{Z}$)

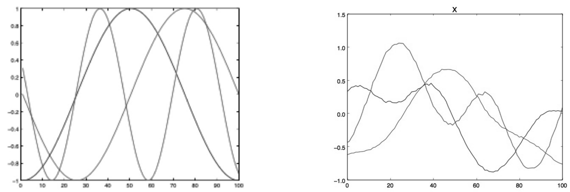

図 5 左：$\boldsymbol{Y}$，$\boldsymbol{Z}$ の基底　右：共有の潜在空間 $\boldsymbol{X}$

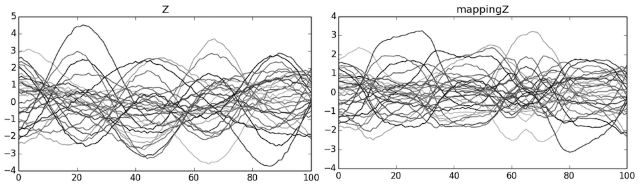

図 6 左：$\boldsymbol{Z}$(オリジナル)　右：マッピング $\boldsymbol{Z}$

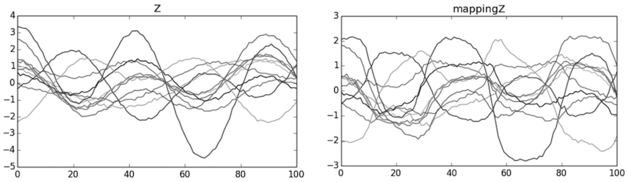

図 7 左：$\boldsymbol{Z}$ の一部　右：マッピング $\boldsymbol{Z}$ の一部

5.3　考察

図 5 を見ると，圧縮結果も基底と類似した形をしていることが分かる．ガウス過程に従う状態で圧縮が出来ており，望ましい結果になった．また，図 6 を見ると，縦軸の範囲に多少差が見られるが $\boldsymbol{Z}$ に近い形が復元されていることが分かる．学習，マッピング共に理想的な結果が得られた．このことから，提案するアルゴリズムにより，実際に人の動作空間とロボットの動作空間に用いることで潜在空間を通して対応関係の学習，マッピングが可能だと考えられる．

6　GPDM

GPLVM では潜在空間 $\boldsymbol{X}$ でのダイナミクスについては考慮されていないため，さらなる精度の向上を図り，潜在空間のダイナミクスを扱うことが出来る GPDM[11] を用いる．GPLVM や sharedGPLVM の際と同様に精度の向上のため MLP を組み込む．GPDM のアルゴリズムを Algorithm5 に示す．$\boldsymbol{\phi_X}$,$\boldsymbol{\phi_Y}$ はいずれも $\boldsymbol{X}$,$\boldsymbol{Y}$ のカーネル関数のパラメータである．$\boldsymbol{W}$ は観測 データの次元毎の分散の違いを補正するスケールパラメータ行列であり，$\boldsymbol{W} \equiv diag(w_1, ..., w_{D_{\boldsymbol{Y}}})$ で表される．

Algorithm 5 GPDM with MLP

Require: $\boldsymbol{X}$,$\boldsymbol{Y}$,$\boldsymbol{K_X}$,$\boldsymbol{K_Y}$,$\boldsymbol{W}$
  Initialize $\boldsymbol{X}$ through PCA.
  while Gradient $\log p(\boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{\phi_Y}, \boldsymbol{\phi_X}, \boldsymbol{M}, \boldsymbol{W}) \leq \varepsilon$ **do**
    Select a new $\boldsymbol{X}$ through feed forward MLP.
    Optimise $\log p(\boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{\phi_Y}, \boldsymbol{\phi_X}, \boldsymbol{M}, \boldsymbol{W})$ with respect to the parameters of $\boldsymbol{K_X}$ and $\boldsymbol{K_Y}$, the synaptic weights of MLP using scaled conjugate gradients.
    for $k = 1$ to $D_{\boldsymbol{Y}}$ **do**
      $w_k \Leftarrow \sqrt{N(\boldsymbol{y}_{:,k}^T \boldsymbol{K_Y}^{-1} \boldsymbol{y}_{:,k})^{-1}}$
    end for
  end while

PCA を用いて観測空間 $\boldsymbol{Y}$ を圧縮し，$\boldsymbol{X}$ を求める．$\boldsymbol{X}$，$\boldsymbol{Y}$ をそれぞれガウス過程に従っていると仮定し正規化を行う．次に MLP を用いる．$\boldsymbol{Y}$ と $\boldsymbol{Z}$ を入力，$\boldsymbol{X}$ を出力としてそれぞれ結合荷重を求め，$\boldsymbol{X}$ と $\boldsymbol{Y}$，$\boldsymbol{X}$ と $\boldsymbol{Z}$ の対応関係を学習させておく．その後，以下の式 (9) で表される $p(\boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{\phi_Y}, \boldsymbol{\phi_X})$ が最大となるよう，$\boldsymbol{Y}$，$\boldsymbol{X}$ それぞれのカーネル関数のパラメータ $\boldsymbol{\phi_Y}$，$\boldsymbol{\phi_X}$ と MLP の結合荷重 $\boldsymbol{w_Y}$ の更新を行い，$\boldsymbol{X}$ を再設定する．勾配が収束するまで繰り返すことで $\boldsymbol{X}$ が更新され，高次元の $\boldsymbol{Y}$ に対して最適な低次元の共有の潜在空間 $\boldsymbol{X}$ が求まる．$\boldsymbol{X}$ のカーネル関数には RBF カーネルに線形カーネルを追加したものを用いる (式 (10))．

$$
\begin{aligned}
& p(\boldsymbol{Y}, \boldsymbol{X}, \phi_{\boldsymbol{Y}}, \phi_{\boldsymbol{X}}) \\
& = p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\phi_Y}) p(\boldsymbol{X}|\boldsymbol{\phi_X}) p(\boldsymbol{\phi_X}) p(\boldsymbol{\phi_Y}) \\
& = \frac{|\boldsymbol{W}|^N}{(2\ \pi)^{D_{\boldsymbol{Y}}N/2} |\boldsymbol{K_Y}|^{D_{\boldsymbol{Y}}/2}} \exp(-\frac{1}{2} \mathrm{tr}(\boldsymbol{K_Y}^{-1} \boldsymbol{Y} \boldsymbol{W}^2 \boldsymbol{Y}^{\mathrm{T}})) \\
& \times p(x_1) \frac{1}{(2\ \pi)^{D_{\boldsymbol{X}}(N-1)/2} |\boldsymbol{K_X}|^{D_{\boldsymbol{X}}/2}} \exp(-\frac{1}{2} \mathrm{tr}(\boldsymbol{K_X}^{-1} \boldsymbol{X}_{out} \boldsymbol{X}_{out}{}^{\mathrm{T}})) \\
& \times p(\boldsymbol{\phi_X}) p(\boldsymbol{\phi_Y}) \qquad (9)
\end{aligned}
$$

$$
\begin{aligned}
k(x_n, x_m) &= \alpha_{\boldsymbol{X}} \exp(-\frac{\gamma_{\boldsymbol{X}}}{2} (x_n - x_m)^{\mathrm{T}} (x_n - x_m)) \\
&+ \delta_{nm} \beta_{\boldsymbol{X}}^{-1} + \eta_{\boldsymbol{X}} x_n \cdot x_m \qquad (10)
\end{aligned}
$$

6.1　実験

GPLVM の場合と同様に，3 つの基底から生成された 20 次元の観測データ $\boldsymbol{Y}$ を GPDM を用いて 3 次元に圧縮した．結果を図 8 に示す．また、処理時間はそれぞれ以下の表のようになった．

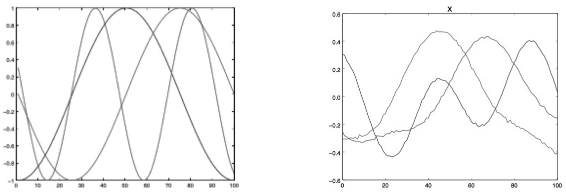

図 8 左：$\boldsymbol{Y}$，$\boldsymbol{Z}$ の基底　右：共有の潜在空間 $\boldsymbol{X}$

表 2 elapsed time

original GPDM	GPDM with MLP
11262.35[sec]	17.42[sec]

6.2　考察

GPDM を用いて高次元の時系列データ $\boldsymbol{Y}$ を潜在空間に圧縮することが出来た．GPLVM との差異を視認することは難しいが，GPDM では潜在空間でのダイナミクスが考慮された形で圧縮がされている．表 2 より，MLP を導入することで圧倒的に実行時間が短縮されたことがわかる．MLP の導入によって，精度の向上と計算量の削減を実現した．よって GPLVM の場合と同様に GPDM に MLP を導入することは有効であると考えられる．

7　sharedGPDM

識別を行うために，高次元の時系列データである 2 つの観測空間が 1 つの潜在空間を共有することで，2 つの観測空間の関係を学習することができ、GPDM を基にした sharedGPDM[11] を用いる．sharedGPDM の構造は sharedGPLVM に潜在ダイナミクスを考えて共有する潜在状態を推定するものである．sharedGPLVM の場合と同様に，sharedGPDM にも MLP を導入する．

7.1　学習

学習フェーズのアルゴリズムを Algorithm6 に示す．

まず PCA を用いて，観測空間 $\boldsymbol{Y}$ と $\boldsymbol{Z}$ を圧縮し，$\boldsymbol{X_Y}$ と $\boldsymbol{X_Z}$ をそれぞれ求める．それらの平均を取り潜在空間 $\boldsymbol{X}$ の初期値を設定する．そして $\boldsymbol{X}$，$\boldsymbol{Y}$，$\boldsymbol{Z}$ それぞれをガウス過程に従っていると仮定した正規化を行う．次に，$\boldsymbol{Y}$ と $\boldsymbol{Z}$ を入力，$\boldsymbol{X}$ を出力としてそれぞれ結合荷重を求め，$\boldsymbol{X}$ と $\boldsymbol{Y}$，$\boldsymbol{X}$ と $\boldsymbol{Z}$ の対応関係を学習させておく．その後，以下の式 (11) で表される $\log p(\boldsymbol{Y},\boldsymbol{Z},\boldsymbol{X};\phi_Y,\phi_Z)$ が最大となるよう，$\boldsymbol{Y}$，$\boldsymbol{Z}$ それぞれのカーネル関数のパラメータ $\boldsymbol{\phi_Y}$，$\boldsymbol{\phi_Z}$ と MLP の結合荷重 $\boldsymbol{w_Y}$,$\boldsymbol{w_Z}$ の更新を行い，$\boldsymbol{X_Y}$ と $\boldsymbol{X_Z}$ を再設定し，再びそれらの平均をとることで $\boldsymbol{X}$ を再設定する．この操作を式 (11) の勾配

Algorithm 6 sharedGPDM with MLP:learning

Require: $\boldsymbol{X}$,$\boldsymbol{X_Y}$,$\boldsymbol{X_Z}$,$\boldsymbol{Y}$,$\boldsymbol{Z}$,$\boldsymbol{K_X}$,$\boldsymbol{K_Y}$,$\boldsymbol{K_Z}$

Initialize $\boldsymbol{X_Y}$,$\boldsymbol{X_Z}$ through PCA.

Initialize $\boldsymbol{\phi_X}$,$\boldsymbol{\phi_Y}$,$\boldsymbol{\phi_Z}$.

Calculate $\boldsymbol{X}$ using $\boldsymbol{X_Y}$ and $\boldsymbol{X_Z}$.

Calculate $p(\boldsymbol{Y},\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\phi_X},\boldsymbol{\phi_Y},\boldsymbol{\phi_Z})$.

while Gradient Eq.(11) $\leq$ ε **do**

Select a new $\boldsymbol{X'_Y}$,$\boldsymbol{X'_Z}$ using the parameter of $\boldsymbol{K_X}$,$\boldsymbol{K_Y}$,$\boldsymbol{K_Z}$ through feed forward MLP.

Optimise Eq.(11) with respect to the parameters of $\boldsymbol{K_X}$,$\boldsymbol{K_Y}$,$\boldsymbol{K_Z}$ and the synaptic weights of MLP using scaled conjugate gradients.

Calculate $\boldsymbol{X'}$ using $\boldsymbol{X'_Y}$ and $\boldsymbol{X'_Z}$.

end while

が収束するまで繰り返すことで最適な $\boldsymbol{X}$ が求められる．

$$\begin{aligned}&\log p(\boldsymbol{Y},\boldsymbol{Z},\boldsymbol{X},\boldsymbol{\phi_X},\boldsymbol{\phi_Y},\boldsymbol{\phi_Z},\boldsymbol{w})\\&=\log p(\boldsymbol{Y},\boldsymbol{Z},\boldsymbol{X},\boldsymbol{\phi_X},\boldsymbol{\phi_Y},\boldsymbol{\phi_Z})\\&+\tfrac{1}{2}\textstyle\sum_j {\boldsymbol{w_Y}_j}^2+\tfrac{1}{2}\sum_k {\boldsymbol{w_Z}_j}^2\end{aligned}\tag{11}$$

$$\begin{aligned}&p(\boldsymbol{Y},\boldsymbol{Z},\boldsymbol{X},\boldsymbol{\phi_X},\boldsymbol{\phi_Y},\boldsymbol{\phi_Z})\\&=p(\boldsymbol{Y},\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\phi_Y},\boldsymbol{\phi_Z})P(\boldsymbol{X}|\boldsymbol{\phi_X})\\&\times P(\boldsymbol{\phi_X})P(\boldsymbol{\phi_Y})P(\boldsymbol{\phi_Z})\\&=P(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\phi_Y})P(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\phi_Z})P(\boldsymbol{X}|\boldsymbol{\phi_X})\\&\times P(\boldsymbol{\phi_X})P(\boldsymbol{\phi_Y})P(\boldsymbol{\phi_Z})\end{aligned}\tag{12}$$

$$\begin{aligned}P(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\phi_Y})&=\frac{|\boldsymbol{W_Y}|^N}{\sqrt{(2\pi)^{ND_Y}|\boldsymbol{K_Y}|^{D_Y}}}\\&\times\exp(-\frac{1}{2}\sum_{k=1}^{D_Y}{\boldsymbol{w_{y_k}}}^2{\boldsymbol{y}_k}^T{\boldsymbol{K_Y}}^{-1}\boldsymbol{y}_k)\end{aligned}\tag{13}$$

$$\begin{aligned}P(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\phi_Z})&=\frac{|\boldsymbol{W_Z}|^N}{\sqrt{(2\pi)^{ND_Z}|\boldsymbol{K_Z}|^{D_Z}}}\\&\times\exp(-\frac{1}{2}\sum_{k=1}^{D_Z}{\boldsymbol{w_{z_k}}}^2{\boldsymbol{z}_k}^T{\boldsymbol{K_Z}}^{-1}\boldsymbol{z}_k)\end{aligned}\tag{14}$$

8　実験：sharedGPDM

8.1　学習

sharedGPLVM の場合と同様に,20 次元の観測データ $\boldsymbol{Y}$ と 30 次元の観測データ $\boldsymbol{Z}$ を sharedGPDM を用いて 3 次元に圧縮した．結果を図 9 に示す．

8.2　マッピング

sharedGPLVM の場合と同様に，学習で生成された $\boldsymbol{X}$ を $\boldsymbol{X}_{new}$ とし，マッピングを行った．結果を図 10 に示す．マッピングの成功例と失敗例を示す．

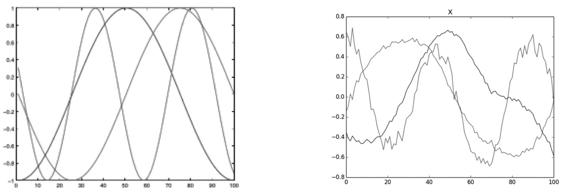

図 9 左：基底　右：結果

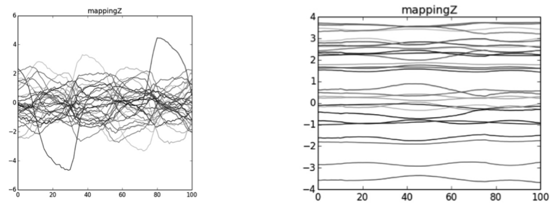

図 10 左：成功例　右：失敗例

8.3　考察

2 つの高次元な時系列データである $\boldsymbol{Y}$ と $\boldsymbol{Z}$ を sharedGPDM を用いて共有の低次元でありダイナミクスを考慮した潜在空間に圧縮することが出来た．しかしながら，複数回潜在空間 $\boldsymbol{X}$ を求めマッピングを行ったところ，半数近くは上手く復元がされず形が崩れてしまった．初期値の設定を PCA を用いて行っており，求められる潜在空間がその初期値の影響を少なからず受けていること，$\boldsymbol{Y}$ と $\boldsymbol{Z}$ それぞれの潜在空間の平均値を取って共有の潜在空間として求めていることが原因として考えられる．

9　まとめと今後の課題

本研究は，ガウス過程に従う高次元の時系列データを対象にし，GPLVM を用いて低次元の潜在空間に圧縮する手法について考察した．また，従来の手法を効率と精度の 2 つの面で改良する手法として，我々は EM アルゴリズムを使用する代わりに，非線形最適化手法である多層パーセプトロンを導入する手法を提案した．さらに，複数の時系列データの対応関係を取得することが出来る sharedGPLVM にも本手法を導入し，sharedGPLVM においても精度よく効率的な潜在空間の推定が出来ることを確認した．GPLVM，sharedGPLVM を改良した技術である潜在空間上でのダイナミクスを考慮した圧縮が可能な GPDM や sharedGPDM にも本手法を導入し，実用性を調査した．本手法は，GPLVM や sharedGPLVM，GPDM，sharedGPDM のような代表的なガウス過程に基づく次元圧縮手法を精度，効率の面で向上させることに成功した．本手法を sharedGPDM に組み込み圧縮を行ったところ，ダイナミクスを考慮した圧縮に成功したが，成功率は決して高いものではなく，マッピングを行った場合に失敗する例も散見した．今後，圧縮の更なる精度の向上を目指し本手法の改良を行い，新たな観測空間が入力された際に潜在空間を通じて他方の観測空間への出力を行う．

参考文献

[1] Neil D.Lawrence, “Gaussian Process Latent Variable Models for Visualisation of High Dimensional Data”, NIPS, 2004.

[2] Wang, J. M., Fleet, D. J., Hertzmann, A. “Gaussian Process Dynamical Models”, In Proc. NIPS 2005. Vancouver, Canada. pp. 1441-1448., December, 2005.

[3] Wang, J. M., Fleet, D. J., Hertzmann, A. “Gaussian Process Dynamical Models for Human Motion”, In IEEE Transactions on Pattern Recognition and Machine Intelligence, pp. 283-298.,February, 2008.

[4] Aaron P. Shon,Keith Grochow, Rajesh P.N. Rao, “Robotic Imitation from Human Motion Capture using Gaussian Processes”, 5th IEEE-RAS International Conference on Humanoid Robots, 2005.

[5] Shon, A., Grochow, K., Hertzmann, A., Rao, R., “Learning shared latent structure for image synthesis and robotic imitation”, Advances in NIPS 18, 2006.

[6] Carl Henrik Ek,, “Shared Gaussian Process Latent Variables Models”, Ph.D thesis, Oxford Brookes University, 2009.

[7] Katsu Yamane, Yuka Ariki,and Jessica Hodgins, “Animating Non-Humanoid Characters with Human Motion Data”, Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 169–178, 2010.

[8] S. Eleftheriadis, O. Rudovic, M. Pantic. “Shared Gaussian Process Latent Variable Model for Multiview Facial Expression Recognition”, Advances in Visual Computing, 9th International Symposium, ISVC. Springer Berlin Heidelberg, pp. 527 - 538, 2013.

[9] S. Eleftheriadis, O. Rudovic, M. Pantic. “View-constrained Latent Variable Model for Multi-view Facial Expression Classification”, Int’l Symposium on Visual Computing (ISVC’14), LNCS, Las Vegas, USA. pp. 292 - 303, December 2014.

[10] S. Eleftheriadis, O. Rudovic, M. Pantic., “Multi-conditional Latent Variable Model for Joint Facial Action Unit Detection”, International Conference on Computer Vision (ICCV). Santiago, Chile, December 2015.

[11] Jixu Chen, Minyoung Kim, Yu Wang, Qiang Ji, “Switching Gaussian Process Dynamic Models for Simultaneous Composite Motion Tracking and Recognition”, 2009 IEEE Conference on Computer Vision and Pattern Recognition, Jun. 2009.