

共有する潜在空間を用いた異なる時系列データの対応関係学習に関する取り組み

A Study on Learning the Corresponding Relation between Different Time-series Data Using Shared Latent Space

大山 まりほ^{*1} 小林 一郎^{*2}
 Mariho Ohyama Ichiro Kobayashi

^{*1}お茶の水女子大学大学院 ^{*2}お茶の水女子大学基幹研究院
 Ochanomizu University Ochanomizu University

In this research, we focus on high dimensional time-series data following Gaussian process. First, we consider the dimensional compression in the latent space using Gaussian Process Latent Variable Models, and to improve the accuracy of the conventional linear identification, we introduced a method that uses Multilayer perceptron(MLP) which can identify nonlinearly. Next, we introduced Shared Gaussian Process Latent Variable Models that can capture correspondence of multiple time-series data. However, because the dynamics of time-series data in the latent space is not taken into consideration when performing dimensional compression, it is difficult to compact accurately into the latent space. In this research, we propose a method applying the optimization method of MLP to Gaussian Process Dynamical Models, which takes the data in the latent space as Gaussian process.

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

観測データ全体を \mathbf{Y} , 圧縮先の潜在空間上のデータ全体を \mathbf{X} とする. まず Principal component analysis(PCA) を用いて \mathbf{Y} を圧縮し, \mathbf{X} の初期値を設定する. そして \mathbf{X}, \mathbf{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 $p(\mathbf{Y}|\mathbf{X}, \alpha, \beta, \gamma)$ 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, \mathbf{X} を再設定する.

$$p(\mathbf{Y}|\mathbf{X}, \alpha, \beta, \gamma) = \frac{1}{(2\pi)^{DN/2} |\mathbf{K}|^{D/2}} \times \exp\left(-\frac{1}{2} \text{tr}(\mathbf{K}^{-1} \mathbf{Y} \mathbf{Y}^T)\right) \quad (1)$$

$$k(\mathbf{x}_n, \mathbf{x}_m) = \alpha \exp\left(-\frac{\gamma}{2} (\mathbf{x}_n - \mathbf{x}_m)^T (\mathbf{x}_n - \mathbf{x}_m)\right) + \delta_{nm} \beta^{-1} \quad (2)$$

カーネルのパラメータの更新を T 回繰り返すことで \mathbf{X} が更新され, 高次元の \mathbf{Y} に対して低次元の潜在空間 \mathbf{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 \mathbf{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 \mathbf{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:** $\mathbf{X}, \mathbf{Y}, \mathbf{K}$ Initialize \mathbf{X} through PCA.**while** Gradient Eq.(3) $\leq \epsilon$ **do**Select a new \mathbf{X} using the parameters of \mathbf{K} through feed forward MLP.Optimise Eq.(3) with respect to the parameters of \mathbf{K} and the synaptic weights of MLP using scaled conjugate gradients.**end while**

し、 \mathbf{X} の初期値を設定する。そして \mathbf{X} , \mathbf{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。 \mathbf{Y} を入力、 \mathbf{X} を出力として結合荷重を求め、 \mathbf{X} と \mathbf{Y} の対応関係を学習させておく。その後、式 (3) で表される $\log p(\mathbf{Y}, \mathbf{X}; \alpha, \beta, \gamma)$ 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 \mathbf{w} の更新を行い、 \mathbf{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 \mathbf{X} が更新され、高次元の \mathbf{Y} に対して低次元の潜在空間 \mathbf{X} が求まる。

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

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[3] 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 \mathbf{Y} を作成する。20 次元の観測データ \mathbf{Y} を 3 次元の潜在空間 \mathbf{X} に圧縮する。作成した \mathbf{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。

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

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

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

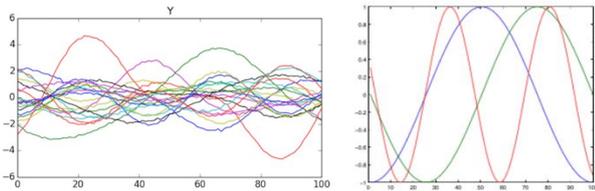


図 1: 左: 観測データ \mathbf{Y} 右: \mathbf{Y} の基底 (式 (4)~(6))

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。

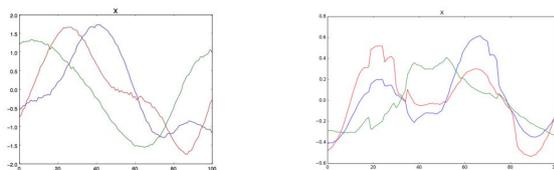


図 2: 左: MLP 有 右: MLP 無

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 \mathbf{K} のパラメータを求める際の \mathbf{X} の初期値および繰り返しの回数によって得られる \mathbf{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。一方で、潜在空間上での表現において結果は初期値に依存するため、同じ観測データを複数回圧縮した際その出力値には再現性がないことも分かり、潜在空間上での複数の時系列データの対応関係の取得は容易ではないことが分かった。

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[3] 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。

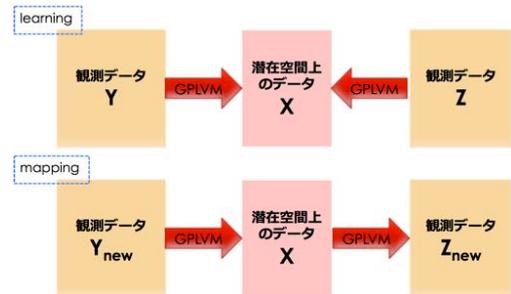


図 3: SharedGPLVM の概要図

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。GPLVM と同様に MLP を組み込んだ学習アルゴリズムを Algorithm 3 に示す。

まず PCA を用いて観測空間 \mathbf{Y} と \mathbf{Z} を圧縮し、 \mathbf{X}_Y と \mathbf{X}_Z をそれぞれ求める。それらの平均を取り \mathbf{X} の初期値を設定する。そして \mathbf{X} , \mathbf{Y} , \mathbf{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。 \mathbf{Y} と \mathbf{Z} を入力、 \mathbf{X} を出力としてそれぞれ結合荷重を求め、 \mathbf{X} と \mathbf{Y} , \mathbf{X} と \mathbf{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 $\log p(\mathbf{Y}, \mathbf{Z}, \mathbf{X}; \phi_Y, \phi_Z)$ が最大となるよう、 \mathbf{Y} , \mathbf{Z} それぞれのカーネル関数のパラメータ ϕ_Y, ϕ_Z と MLP の結合荷重 $\mathbf{w}_Y, \mathbf{w}_Z$ の更新を行い、 \mathbf{X}_Y と \mathbf{X}_Z を再設定し、再びそれらの平均をとることで \mathbf{X} を再設定する。勾配が収束するまで繰り返すことで \mathbf{X} が更新され、高次元の \mathbf{Y} , \mathbf{Z} に対して低次

*1 <http://github.com/jameshensman>

Algorithm 3 SharedGPLVM with MLP:learning

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

元の共有の潜在空間 \mathbf{X} が求まる.

$$\log p(\mathbf{Y}, \mathbf{Z}, \mathbf{X}; \phi_Y, \phi_Z) = \log p(\mathbf{Y}, \mathbf{Z}|\mathbf{X}, \phi_Y, \phi_Z) + \frac{1}{2} \sum_i \mathbf{X}_i^2 + \frac{1}{2} \sum_j \mathbf{w}_{Yj}^2 + \frac{1}{2} \sum_k \mathbf{w}_{Zk}^2 \quad (7)$$

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

$$P(\mathbf{Y}|\mathbf{X}, \phi_Y) = \frac{1}{\sqrt{(2\pi)^{ND_Y} |\mathbf{K}_Y|^{D_Y}}} \times \exp\left(-\frac{1}{2} \sum_{k=1}^{D_Y} \mathbf{y}_k^T \mathbf{K}_Y^{-1} \mathbf{y}_k\right)$$

$$P(\mathbf{Z}|\mathbf{X}, \phi_Z) = \frac{1}{\sqrt{(2\pi)^{ND_Z} |\mathbf{K}_Z|^{D_Z}}} \times \exp\left(-\frac{1}{2} \sum_{k=1}^{D_Z} \mathbf{z}_k^T \mathbf{K}_Z^{-1} \mathbf{z}_k\right)$$

4.2 マッピング

新たな高次元の観測データ \mathbf{Y}_{new} が入力された際、潜在空間 \mathbf{X} を通じて他の高次元の \mathbf{Z}_{new} を出力する. \mathbf{Y}_{new} の \mathbf{y}_{new} に対応する潜在空間 \mathbf{X} 上の点 \mathbf{x}_{new} から、以下の式 (8) で表される \mathbf{z}_{new} の確率分布が得られる. $k(\mathbf{x}_{new})$ は i 行目が $k_i(\mathbf{x}_{new}) = k(\mathbf{x}_{new}, \mathbf{x}_i)$ 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.

$$\begin{aligned} \bar{\mathbf{z}}(\mathbf{x}_{new}) &= \boldsymbol{\mu}_Z + \mathbf{Z}^T \mathbf{K}_Z^{-1} k(\mathbf{x}_{new}) \\ \sigma_Z^2(\mathbf{x}_{new}) &= k(\mathbf{x}_{new}, \mathbf{x}_{new}) \\ &\quad - k(\mathbf{x}_{new})^T \mathbf{K}_Z^{-1} k(\mathbf{x}_{new}) \end{aligned} \quad (8)$$

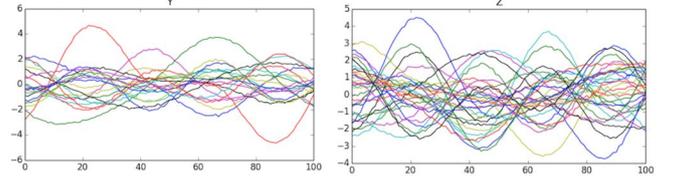
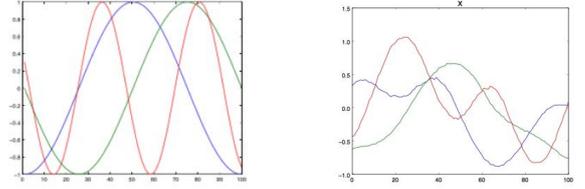
5. 実験: SharedGPLVM

5.1 学習

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

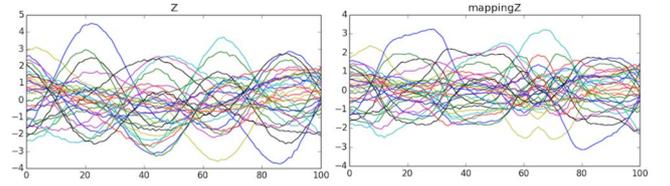
Algorithm 4 SharedGPLVM with MLP:mapping

Require: $\mathbf{X}_{new}, \mathbf{Z}, \mathbf{K}_Z$ **for** \mathbf{X}_{new} **do**Calculate Eq.(8) using \mathbf{Z}, \mathbf{K}_Z .**end for**

図 4: 観測データ (左: \mathbf{Y} 右: \mathbf{Z})図 5: 左: \mathbf{Y}, \mathbf{Z} の基底 右: 共有の潜在空間 \mathbf{X}

5.2 マッピング

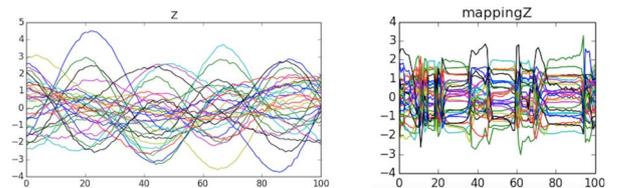
学習で生成された \mathbf{X} を \mathbf{X}_{new} とし、マッピングを行った. 結果として \mathbf{Z} が復元されるのが理想である. 結果は図 6 の右図のようになった.

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

5.3 学習+マッピング

マッピングの実験では \mathbf{X} を \mathbf{X}_{new} とし行っていたため、図 3 にあるように観測データ \mathbf{Y}_{new} の潜在空間上のデータを通じて \mathbf{Z}_{new} を出力する. \mathbf{Y}_{new} の潜在空間上のデータは式 (9) で得られる. 今回は \mathbf{Y} を \mathbf{Y}_{new} として実験を行った. 結果は図 7 のようになった.

$$\hat{\mathbf{x}} = \arg \max_x P(\mathbf{y}_{new} | \mathbf{x}, \mathbf{Y}, \mathbf{X}, \phi_Y) \quad (9)$$

図 7: 左: \mathbf{Z} (オリジナル) 右: マッピング \mathbf{Z}

5.4 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 \mathbf{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。しかし、一方の観測データから潜在空間を通して復元を行うと形が崩れてしまった。したがって、GPLVMを用いて複数の時系列データを共有の潜在空間へ圧縮する手法では高次元のデータの中の重要な要素を保持出来ない可能性が高いことが分かった。

6. GPDM

GPLVMでは潜在空間 \mathbf{X} でのダイナミクスについては考慮されていないため、潜在空間のダイナミクスを扱うことが出来るGPDM[5]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Algorithm4に示す。 $\phi_{\mathbf{X}}, \phi_{\mathbf{Y}}$ はいずれも \mathbf{X}, \mathbf{Y} のカーネル関数のパラメータである。 \mathbf{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、 $\mathbf{W} \equiv \text{diag}(w_1, \dots, w_{D_{\mathbf{Y}}})$ で表される。

Algorithm 5 GPDM with MLP

Require: $\mathbf{X}, \mathbf{Y}, \mathbf{K}_{\mathbf{X}}, \mathbf{K}_{\mathbf{Y}}$

Initialize \mathbf{X} through PCA.

while Gradient log Eq.(10) $\leq \epsilon$ **do**

Select a new \mathbf{X} using the parameters of $\mathbf{K}_{\mathbf{X}}$ and $\mathbf{K}_{\mathbf{Y}}$ through feed forward MLP.

Optimise $\log p(\mathbf{Y}, \mathbf{X}, \phi_{\mathbf{Y}}, \phi_{\mathbf{X}})$ with respect to the parameters of $\mathbf{K}_{\mathbf{X}}$ and $\mathbf{K}_{\mathbf{Y}}$, the synaptic weights of MLP using scaled conjugate gradients.

for $k = 1$ to $D_{\mathbf{Y}}$ **do**

$$w_k \leftarrow \sqrt{N(\mathbf{y}_{:,k}^T \mathbf{K}_{\mathbf{Y}}^{-1} \mathbf{y}_{:,k})^{-1}}$$

end for

end while

PCAを用いて観測空間 \mathbf{Y} を圧縮し、 \mathbf{X} を求める。 \mathbf{X}, \mathbf{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。 \mathbf{Y} と \mathbf{Z} を入力、 \mathbf{X} を出力としてそれぞれ結合荷重を求め、 \mathbf{X} と \mathbf{Y} 、 \mathbf{X} と \mathbf{Z} の対応関係を学習させておく。その後、以下の式(10)で表される $p(\mathbf{Y}, \mathbf{X}, \phi_{\mathbf{Y}}, \phi_{\mathbf{X}})$ が最大となるよう、 \mathbf{Y}, \mathbf{X} それぞれのカーネル関数のパラメータ $\phi_{\mathbf{Y}}, \phi_{\mathbf{X}}$ とMLPの結合荷重 $\mathbf{w}_{\mathbf{Y}}$ の更新を行い、 \mathbf{X} を再設定する。勾配が収束するまで繰り返すことで \mathbf{X} が更新され、高次元の \mathbf{Y} に対して最適な低次元の共有の潜在空間 \mathbf{X} が求まる。 \mathbf{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(11))。

$$\begin{aligned} p(\mathbf{Y}, \mathbf{X}, \phi_{\mathbf{Y}}, \phi_{\mathbf{X}}) &= p(\mathbf{Y}|\mathbf{X}, \phi_{\mathbf{Y}})p(\mathbf{X}|\phi_{\mathbf{X}})p(\phi_{\mathbf{X}})p(\phi_{\mathbf{Y}}) \\ &= \frac{|\mathbf{W}|^N}{(2\pi)^{D_{\mathbf{Y}}N/2} |\mathbf{K}_{\mathbf{Y}}|^{D_{\mathbf{Y}}/2}} \exp\left(-\frac{1}{2} \text{tr}(\mathbf{K}_{\mathbf{Y}}^{-1} \mathbf{Y} \mathbf{W}^2 \mathbf{Y}^T)\right) \\ &\times p(x_1) \frac{1}{(2\pi)^{D_{\mathbf{X}}(N-1)/2} |\mathbf{K}_{\mathbf{X}}|^{D_{\mathbf{X}}/2}} \exp\left(-\frac{1}{2} \text{tr}(\mathbf{K}_{\mathbf{X}}^{-1} \mathbf{X}_{out} \mathbf{X}_{out}^T)\right) \\ &\times p(\phi_{\mathbf{X}})p(\phi_{\mathbf{Y}}) \end{aligned} \quad (10)$$

$$\begin{aligned} k(x_n, x_m) &= \alpha_{\mathbf{X}} \exp\left(-\frac{\gamma_{\mathbf{X}}}{2} (x_n - x_m)^T (x_n - x_m)\right) \\ &+ \delta_{nm} \beta_{\mathbf{X}}^{-1} + \eta_{\mathbf{X}} x_n \cdot x_m \end{aligned} \quad (11)$$

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 \mathbf{Y} をGPDMを用いて3次元に圧縮した。結果を図8に示す。

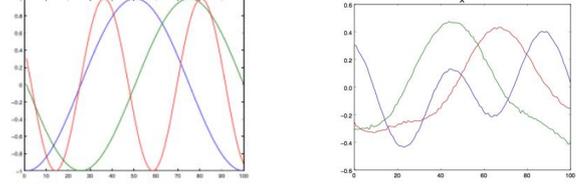


図8: 左: \mathbf{Y}, \mathbf{Z} の基底 右: 共有の潜在空間 \mathbf{X}

6.2 考察

GPDMを用いて高次元の時系列データ \mathbf{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。

7. まとめと今後の課題

本研究においては、共有する潜在空間上で人の動作識別への取り組みを行った。GPLVMを用いた時系列データの次元圧縮のサンプルプログラムを作成しMLPを組み込んだ場合との結果の比較を行ったところ、MLPを組み込むことで精度が向上することが分かった。また、GPLVMでは潜在データの初期値によって結果が大きく異なることを確認した。そこで、2つの観測空間を共有の潜在空間に圧縮するSharedGPLVMを用い、潜在空間への次元圧縮、及び潜在空間から観測空間へのマッピングを行うプログラムを作成したが、潜在空間内での系列を捉えていないため観測データの対応関係を明確に捉えることが出来なかった。そこで、潜在空間のダイナミクスを考慮するGPDMにMLPを組み込み次元圧縮を行ったところ、潜在空間上でもガウス過程に従う圧縮をすることが出来た。今後の課題として、GPDMを用いた2つの時系列データの対応関係を学習し、2つの観測空間から共有のダイナミクス性のある潜在空間に圧縮するsharedGPDMを作成する。

参考文献

- [1] Neil D.Lawrence“Gaussian Process Latent Variable Models for Visualisation of High Dimensional Data”, 2004.
- [2] Aaron P. Shon, Keith Grochow, Rajesh P.N. Rao“Robotic Imitation from Human Motion Capture using Gaussian Processes”, 2005.
- [3] Carl Henrik Ek“Shared Gaussian Process Latent Variables Models”, 2009.
- [4] Katsu Yamane, Yuka Ariki, and Jessica Hodgins“Animating Non-Humanoid Characters with Human Motion Data”, 2010.
- [5] J.Chen, “Switching Gaussian Process Dynamic Models for Simultaneous Composite Motion Tracking and Recognition”, IEEE 2009.