# *Introduction and Advances in Gaussian Processes*

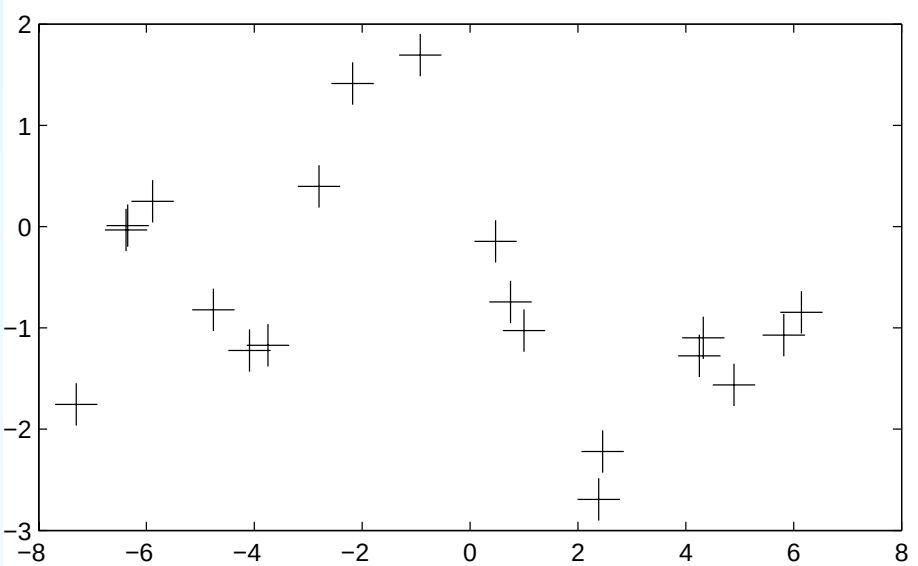
持橋大地
NTT コミュニケーション科学基礎研究所
*daichi@cslab.kecl.ntt.co.jp*

SVM 2009
Modified for NTT 厚木基礎研
2009–10–13 (Tue)

# 流れ

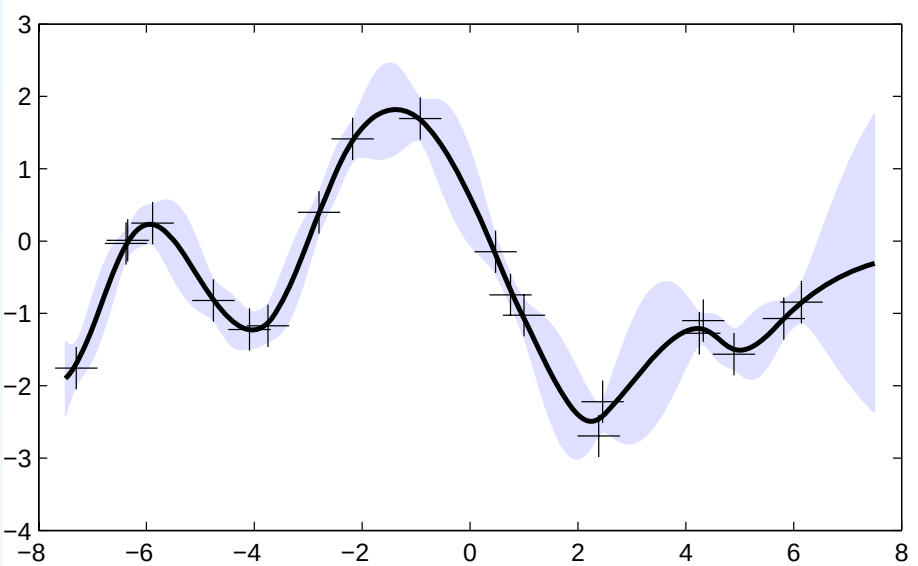
- Gaussian Process とは?
  - SVM との関係
  - Dirichlet Process との関係
  - 例と簡単な応用
- Gaussian Process Latent Variable Models (Lawrence 2003)
- Gaussian Process Dynamical Models (Hertzmann+ 2005)
- Gaussian Process Density Sampler (Adams+ ICML 2008)
- Discussion
- Literature and Readings

# Gaussian Process とは



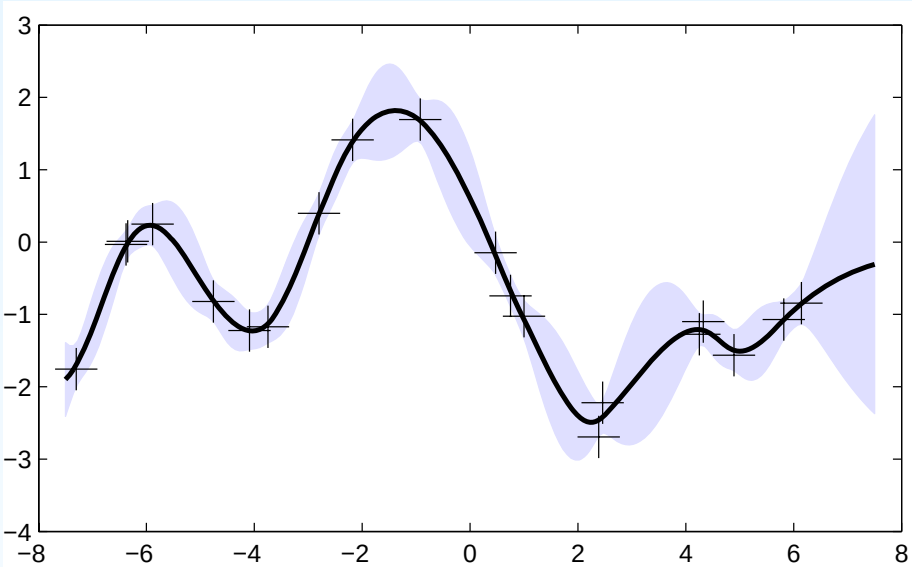

- 入力 $\mathbf{x} \to y$ を予測する回帰関数 (regressor) の確率モデル
  - データ $D = \left\{(\mathbf{x}^{(n)}, y^{(n)})\right\}_{n=1}^{N}$ が与えられた時, 新しい $\mathbf{x}^{(n+1)}$ に対する $y^{(n+1)}$ を予測
  - ランダムな関数の確率分布
  - 連続空間で動く, ベイズ的なカーネルマシン (後で)

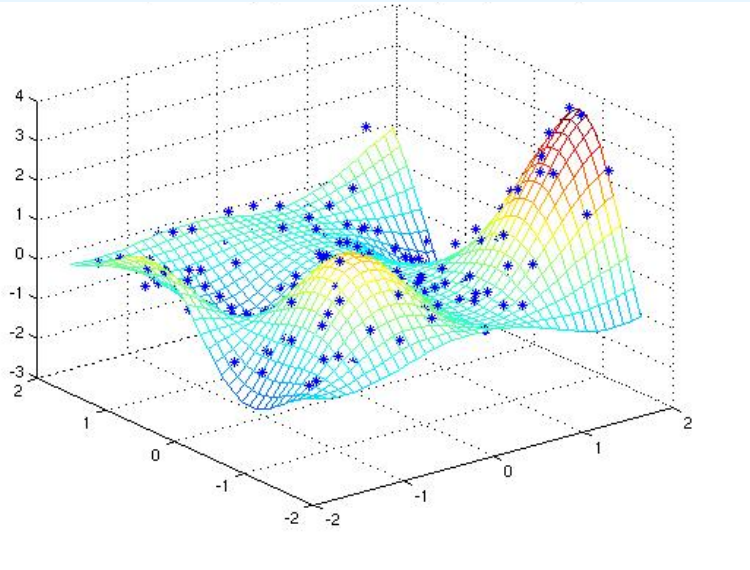
# Gaussian Process とは




- 入力 $\mathbf{x} \to y$ を予測する回帰関数 (regressor) の確率モデル
  - データ $D = \left\{(\mathbf{x}^{(n)}, y^{(n)})\right\}_{n=1}^{N}$ が与えられた時, 新しい $\mathbf{x}^{(n+1)}$ に対する $y^{(n+1)}$ を予測
  - ランダムな関数の確率分布
  - 連続空間で動く, ベイズ的なカーネルマシン (後で)

# Gaussian Process とは




- 入力 $\mathbf{x} \to y$ を予測する回帰関数 (regressor) の確率モデル
  - データ $D = \left\{(\mathbf{x}^{(n)}, y^{(n)})\right\}_{n=1}^{N}$ が与えられた時, 新しい $\mathbf{x}^{(n+1)}$ に対する $y^{(n+1)}$ を予測
  - ランダムな関数の確率分布
  - 連続空間で動く, ベイズ的なカーネルマシン (後で)

# 線形回帰モデル

- 入力 $\mathbf{x}$ から出力 $y \in \mathbb{R}$ を予測する回帰関数 $y = f(\mathbf{x})$ を求めたい
  - $\mathbf{x} = (x_1, \cdots, x_d) \in \mathbb{R}^d$ は時間や任意のベクトル
- $y = f(\mathbf{x})$ を, $\mathbf{x}$ を一般の関数 $\phi(\mathbf{x})$ で変換した上で線形モデルで表してみる

$$y = \mathbf{w}^T \phi(\mathbf{x}) \tag{1}$$

  - **例:** $\phi(\mathbf{x}) = (1, x_1, \cdots, x_d, x_1^2, \cdots, x_d^2)$

    $\mathbf{w} = (w_0, w_1, \cdots, w_d, w_{d+1}, \cdots, w_{2d})$ とおくと,

$$\begin{aligned} y &= \mathbf{w}^T \phi(\mathbf{x}) \\ &= w_0 + w_1 x_1 + \cdots + w_d x_d \\ &\quad + w_{d+1} x_1^2 + \cdots + w_{2d} x_d^2 \, . \end{aligned}$$

  - 一般に, 基底関数 $\phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \cdots, \phi_H(\mathbf{x}))$

# GP の導入 (1)

- $y^{(1)} \dots y^{(N)}$ について同時に書くと, 下のように $\mathbf{y} = \mathbf{\Phi w}$ と行列形式で書ける ($\mathbf{\Phi}$ : 計画行列)

$$\begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{pmatrix} = \begin{pmatrix} \phi_1(\mathbf{x}^{(1)}) & \cdots & \phi_H(\mathbf{x}^{(1)}) \\ \phi_1(\mathbf{x}^{(2)}) & \cdots & \phi_H(\mathbf{x}^{(2)}) \\ \vdots & & \vdots \\ \phi_1(\mathbf{x}^{(N)}) & \cdots & \phi_H(\mathbf{x}^{(N)}) \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ \vdots \\ w_H \end{pmatrix} \qquad (2)$$

$$\mathbf{y} \qquad\qquad \mathbf{\Phi} \qquad\qquad \mathbf{w}$$

- 重み $\mathbf{w}$ がガウス分布 $p(\mathbf{w}) = \mathrm{N}(\mathbf{0}, \alpha^{-1}\mathbf{I})$ に従っているとすると, $\mathbf{y} = \mathbf{\Phi w}$ もガウス分布に従い,
- 平均 0, 分散

$$\langle \mathbf{y}\mathbf{y}^T \rangle = \left\langle (\mathbf{\Phi w})(\mathbf{\Phi w})^T \right\rangle = \mathbf{\Phi} \langle \mathbf{w}\mathbf{w}^T \rangle \mathbf{\Phi}^T \qquad (3)$$

$$= \alpha^{-1}\mathbf{\Phi}\mathbf{\Phi}^T \qquad \text{の正規分布となる}$$

# GP の導入 (2)

$$p(\mathbf{y}) = \mathrm{N}(\mathbf{y}|\mathbf{0}, \alpha^{-1}\mathbf{\Phi}\mathbf{\Phi}^T) \tag{4}$$

は, どんな入力 $\{\mathbf{x}_n\}_{n=1}^N$ についても成り立つ → ガウス過程の定義

- どんな入力 $(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N)$ についても, 対応する出力 $\mathbf{y} = (y_1, y_2, \cdots, y_N)$ がガウス分布に従うとき, $p(\mathbf{y})$ はガウス過程に従う　という.
  - ガウス過程 = 無限次元のガウス分布
  - ガウス分布の周辺化はまたガウス分布なので, 実際にはデータのある所だけの有限次元
- $\mathbf{K} = \alpha^{-1}\mathbf{\Phi}\mathbf{\Phi}^T$ の要素であるカーネル関数

$$k(\mathbf{x}, \mathbf{x}') = \alpha^{-1}\phi(\mathbf{x})^T\phi(\mathbf{x}') \tag{5}$$

だけでガウス分布が定まる
  - $k(\mathbf{x}, \mathbf{x}')$ は $\mathbf{x}$ と $\mathbf{x}'$ の距離 ； 入力 $\mathbf{x}$ が近い→出力 $y$ が近い

# GP の導入 (3)

- 実際には, 観測値にはノイズ $\epsilon$ が乗っている

$$\begin{cases} y = \mathbf{w}^T \phi(\mathbf{x}) + \epsilon \\ \epsilon \sim \mathrm{N}(0, \beta^{-1}\mathbf{I}) \end{cases} \quad \Longrightarrow \quad p(y|f) = \mathrm{N}(\mathbf{w}^T \phi(\mathbf{x}), \beta^{-1}\mathbf{I}) \tag{6}$$

- 途中の $f = \mathbf{w}^T \phi(\mathbf{x})$ を積分消去

$$p(y|\mathbf{x}) = \int p(y|f)p(f|\mathbf{x})df \tag{7}$$

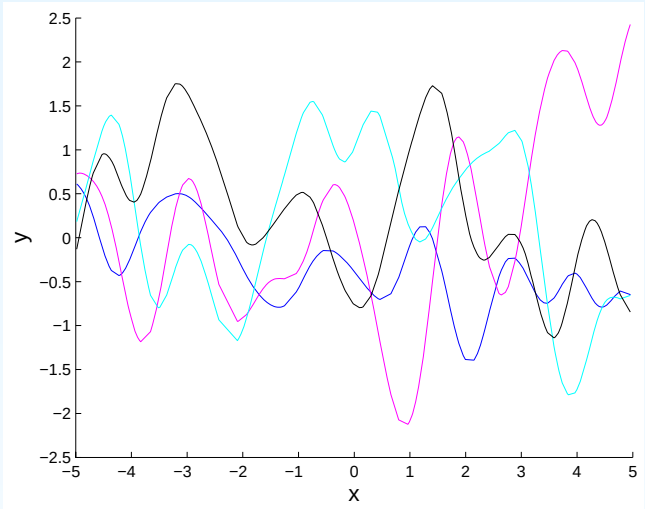
$$= \mathrm{N}(0, \mathbf{C}) \tag{8}$$

  - 二つの独立な Gaussian の畳み込みなので, $\mathbf{C}$ の要素は共分散の単純な和:

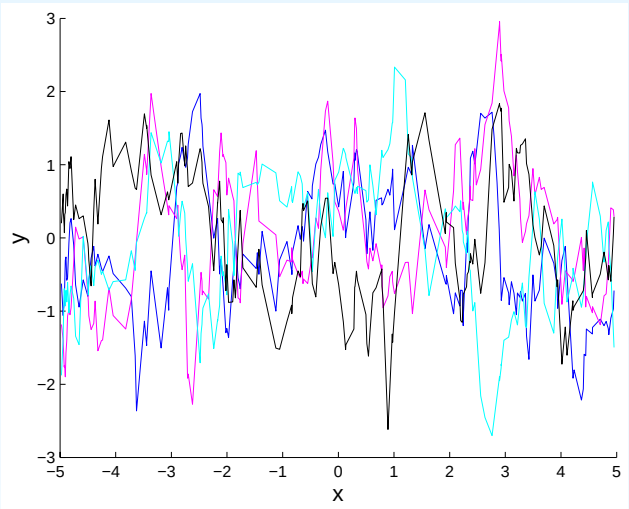
$$C(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j) + \beta^{-1}\delta(i, j). \tag{9}$$

  - GP は, カーネル関数 $k(\mathbf{x}, \mathbf{x}')$ とハイパーパラメータ $\alpha, \beta$ だけで表すことができる.

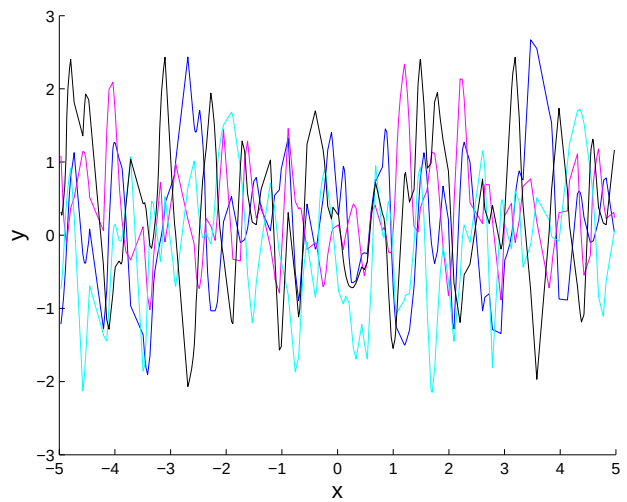
# 様々なカーネル



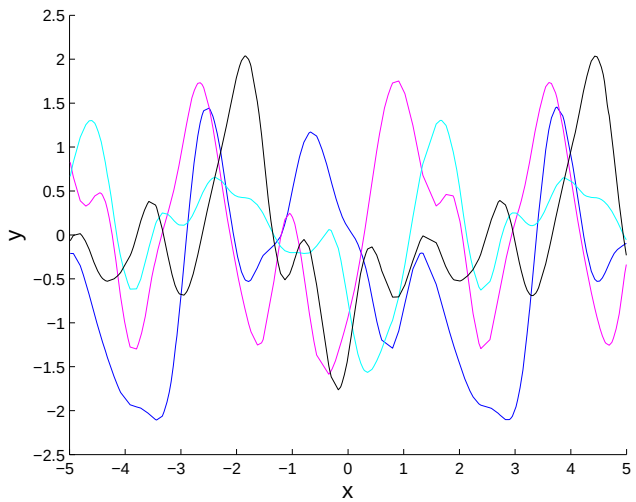

Gaussian: $\exp(-(\mathbf{x}-\mathbf{x}')^2/l)$




Exponential: $\exp(-|\mathbf{x}-\mathbf{x}'|/l)$ (OU process)




Periodic: $\exp(-2\sin^2(\frac{\mathbf{x}-\mathbf{x}'}{2})/l^2)$




Periodic(L): $\exp(-2\sin^2(\frac{\mathbf{x}-\mathbf{x}'}{2})/(10l)^2)$

# 「基底関数」の消去

- RBF 基底関数 $\phi(\mathbf{x}) = \exp((\mathbf{x} - \mathbf{h})^2/r^2)$ を考えてみる
- 1 次元の場合, $h$ を無限個用意すると

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \sum_{h=1}^{H} \phi_h(\mathbf{x})\phi_h(\mathbf{x}') \tag{10}$$

$$\rightarrow \int_{-\infty}^{\infty} \exp\left(-\frac{(x-h)^2}{r^2}\right) \exp\left(-\frac{(x'-h)^2}{r^2}\right) dh \tag{11}$$

$$= \sqrt{\pi r^2} \exp\left(-\frac{(x-x')^2}{2r^2}\right) \equiv \theta_1 \exp\left(-\frac{(x-x')^2}{\theta_2^2}\right) \tag{12}$$

  - $(\mathbf{x}, \mathbf{x}')$ の RBF カーネルは, 無限個の RBF 基底関数を考えたことと等価.
  - カーネルのパラメータ $\theta_1, \theta_2$ は最尤推定で最適化できる

# GP の予測

- 新しい入力 $y^{\text{new}}$ とこれまでの $\mathbf{y}$ の結合分布がまた Gaussian になるので,

$$p(y^{\text{new}}|\mathbf{x}^{\text{new}}, \mathbf{X}, \mathbf{y}, \theta) \tag{13}$$

$$= \frac{p((\mathbf{y}, y^{\text{new}})|(\mathbf{X}, \mathbf{x}^{\text{new}}), \theta)}{p(\mathbf{y}|\mathbf{X}, \theta)} \tag{14}$$

$$\propto \exp\left(-\frac{1}{2}([\mathbf{y}, y^{\text{new}}]\begin{bmatrix}\mathbf{K} & \mathbf{k}\\ \mathbf{k}^T & k\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{y}\\ y^{\text{new}}\end{bmatrix} - \mathbf{y}^T\mathbf{K}^{-1}\mathbf{y})\right) \tag{15}$$

$$\sim \mathrm{N}(\mathbf{k}^T\mathbf{K}^{-1}\mathbf{y}, k - \mathbf{k}^T\mathbf{K}^{-1}\mathbf{k}). \tag{16}$$

ここで

- $\mathbf{K} = [k(\mathbf{x}, \mathbf{x}')]$, $\mathbf{k} = (k(\mathbf{x}^{\text{new}}, \mathbf{x}_1), \cdots, k(\mathbf{x}^{\text{new}}, \mathbf{x}_N))$,
- $k = k(\mathbf{x}^{\text{new}}, \mathbf{x}^{\text{new}}) + \beta^{-1}$.

# GP の適用例

- 『ガウシアンプロセスによる名演奏の学習』寺村&前田+, IPSJ-MUS 2008 & ICMPC 2008.
  - 楽譜情報 $\mathbf{x}$ →実際の演奏 $y$ の対応を学習
  - 実際には,
    - $\mathbf{x}$: 楽譜からの素性 (拍子, 音高, 音長, 強弱指定, 主旋律か, ⋯)
    - $y$: 楽譜の指定と実際の出力との差 (アタック, リリース, 強弱の 3 種類の $y$)
  - $\{\mathbf{x}^{(n)}, y^{(n)}\}_{n=1}^{N}$ のセットから対応関係を学習
- 学習のポイント: $(\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(N)}) \rightarrow (y^{(1)}, \cdots, y^{(N)})$ への対応を「まとめて」学習していること
  - カーネルは

$$k(\mathbf{x}, \mathbf{x}') = \frac{a}{(1 + (\mathbf{x} - \mathbf{x}')^T B (\mathbf{x} - \mathbf{x}'))^c}$$

  - ただし, $(\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(N)})$ が時系列という仮定は無いのに注意

# GP による分類と SVM

- $y = \{+1, -1\}$ のとき, $p(y|f) = \sigma(y \cdot f)$ (logit) または $\Psi(y \cdot f)$ (probit) で分類器になる

$$\text{minimize: } -\log p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{X})$$

$$= \frac{1}{2}\mathbf{f}^T K^{-1}\mathbf{f} - \sum_{i=1}^{N} \log p(y_i|f_i) \qquad (17)$$

- ソフトマージン SVM では $K\alpha = \mathbf{f}$ とすると,

$$\mathbf{w} = \sum_i \alpha_i \mathbf{x}_i \;\rightarrow\; |\mathbf{w}|^2 = \alpha^T K \alpha = \mathbf{f}^T K^{-1}\mathbf{f} \;\text{ ゆえ,}$$

$$\text{minimize: } \frac{1}{2}|\mathbf{w}|^2 - C\sum_{i=1}^{N}(1 - y_i f_i)_+$$

$$= \frac{1}{2}\mathbf{f}^T K^{-1}\mathbf{f} - C\sum_{i=1}^{N}(1 - y_i f_i)_+ \,. \qquad (18)$$

  - そっくりだが, SVM は hinge loss なのでスパース.

# Loss functions



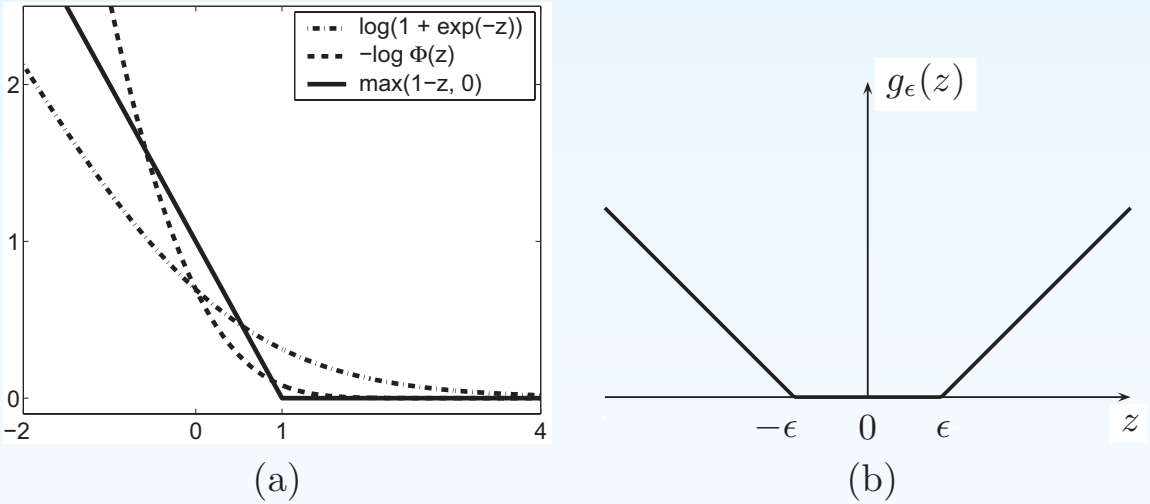
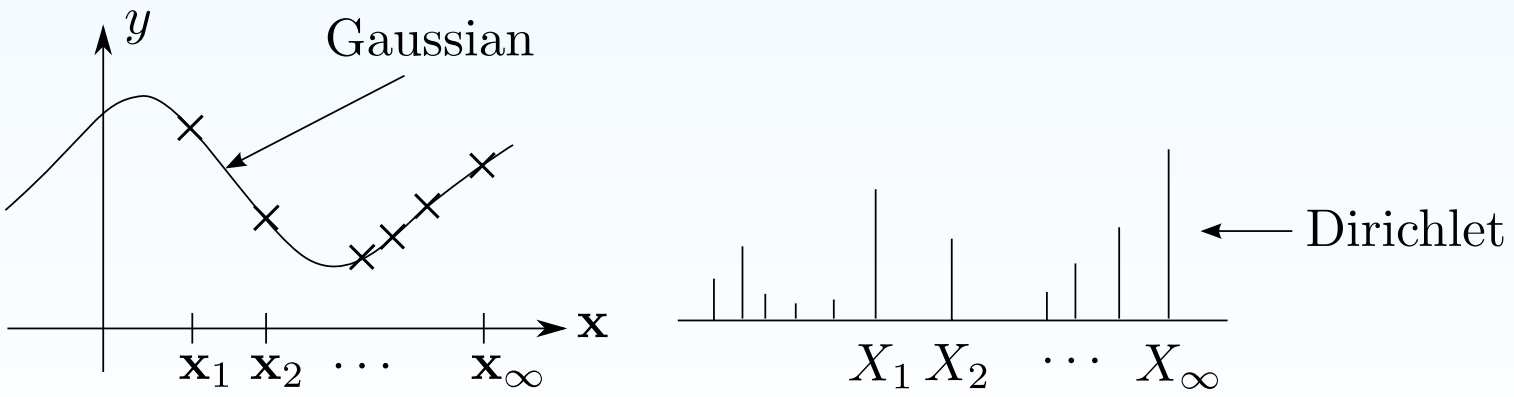



Figure 6.3: (a) A comparison of the hinge error, $g_\lambda$ and $g_\Phi$. (b) The $\epsilon$-insensitive error function used in SVR.

- お馴染みの議論かも
  - SVM と ME の関係についても, 以前工藤君がそっくりの議論をしていた
- CS 研での議論: 分類したいだけなら, GP classifier は回りくどすぎる (他の理由があるなら有効)

# DP との関係

- Gaussian process と Dirichlet process の定義はそっくり [偶然ではない]
  - **GP:** どんな $(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_\infty)$ をとってきても, 対応する $(y_1, y_2, \cdots, y_\infty)$ がガウス分布に従う
  - **DP:** 空間のどんな離散化 $(X_1, X_2, \cdots, X_\infty)$ についても, 対応する離散分布がディリクレ分布 $\mathrm{Dir}(\alpha(X_1), \alpha(X_2), \cdots, \alpha(X_\infty))$ に従う

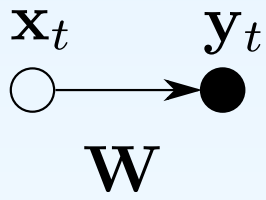



- どちらも, 無限次元の smoother になっている

# Gaussian Process Latent Variable Models (GPLVM)

- Probabilistic PCA (Tipping&Bishop 1999):

$$p(\mathbf{y}_n|\mathbf{W},\beta) = \int p(\mathbf{y}_n|\mathbf{x}_n,\mathbf{W},\beta)p(\mathbf{x}_n)d\mathbf{x}_n \quad (19)$$




$$p(\mathbf{Y}|\mathbf{W},\beta) = \prod_n p(\mathbf{y}_n|\mathbf{W},\beta) \quad \rightarrow \mathbf{W} \text{ を最適化}$$

- GPLVM (Lawrence, NIPS 2003): $\mathbf{W}$ の方に prior を与えて積分消去

$$p(\mathbf{W}) = \prod_{i=1}^{D} \mathrm{N}(\mathbf{w}_i|0,\alpha^{-1}\mathbf{I}) \quad (20)$$

$$p(\mathbf{Y}|\mathbf{X},\beta) = \int p(\mathbf{Y}|\mathbf{X},\beta)p(\mathbf{W})d\mathbf{W} \quad (21)$$

$$= \frac{1}{(2\pi)^{DN/2}|K|^{D/2}} \exp\left(-\frac{1}{2}\mathrm{tr}(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T)\right) \quad (22)$$

# GPLVM (2): PPCA の Dual

$$\log p(\mathbf{Y}|\mathbf{X},\beta) = -\frac{DN}{2}\log(2\pi) - \frac{D}{2}\log|K| - \frac{1}{2}\mathrm{tr}(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T) \tag{23}$$

$$\mathbf{K} = \alpha\mathbf{X}\mathbf{X}^T + \beta^{-1}\mathbf{I} \tag{24}$$

$$\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N]^T \tag{25}$$

- $\mathbf{X}$ に関して微分すると,

$$\frac{\partial L}{\partial \mathbf{X}} = \alpha\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T\mathbf{K}^{-1}\mathbf{X} - \alpha D\mathbf{K}^{-1}\mathbf{X} = 0 \tag{26}$$

$$\Longleftrightarrow \quad \mathbf{X} = \frac{1}{D}\mathbf{Y}\mathbf{Y}^T\mathbf{K}^{-1}\mathbf{X} \quad \Rightarrow \quad \mathbf{X} \simeq U_Q L V^T \tag{27}$$

  - $U_Q$ $(N \times Q)$ : $\mathbf{Y}\mathbf{Y}^T$ の $Q$ 個の最大固有値 $\lambda_1 \cdots \lambda_Q$ に対応する固有ベクトル
  - $L = \mathrm{diag}(l_1, \cdots, l_Q);\ l_i = 1/\sqrt{\frac{\lambda_i}{\alpha D} - \frac{1}{\alpha\beta}}$

# GPLVM (3) : Kernel 化

$$\log p(\mathbf{Y}|\mathbf{X},\beta) = -\frac{DN}{2}\log(2\pi) - \frac{D}{2}\log|K| - \frac{1}{2}\mathrm{tr}(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T)$$

$$\mathbf{K} = \alpha\mathbf{X}\mathbf{X}^T + \beta^{-1}\mathbf{I}, \tag{28}$$

$$\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N]^T \tag{29}$$

- 自然にカーネル化されている $\Longrightarrow$ 任意のカーネル $\mathbf{K}$ を導入

$$k(\mathbf{x}_n, \mathbf{x}_m) = \alpha\exp\left(-\frac{\gamma}{2}(\mathbf{x}_n - \mathbf{x}_m)^2\right) + \delta(n,m)\beta^{-1} \tag{30}$$

- $\dfrac{\partial L}{\partial \mathbf{K}} = \mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T\mathbf{K}^{-1} - D\mathbf{K}^{-1}$
  - $\dfrac{\partial L}{\partial x_{n,j}} = \dfrac{\partial L}{\partial \mathbf{K}}\dfrac{\partial \mathbf{K}}{\partial x_{n,j}}$ を適用して微分
  - Scaled Conjugate Gradient で解ける
    GPLVM in MATLAB: http://www.cs.man.ac.uk/~neill/gplvm/

# GPLVM (4): 例

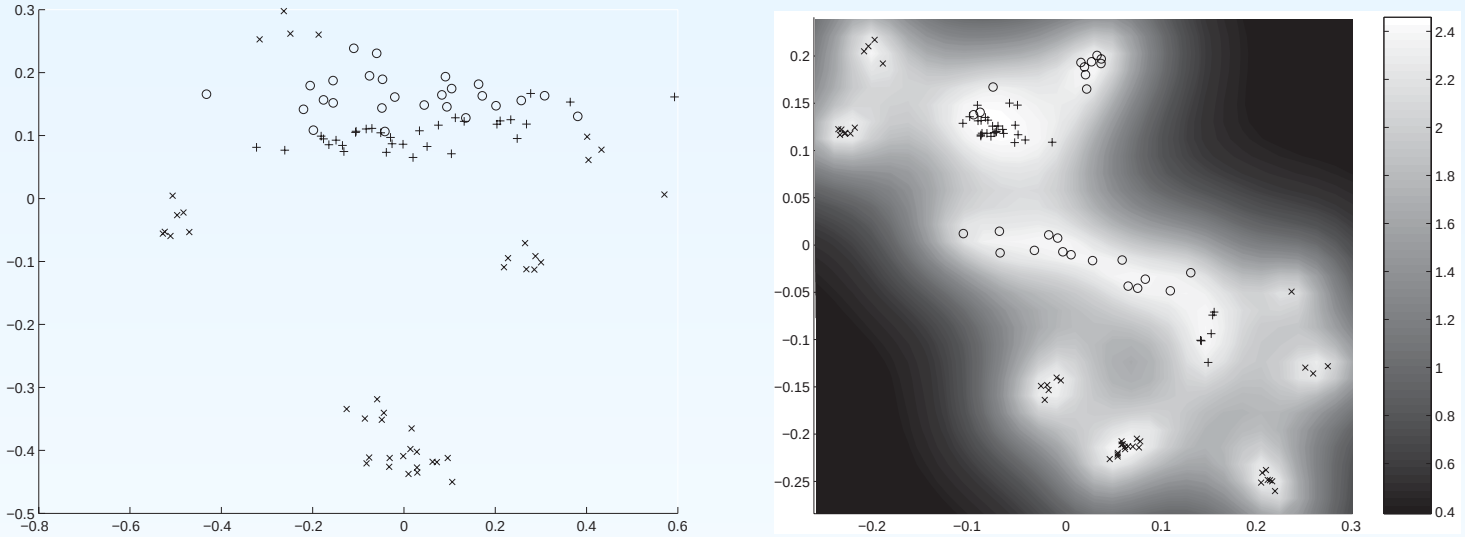
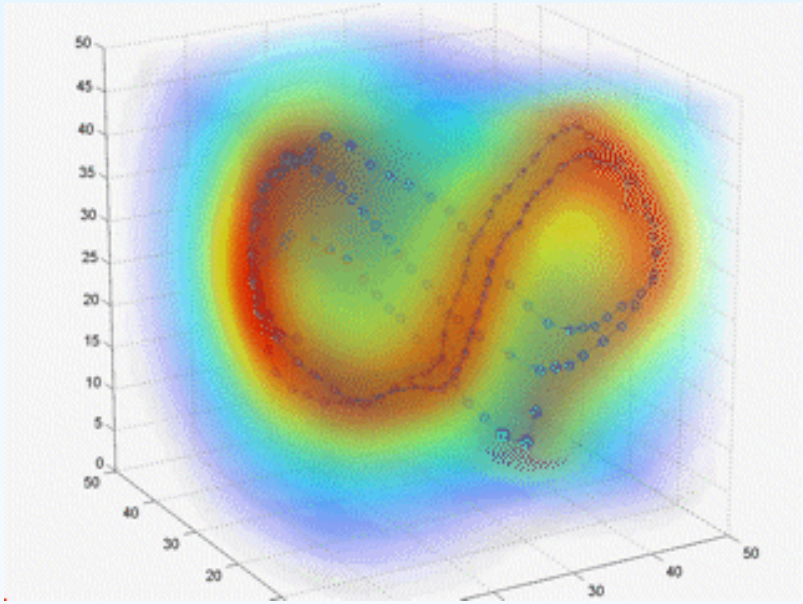



Figure 1: Visualisation of the Oil data with (a) PCA (a linear GPLVM) and (b) A GPLVM which uses an RBF kernel. Crosses, circles and plus signs represent stratified, annular and homogeneous flows respectively. The greyscales in plot (b) indicate the precision with which the manifold was expressed in data-space for that latent point. The optimised parameters of the kernel were $\gamma = 150$, $\alpha = 0.403$ and $\beta = 316$.

- 線形の PPCA(左) より, GP-LVM(右) の方が分離性能が高い
  - ベイズなので, Confidence の分布が同時に得られる
- 計算量が問題 ($O(N^3)$): active set (サポートベクターみたいなもの) を選んで, データをスパース化して最適化

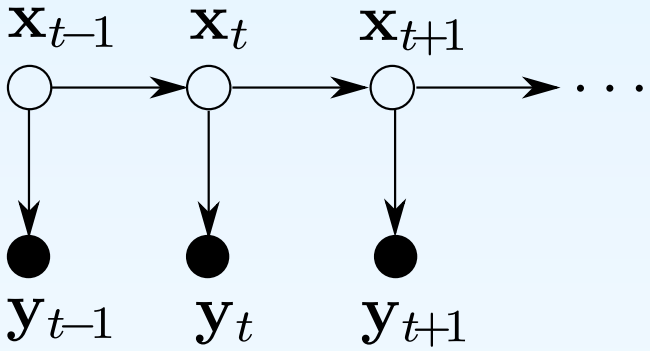
# Gaussian Process Dynamical Model (Hertzmann 2005)




http://www.dgp.toronto.edu/~jmwang/gpdm/

- GPLVM では, 潜在変数 $\mathbf{x}_n$ に分布がなかった
  ⇓
- $\mathbf{x}_n$ が (GP で) 時間発展するモデル.
  - 人間の動き (角度ベクトル) 等の時系列データ.
  - 自然言語の時系列データ?

# GPDM (2): Formulation (1)




$$\begin{cases} \mathbf{x}_t = f(\mathbf{x}_{t-1}; \mathbf{A}) + \epsilon_{x,t} & f \sim \mathrm{GP}(0, \mathbf{K_x}) \quad (31) \\ \mathbf{y}_t = g(\mathbf{x}_t; \mathbf{B}) + \epsilon_{y,t} & g \sim \mathrm{GP}(0, \mathbf{K_y}) \quad (32) \end{cases},$$

- 結合確率 $p(\mathbf{Y}, \mathbf{X}|\alpha, \beta) = p(\mathbf{Y}|\mathbf{X}, \beta)p(\mathbf{X}|\alpha)$ を考える.
- 第 1 項

$$p(\mathbf{Y}|\mathbf{X}, \beta) = \frac{|\mathbf{W}|^N}{(2\pi)^{ND/2}|\mathbf{K}_Y|^{D/2}} \exp\left(-\frac{1}{2}\mathrm{tr}(\mathbf{K}_Y^{-1}\mathbf{Y}\mathbf{W}^2\mathbf{Y}^T)\right) \quad (33)$$

  は GPLVM と基本的に同じ.
  - $\mathbf{K}_Y$ は (この論文では) 普通の RBF カーネル

# GPDM (3): Formulation (2)

- 第 2 項は Markov 時系列

$$p(\mathbf{X}|\alpha) = p(\mathbf{x}_1) \int \prod_{t=2}^{N} p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{A}, \alpha) \underbrace{p(\mathbf{A}|\alpha)}_{\text{Gaussian}} d\mathbf{A} \tag{34}$$

$$= p(\mathbf{x}_1) \frac{1}{(2\pi)^{d(N-1)/2} |\mathbf{K}_X|^d} \exp\left(-\frac{1}{2}\mathrm{tr}(\mathbf{K}_X^{-1}\mathbf{X}_-\mathbf{X}_-^T)\right) \tag{35}$$

  - $\mathbf{X}_- = [\mathbf{x}_2, \cdots, \mathbf{x}_N]^T$ とおいた
  - $\mathbf{K}_X$ は $\mathbf{x}_1 \cdots \mathbf{x}_{N-1}$ の RBF+線形カーネル

$$k(\mathbf{x}, \mathbf{x}') = \alpha_1 \exp\left(-\frac{\alpha_2}{2}||\mathbf{x} - \mathbf{x}'||^2\right) + \alpha_3 \mathbf{x}^T\mathbf{x} + \alpha_4^{-1}\delta(\mathbf{x}, \mathbf{x}') \,. \tag{36}$$

# GPDM (4): Formulation(3)

$$p(\mathbf{Y}, \mathbf{X}, \alpha, \beta) = p(\mathbf{Y}|\mathbf{X}, \beta)p(\mathbf{X}|\alpha)p(\alpha)p(\beta) \tag{37}$$

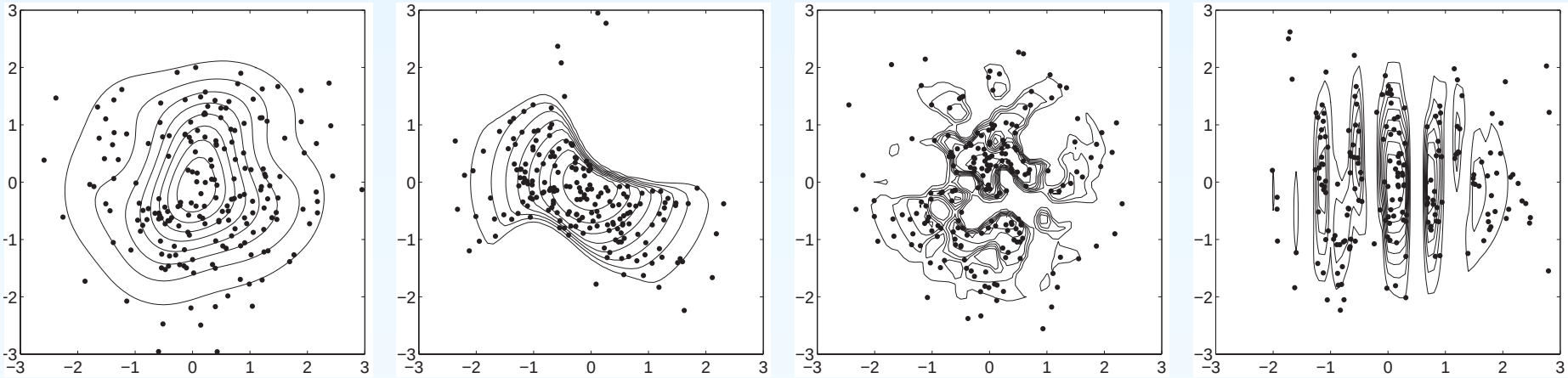
$$p(\alpha) \propto \prod_i \alpha_i^{-1},\ p(\beta) \propto \prod_i \beta_i^{-1}. \tag{38}$$

- 対数尤度は

$$\begin{aligned} -\log p(\mathbf{Y}, \mathbf{X}, \alpha, \beta) =& \frac{1}{2}\mathrm{tr}(\mathbf{K}_X^{-1}\mathbf{X}_-\mathbf{X}_-^T) + \frac{1}{2}\mathrm{tr}(\mathbf{K}_Y^{-1}\mathbf{Y}\mathbf{W}^2\mathbf{Y}^T) \\ & + \frac{d}{2}\log|\mathbf{K}_X| + \frac{D}{2}\log|\mathbf{K}_Y| \quad \text{(正則化項)} \\ & \underbrace{-\log|\mathbf{W}| + \sum_j \log\alpha_j + \sum_j \log\beta_j}_{\text{(定数)}} \end{aligned} \tag{39}$$

$$\longrightarrow \text{最小化.} \tag{40}$$

# Gaussian Process Density Sampler (1)



(a) $\ell_x = 1, \ell_y = 1, \alpha = 1$ (b) $\ell_x = 1, \ell_y = 1, \alpha = 10$ (c) $\ell_x = 0.2, \ell_y = 0.2, \alpha = 5$ (d) $\ell_x = 0.1, \ell_y = 2, \alpha = 5$

- GP は任意の関数の prior →確率密度関数のモデルに使えないか?

$$p(\mathbf{x}) = \frac{1}{Z(f)} \Phi(f(\mathbf{x})) \pi(\mathbf{x}) \tag{41}$$

  - $f(\mathbf{x}) \sim \mathrm{GP}(\mathbf{x})$ ； $\pi(\mathbf{x})$：事前分布
  - $\Phi(\mathbf{x}) \in [0, 1]$：シグモイド関数
    - ex. $\Phi(x) = 1/(1 + \exp(-x))$

# Gaussian Process Density Sampler (2)

$$p(\mathbf{x}) = \frac{1}{Z(f)} \Phi(f(\mathbf{x})) \pi(\mathbf{x}) \tag{42}$$

- 生成プロセス: Rejection sampling
  1. Draw $\mathbf{x} \sim \pi(\mathbf{x})$.
  2. Draw $r \sim \mathrm{Uniform}[0, 1]$.
  3. If $r < \Phi(g(\mathbf{x}))$ then accept $\mathbf{x}$; else reject $\mathbf{x}$
- Accept された $N$ 個の観測データの背後に, reject された $M$ 個のデータとその場所が存在 (隠れ変数)
  - 分配関数 $Z(f)$ は求まらないが, $\Phi(g(\mathbf{x}))$ は求まる
    $\rightarrow$ MCMC!
  - Infinite Mixture とは別の確率密度のモデル化

# Gaussian Processes for NLP?

- 言語では出力 $\mathbf{y} = (y_1, \cdots, y_N)$ は Gaussian ではないので, 直接は使えない
- 隠れ変数の (連続的な) 分布に使える可能性は高い
  - Ex: Latent CCA in (Haghighi+, ACL 2008)
- 分類もできるが, 特に GP でないといけない理由は希少
- 言語+$\alpha$ のモデル化に特に重要だと思われる

# まとめ

- Gaussian Process…連続的な関数のベイズモデル
  - カーネル関数で定義される, 無限次元のガウス分布
  - 基底関数の空間での線形モデルで, 重みを積分消去したもの
- カーネル設計が重要
  - カーネルの違いにより, さまざまな振舞い
  - カーネルのパラメータは, 確率モデルなので最適化できる
- 回帰問題だけでなく, 隠れ変数や確率分布のモデルにも使える, 確率モデルの Building Block
- 計算量が課題→研究は最近様々に進んでいる

# Literature

- “Gaussian Process Dynamical Models”. J. Wang, D. Fleet, and A. Hertzmann. NIPS 2005. http://www.dgp.toronto.edu/ jmwang/gpdm/
- “Gaussian Process Latent Variable Models for Visualization of High Dimensional Data”. Neil D. Lawrence, NIPS 2003.
- “The Gaussian Process Density Sampler”. Ryan Prescott Adams, Iain Murray and David MacKay. NIPS 2008.
- “Archipelago: Nonparametric Bayesian Semi-Supervised Learning”. Ryan Prescott Adams and Zoubin Ghahramani. ICML 2009.

## Readings

- “Gaussian Processes for Machine Learning”. Rasmussen and Williams, MIT Press, 2006. http://www.gaussianprocess.org/gpml/
- “Gaussian Processes — A Replacement for Supervised Neural Networks?”. David MacKay, Lecture notes at NIPS 1997. http://www.inference.phy.cam.ac.uk/mackay/GP/
  - Videolectures.net: “*Gaussian Process Basics.*” http://videolectures.net/gpip06_mackay_gpb/
- Pattern Recognition and Machine Learning, Chapter 6. Christopher Bishop, Springer, 2006. http://ibisforest.org/index.php?PRML
- ガウス過程に関するメモ (1). 正田備也, 2007. http://www.iris.dti.ne.jp/~tmasada/2007071101.pdf

## Codes

- GPML MATLAB codes. http://www.gaussianprocess.org/gpml/code/
- GPLVM MATLAB/C++ codes. http://www.cs.manchester.ac.uk/~neill/gplvm/
- GPDM code. http://www.dgp.toronto.edu/~jmwang/gpdm/