Bayesian Deep Neural Networks

Gaussian Process

Sungjoon Choi

February 9, 2018

Seoul National University

Table of contents

Introduction

Introduction

A **Gaussian process** is a collection of random variables, any finite number of which have a joint Gaussian distribution [1].

Most of the contents are from Prof. Songhwai Oh's slides.

Introduction

- **univariate Gaussian distribution**

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}(x-\mu)^2\right)$$

- **central limit theorem**: Let $X_1, X_2, \cdots$ be independent and identically distributed with $\mathbb{E}(X_i) = \mu$ and $\mathbf{var}(X_i) = \sigma^2$. If $S_n = X_1 + X_2 + \cdots + X_n$, then

$$\frac{S_n - n\mu}{\sigma\sqrt{n}} \overset{d}{\sim} \mathcal{N}(0,1)$$

Introduction

- **multivariate Gaussian distribution**

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right)$$

$\mathbf{x} \sim \mathcal{N}(\mu, \Sigma)$, where μ is the mean vector and Σ is the covariance matrix.

$$\mu = \mathbb{E}(\mathbf{x}) = \begin{bmatrix} \mathbb{E}(X_1) \\ \vdots \\ \mathbb{E}(X_n) \end{bmatrix} \qquad \Sigma = \mathbf{cov}(\mathbf{x})$$

Introduction

- **conditional Gaussian distribution**
  If $\mathbf{x} \in \mathbb{R}^r$ and $\mathbf{y} \in \mathbb{R}^m$ are jointly Gaussian with $n = r + m$, mean vector

$$\mu = \begin{bmatrix} \mathbb{E}(\mathbf{x}) \\ \mathbb{E}(\mathbf{y}) \end{bmatrix}, \text{ and covariance matrix } \Sigma = \begin{bmatrix} \Sigma_{\mathbf{xx}} & \Sigma_{\mathbf{xy}} \\ \Sigma_{\mathbf{yx}} & \Sigma_{\mathbf{yy}} \end{bmatrix}.$$

  Then the conditional pdf $p(\mathbf{x}|\mathbf{y})$ is also a Gaussian random vector with mean $\mathbb{E}(\mathbf{x}|\mathbf{y})$ and covariance matrix $\Sigma_{\mathbf{x}|\mathbf{y}}$ where

$$\mathbb{E}(\mathbf{x}|\mathbf{y}) = \mathbb{E}(\mathbf{x}) + \Sigma_{\mathbf{xy}}\Sigma_{\mathbf{yy}}^{-1}(\mathbf{y} - \mathbb{E}(\mathbf{y}))$$
$$\Sigma_{\mathbf{x}|\mathbf{y}} = \Sigma_{\mathbf{xx}} - \Sigma_{\mathbf{xy}}\Sigma_{\mathbf{yy}}^{-1}\Sigma_{\mathbf{yx}}.$$

Gaussian process

Gaussian process

- **Gaussian process**: A random process $X(t)$ is a **Gaussian process** if for all $k \in \mathbb{N}$ for all $t_1, \ldots, t_k$, a random vector formed by $X(1), \ldots, X(t_k)$ is jointly Gaussian.
- The joint density is completely specified by
  - Mean: $m(t) = \mathbb{E}(X(t))$, where $m(\cdot)$ is known as a mean function.
  - Covariance: $k(t, s) = \mathbf{cov}(X(t), X(s)) =$, where $k(\cdot, \cdot)$ is known as a covariance function.
- Notation: $X(t) \sim \mathcal{GP}(m(t), k(t, s))$

Gaussian process

- Example: $X(t) = tA$, where $A \sim \mathcal{N}(0, 1)$ and $t \in \mathbb{R}$.
  - Mean: $m(t) = \mathbb{E}(X(t)) = t\mathbb{E}(A) = 0$
  - Covariance: $k(t, s) = \mathbb{E}(tAsA) = ts$

Gaussian process regression

- **Gaussian process** and **Gaussian process regression** are different.
- Notations
  - $\mathcal{X}$: index set (e.g., time $\mathbb{R}$ or space $\mathbb{R}^3$)
  - $z(x)$: a collection of random variables with $x \in \mathcal{X}$.
- $z(x)$ is a **Gaussian process** if for any finite set $\{x_1, \ldots, x_n\}$, $\{z(x_1), \ldots, z(x_n)\}$ has a multivariate Gaussian distribution, with mean $\mu \in \mathbb{R}^n$ and covariance $K \in \mathbb{R}^{n \times n}$.
- The mean μ and covariance K depends on the chosen finite set $\{x_1, \ldots, x_n\}$.

Gaussian process regression

- **Gaussian process regression**: A nonparametric Bayesian regression method using the properties of Gaussian processes.
- Two views to interpret **Gaussian process regression**
  - Weight-space view
  - Function-space view

Weight space view

Linear regression

- $f(\mathbf{x}) = \mathbf{x}^T\mathbf{w}$
- $y(\mathbf{x}) = f(\mathbf{x}) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$.
- Suppose we have collected n input-output pairs $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$.
- Define $X = [\mathbf{x}_1^T; \cdots; \mathbf{x}_n^T]^T$. Then

$$\begin{aligned} p(\mathbf{y}|X, \mathbf{w}) &= \prod_{i=1}^{n} p(y_i|\mathbf{x}_i, \mathbf{w}) \\ &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp(-\frac{(y_i - \mathbf{x}_i^T w)^2}{2\sigma_n^2}) \\ &= \frac{1}{(2\pi\sigma_n^2)^{n/2}} \exp(-\frac{1}{2\sigma_n^2}\|\mathbf{y} - X^T\mathbf{w}\|^2) \\ &= \mathcal{N}(\mathbf{y}; X^T\mathbf{w}, \sigma_n^2\mathbf{I}). \end{aligned}$$

- Goal of **linear regression** is to find $\mathbf{w}$ such that
  - $\|\mathbf{y} - X^T\mathbf{w}\|^2$ is minimized.
  - Solution: $\hat{\mathbf{w}} = (XX^T)^{-1}X\mathbf{y}$

Bayesian formulation

- **Bayesian formulation:** Put a prior over the parameters, i.e., $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \Sigma_p)$.
- Finding the posterior distribution is the goal of a Bayesian method:

$$p(\mathbf{w}|\mathbf{y}, X) = \frac{p(\mathbf{y}|X, \mathbf{w})p(\mathbf{w})}{p(\mathbf{y}|X)}$$

Bayesian formulation

- **Bayesian formulation:**

$$p(\mathbf{w}|\mathbf{y}, X) = \frac{p(\mathbf{y}|X, \mathbf{w})p(\mathbf{w})}{p(\mathbf{y}|X)}$$
$$\propto \exp\left(\frac{1}{2\sigma_n^2}(\mathbf{y} - X^T\mathbf{w})^T(\mathbf{y} - X^T\mathbf{w})\right)\exp\left(-\frac{1}{2}\mathbf{w}^T\Sigma_p^{-1}\mathbf{w}\right)$$
$$\propto \exp\left(\frac{1}{2}(\mathbf{w} - \bar{\mathbf{w}})^T A(\mathbf{w} - \bar{\mathbf{w}})\right)$$

  where $\bar{\mathbf{w}} = \frac{1}{\sigma_n^2}A^{-1}X\mathbf{y}$ and $A = \left(\frac{1}{\sigma_n^2}XX^T + \Sigma_p^{-1}\right)$.
- Hence,

$$P(\mathbf{w}|\mathbf{y}, X) = \mathcal{N}(\bar{\mathbf{w}}, A^{-1}).$$

- Computing the analytic form a posterior distribution is not alway possible.
- In fact, there are not many cases and the priors that enable the analytic posterior forms are known as conjugate priors.

Bayesian formulation

- The parameter that maximizes the posterior distribution is called the maximum a posteriori (MAP) solution:

$$\hat{\mathbf{w}}_{MAP} = \frac{1}{\sigma_n^2}\left(\frac{1}{\sigma_n^2}XX^T + \Sigma_p^{-1}\right)^{-1}X\mathbf{y}$$

- We also had the solution that maximizes the likelihood distribution (maximum likelihood estimation (MLE) solution):

$$\hat{\mathbf{w}}_{MLE} = (XX^T)^{-1}X\mathbf{y}$$

- Then, what is a Bayesian solution?

Bayesian formulation

Suppose that we want to predict at a new input $\mathbf{x}_*$, then the predictive distribution of $f_* = f_*(\mathbf{x}_*)is$

$$\begin{aligned} p(f_*|\mathbf{x}_*, X, \mathbf{y}) &= \int p(f_*|\mathbf{x}_*, \mathbf{w}) p(\mathbf{w}|X, \mathbf{y}) d\mathbf{w} \\ &= \mathcal{N}\left(\frac{1}{\sigma_n^2}\mathbf{x}_*^T A^{-1} X\mathbf{y},\ \mathbf{x}_*^T A^{-1}\mathbf{x}_*\right) \end{aligned}$$

where $A = \left(\frac{1}{\sigma_n^2}XX^T + \Sigma_p^{-1}\right)$.

Kernel trick

- Let $\phi : \mathbb{R}^D \to \mathbb{R}^N$ be a mapping from the input space to the high dimensional feature space ($N \gg D$).
- $f(\mathbf{x}) = \phi(\mathbf{x})^T \mathbf{w}$
- Define $\Phi(X) = \begin{bmatrix} | & | & | \\ \phi(x_1) & \cdots & \phi(x_n) \\ | & | & | \end{bmatrix} \in \mathbb{R}^{N \times n}$.
- Recall our previous posterior:
  $f_* | \mathbf{x}_*, X, y \sim \mathcal{N}(\frac{1}{\sigma_n^2} x_*^T A^{-1} X y, x_*^T A^{-1} x_*)$
- Then

$$f_* | \mathbf{x}_*, X, \mathbf{y} \sim \mathcal{N}\left(\frac{1}{\sigma_n^2} \phi(\mathbf{x}_*)^T A^{-1} \Phi \mathbf{y}, \phi(\mathbf{x}_*)^T A^{-1} \phi(\mathbf{x}_*)\right),$$

  where $A = \frac{1}{\sigma_n^2} \Phi \Phi^T + \Sigma_p^{-1}$ and $A \in \mathbb{R}^{N \times N}$.
- If $N \gg 1$, then inverting $A \in \mathbb{R}^{N \times N}$ could be computationally intractable.

Kernel trick

- Let $K = \Phi^T \Sigma_p \Phi$ and $\phi_* = \phi(\mathbf{x}_*)$ and consider (recall $A = \sigma^{-2}\Phi\Phi^T + \Sigma_p^{-1}$)

$$\begin{aligned} A\Sigma_p\Phi &= (\sigma_n^{-2}\Phi\Phi^T + \Sigma_p^{-1})\Sigma_p\Phi = \sigma_n^{-2}\Phi\Phi^T\Sigma_p\Phi + \Phi \\ &= \sigma_n^{-2}\Phi(\Phi^T\Sigma_p\Phi + \sigma_n^2 I) \\ &= \sigma_n^{-2}\Phi(K + \sigma_n^2 I). \end{aligned}$$

- Premultiply A^{-1} and post-multiply $(K + \sigma_n^2 I)^{-1}$ to get

$$\begin{aligned} \sigma_n^{-2}A^{-1}\Phi(K + \sigma_n^2 I)(K + \sigma_n^2 I)^{-1} &= A^{-1}A\Sigma_p\Phi(K + \sigma_n^2 I)^{-1} \\ \sigma_n^{-2}A^{-1}\Phi &= \Sigma_p\Phi(K + \sigma_n^2 I)^{-1} \end{aligned}$$

Kernel trick

- Predictive distribution of f_*:

$$f_*|\mathbf{x}_*, X, \mathbf{y} \sim \mathcal{N}\left(\sigma_n^{-2}\phi(\mathbf{x}_*)^T A^{-1}\Phi\mathbf{y}, \phi(\mathbf{x}_*)^T A^{-1}\phi(\mathbf{x}_*)\right),$$

- Hence the predictive mean becomes:

$$\sigma_n^{-2}\phi(\mathbf{x}_*)^T A^{-1}\Phi\mathbf{y} = \phi_*^T\Sigma_p\Phi(K + \sigma^2 I)^{-1}\mathbf{y}.$$

- Using the matrix inversion lemma, the predictive variance becomes:

$$\phi(\mathbf{x}_*)^T A^{-1}\phi(\mathbf{x}_*) = \phi_*^T\Sigma_p\phi_* - \phi_*^T\Sigma_p\Phi(K + \sigma_n^2 I)^{-1}\Phi^T\Sigma_p\phi_*.$$

Kernel trick

- (Original) predictive distribution of f_*:

$$f_*|\mathbf{x}_*, X, \mathbf{y} \sim \mathcal{N}\left(\sigma_n^{-2}\phi(\mathbf{x}_*)^T A^{-1}\Phi\mathbf{y}, \phi(\mathbf{x}_*)^T A^{-1}\phi(\mathbf{x}_*)\right),$$

- Substitute new terms to get the predictive distribution of f_*:

$$f_*|\mathbf{x}_*, X, \mathbf{y} \sim \mathcal{N}\left(\mu_*, \Sigma_*\right)$$

  where $\mu_* = \phi_*^T\Sigma_p\Phi(K+\sigma^2 I)^{-1}\mathbf{y}$ and
  $\Sigma_* = \phi_*^T\Sigma_p\phi_* - \phi_*^T\Sigma_p\Phi(K+\sigma_n^2 I)^{-1}\Phi^T\Sigma_p\phi_*$

- Note that $A \in \mathbb{R}^{N\times N}$ and $K \in \mathbb{R}^{D\times D}$ where N is a feature dimension and D is an input dimension where $N \gg D$.
- We can now apply the **kernel trick** and replace $\phi(\mathbf{x})^T\Sigma_p\phi(\mathbf{x}')$ by $k(\mathbf{x}, \mathbf{x}')$.

Kernel trick

- The predictive distribution of f_* using the kernel trick:

$$f_* | \mathbf{x}_*, X, \mathbf{y} \sim \mathcal{N}(\mu_*, \Sigma_*)$$

  where $\mu_* = k(\mathbf{x}_*, X)(k(X, X) + \sigma_n^2 I)^{-1}\mathbf{y}$ and
  $\Sigma_* = k(\mathbf{x}_*, \mathbf{x}_*) - k(\mathbf{x}_*, X)(k(X, X) + \sigma_n^2 I)^{-1}k(X, \mathbf{x}_*)$

- Now, we have Gaussian process regression.
- But, why?

Function space view

Function space view

- Recall a Gaussian process:

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$$

where $m(\mathbf{x}) = \mathbb{E}(f(\mathbf{x}))$ is a mean function and
$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$ is a covariance function.

Function space view

- Recall our previous example:

$$g(\mathbf{x}) = \phi(\mathbf{x})^T \mathbf{w}$$

  where $\mathbf{w} \sim \mathcal{N}(0, \Sigma_p)$.
- Is $g(\mathbf{x})$ a Gaussian process?
  - Yes!
  - $\mathbb{E}(g(\mathbf{x})) = \phi(\mathbf{x})^T \mathbb{E}(\mathbf{w}) = 0$
  - $\mathbb{E}(g(\mathbf{x})g(\mathbf{x}')) = \phi(\mathbf{x})^T \mathbb{E}(\mathbf{w}\mathbf{w}^T)\phi(\mathbf{x}') = \phi(\mathbf{x})^T \Sigma_p \phi(\mathbf{x})' = k(\mathbf{x}, \mathbf{x}')$
  - Hence, for $[g(\mathbf{x}_1), \ldots, g(\mathbf{x}_n)]$ are jointly Gaussian.
  - Therefore, $g(\mathbf{x})$ is a Gaussian process.

Function space view

Let $f(\mathbf{x})$ be a (zero-mean) Gaussian process. Then

$f(\mathbf{X}) = \begin{bmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_n) \end{bmatrix} \in \mathbb{R}^n$ and $f(\mathbf{x}_*) \in \mathbb{R}$ are jointly Gaussian, i.e.,

$$\begin{bmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_n) \\ f(\mathbf{x}_*) \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \cdots & k(\mathbf{x}_1, \mathbf{x}_n) & k(\mathbf{x}_1, \mathbf{x}_*) \\ \vdots & \ddots & \vdots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & \cdots & k(\mathbf{x}_n, \mathbf{x}_n) & k(\mathbf{x}_n, \mathbf{x}_*) \\ k(\mathbf{x}_*, \mathbf{x}_1) & \cdots & k(\mathbf{x}_*, \mathbf{x}_n) & k(\mathbf{x}_*, \mathbf{x}_*) \end{bmatrix}\right).$$

Function space view

- We rewrite the joint Gaussian distribution as

$$\begin{bmatrix} \mathbf{f} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) & K(X,\mathbf{x}_*) \\ K(\mathbf{x}_*,X) & k(\mathbf{x}_*,\mathbf{x}_*) \end{bmatrix}\right).$$

- Recall that the conditional distribution $p(\mathbf{x}|\mathbf{y})$ of a jointly Gaussian random vector $\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}$ is also a Gaussian random vector with mean $\mathbb{E}(\mathbf{x}|\mathbf{y})$ and covariance matrix $\Sigma_{\mathbf{x}|\mathbf{y}}$ where

$$\mathbb{E}(\mathbf{x}|\mathbf{y}) = \mathbb{E}(\mathbf{x}) + \Sigma_{\mathbf{xy}}\Sigma_{\mathbf{yy}}^{-1}(\mathbf{y} - \mathbb{E}(\mathbf{y}))$$
$$\Sigma_{\mathbf{x}|\mathbf{y}} = \Sigma_{\mathbf{xx}} - \Sigma_{\mathbf{xy}}\Sigma_{\mathbf{yy}}^{-1}\Sigma_{\mathbf{yx}}.$$

Function space view

By conditioning, we get

$$f_*|\mathbf{x}_*, X, \mathbf{f} \sim \mathcal{N}(\mu_*, \sigma_*^2)$$

where

$$\mu_* = K(\mathbf{x}_*, X)K(X, X)^{-1}\mathbf{f}$$

and

$$\sigma_* = k(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, X)K(X, X)^{-1}K(X, \mathbf{x}_*).$$

Function space view

- In the previous case, measurement noise is not included.
- Let $y(\mathbf{x}) = f(\mathbf{x}) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$. Then the covariance between two outputs becomes:

$$\mathbf{cov}(y(\mathbf{x}_1, \mathbf{x}_2)) = k(\mathbf{x}_1, \mathbf{x}_2) + \sigma_n^2.$$

- Consequently, the joint distribution of y and $\mathbf{f}_*$ becomes:

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X, \mathbf{x}_*) \\ K(\mathbf{x}_*, X) & k(\mathbf{x}_*, \mathbf{x}_*) \end{bmatrix}\right).$$

Function space view

By conditioning, we get

$$f_*|\mathbf{x}_*, X, \mathbf{y} \sim \mathcal{N}(\mu_*, \sigma_*^2)$$

where

$$\mu_* = K(\mathbf{x}_*, X)(K(X, X) + \sigma_n^2 I)^{-1}\mathbf{y}$$

and

$$\sigma_* = k(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, X)(K(X, X) + \sigma_n^2 I)^{-1}K(X, \mathbf{x}_*).$$

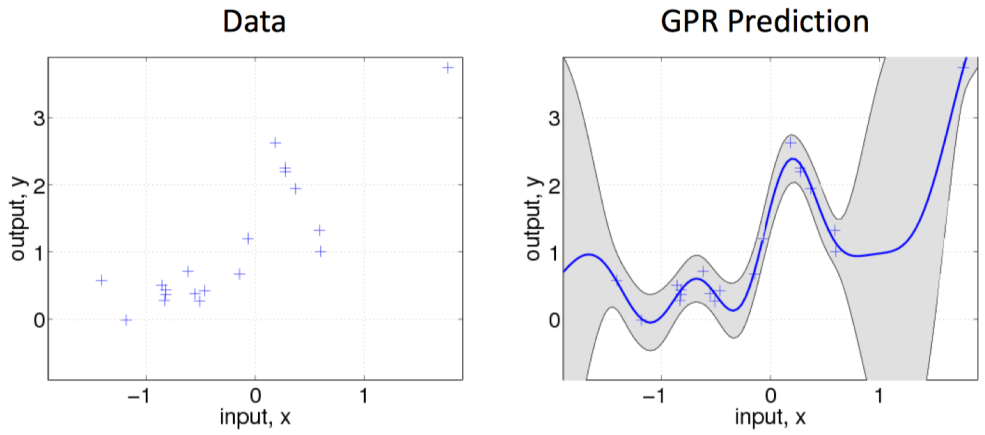
Function space view

- The predictive mean of a Gaussian process is a linear predictor (linear combination of $\mathbf{y}$):

$$f_* = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x}_*)$$

  where $\alpha = (K + \sigma_n^2 I)\mathbf{y}$.
- Interestingly, the form above is identical to the solution of kernel ridge regression.
- f_* can also be interpreted within the theory of Harmonic analysis.

Comments on Gaussian process regression

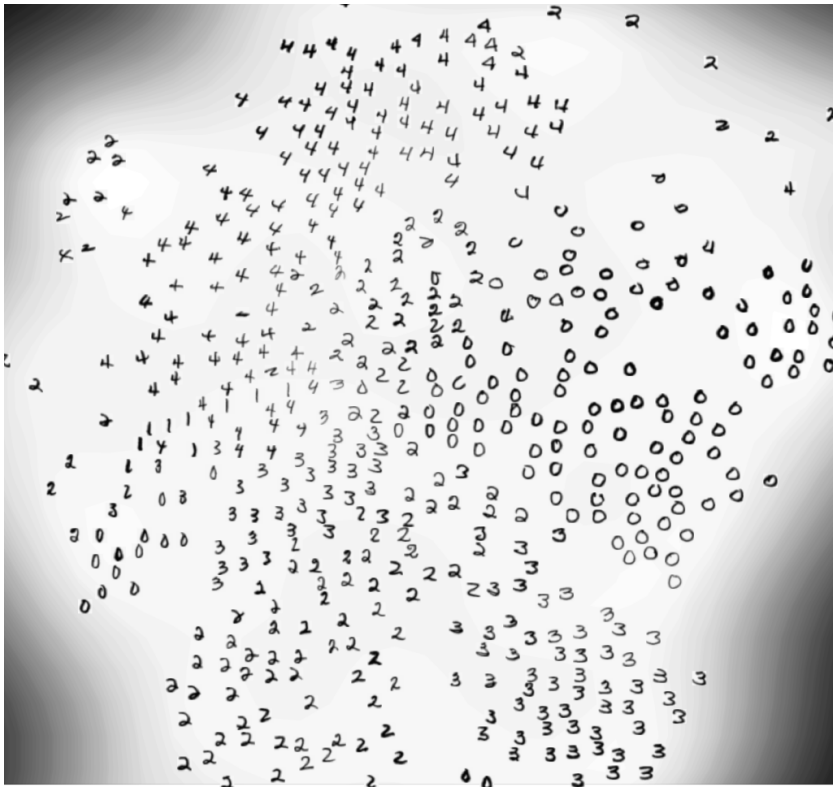


- **Pros**: principled, probabilistic, predictive uncertainty
- **Cons**: computationally intensive ($O(n^3)$ where n is the number of data)

Gaussian process latent variable model (GPLVM)

Gaussian process latent variable model

Gaussian process latent variable models for visualization of high dimensional data



Gaussian process latent variable model

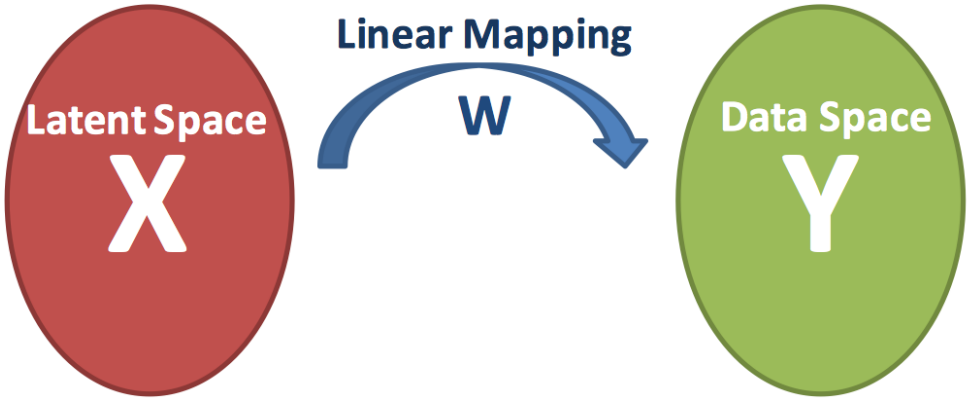
- GPLVM is non-linear probabilistic PCA (PPCA).
  - Dimension reduction
  - Non-linear mapping
- We will see the relationship between PPCA and GPLVM.

Gaussian process latent variable model

- Notations
  - observed data: $Y = \begin{bmatrix} -\mathbf{y}_1 - \\ \vdots \\ -\mathbf{y}_n - \end{bmatrix} \in \mathbb{R}^{n \times d}$
  - latent data: $X = \begin{bmatrix} -\mathbf{x}_1 - \\ \vdots \\ -\mathbf{x}_n - \end{bmatrix} \in \mathbb{R}^{n \times q}$
  - linear mapping matrix $W \in \mathbb{R}^{d \times q} \Rightarrow Y = XW^T$
  - $a_{(i)}$: i-th row vector of A
  - a_i: i-th column vector of A
  - YY^T: matrix inner product $\Leftarrow$ kernel trick can be used.
  - $Y^T Y$: covariance matrix

Gaussian process latent variable model

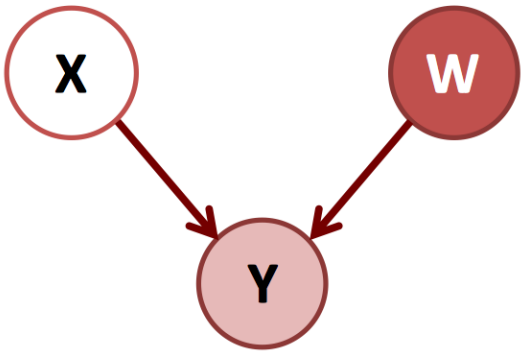
- Linear mapping



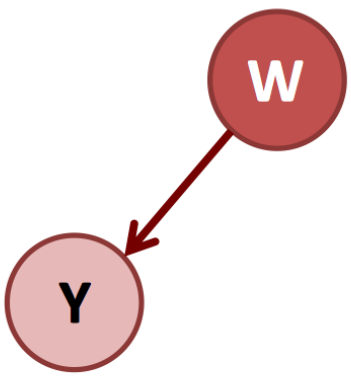
- We want to represent our observed data Y with lower dimensional data X.
- Assum a linear mapping using W.

$$Y = XW^T$$

Gaussian process latent variable model



Arrow represents **conditional distribution.**



Marginalize X by **Integrating** Out.

- Probabilistic PCA
  - Prior distribution:

$$p(X) = \prod_{i=1}^{n} \mathcal{N}(\mathbf{x}_{(i)}|0, I)$$

  - Likelihood:

$$p(Y|X, W) = \prod_{i=1}^{n} \mathcal{N}(y_{(i)}|W\mathbf{x}_{(i)}, \beta^{-1}I)$$

  - Marginal likelihood:

$$\begin{aligned} p(Y|W) &= \int_X p(Y|X, W)p(X)dX \\ &= \prod_{i=1}^{n} \mathcal{N}(y_{(i)}|0, WW^T + \beta^{-1}I) \end{aligned}$$

Gaussian process latent variable model

- Probabilistic PCA
  - Marginal likelihood

$$p(Y|W) = \prod_{i=1}^{n} \mathcal{N}(y_{(i)}|0, WW^T + \beta^{-1}I)$$

  - Log marginal likelihood

$$\ln p(Y|W) = \frac{n}{2} \ln |C| - \frac{1}{2}\text{trace}(C^{-1}Y^T Y)$$

    where $C = WW^T + \beta^{-1}I$
  - Derivatives

$$\frac{\partial \ln p(Y|W)}{\partial W} = n(-C^{-1}W + C^{-1}Y^T YC^{-1}W)$$

  - Solution

$$\frac{\partial \ln p(Y|W)}{\partial W} = 0 \Rightarrow \hat{W} = U_q L V^T$$

    where U_q and Λ_q are first q eigenvectors and eigenvalues of $Y^T Y$, $L = (\Lambda_q - \beta^{-1}I)^{1/2}$, and V is an arbitrary rotation matrix.

Gaussian process latent variable model

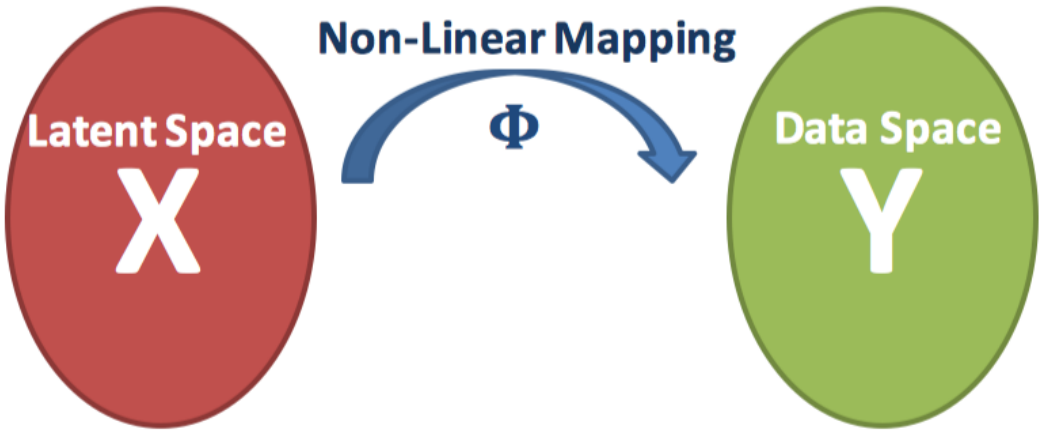
- Probabilistic PCA
  - Solution

$$\frac{\partial \ln p(Y|W)}{\partial W} = 0 \Rightarrow \hat{W} = U_q L V^T$$

    where U_q and Λ_q are first q eigenvectors and eigenvalues of $Y^T Y$, $L = (\Lambda_q - \beta^{-1} I)^{1/2}$, and V is an arbitrary rotation matrix.
  - Note that U_q is the solution of standard PCA.
  - $L = (\Lambda_q - \beta^{-1} I)^{1/2}$ is a scaling diagonal matrix.

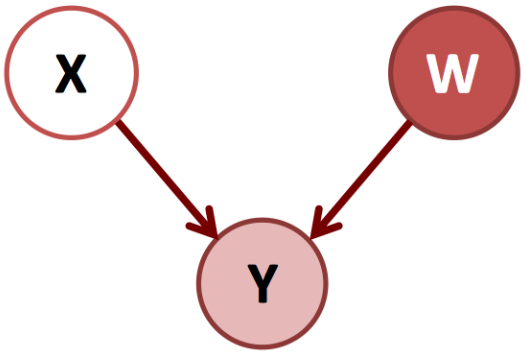
Gaussian process latent variable model

- Nonlinear mapping

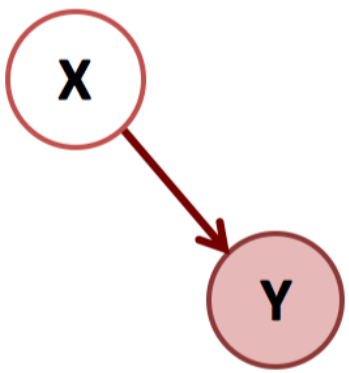


- GPLVM starts with dual PPCA.

Gaussian process latent variable model



Arrow represents **conditional distribution.**



Marginalize W by **Integrating** Out.

- Dual probabilistic PCA
  - Prior distribution:

$$p(W) = \prod_{j=1}^{d} \mathcal{N}(\mathbf{w}_{(j)}|0, I)$$

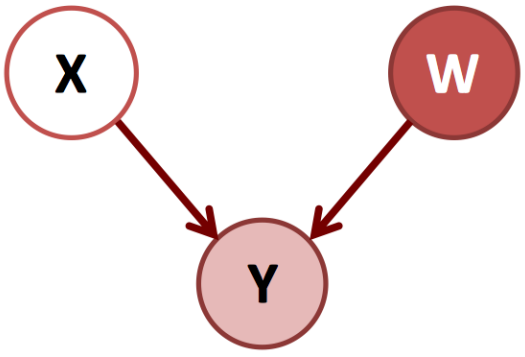
  - Likelihood:

$$p(Y|X, W) = \prod_{i=1}^{n} \mathcal{N}(y_{(i)}|W\mathbf{x}_{(i)}, \beta^{-1}I)$$

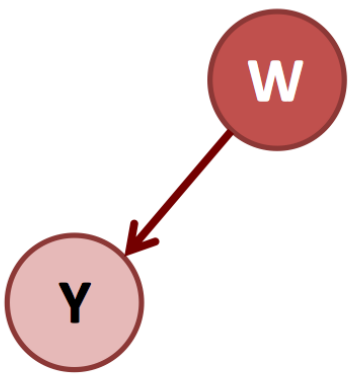
  - Marginal likelihood:

$$\begin{aligned} p(Y|X) &= \int_W p(Y|X, W)p(W)dW \\ &= \prod_{j=1}^{d} \mathcal{N}(y_j|0, XX^T + \beta^{-1}I) \end{aligned}$$

Gaussian process latent variable model



Arrow represents **conditional distribution.**



Marginalize X by **Integrating** Out.

- Probabilistic PCA (recall)
  - Prior distribution:

$$p(X) = \prod_{i=1}^{n} \mathcal{N}(\mathbf{x}_{(i)}|0, I)$$

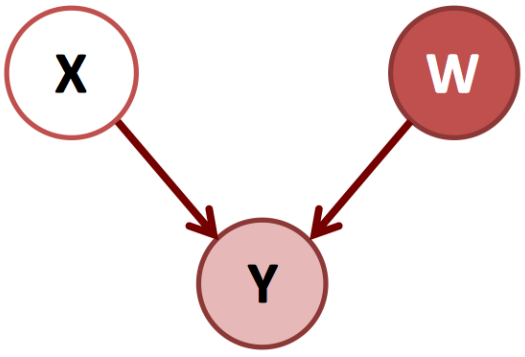
  - Likelihood:

$$p(Y|X, W) = \prod_{i=1}^{n} \mathcal{N}(y_{(i)}|W\mathbf{x}_{(i)}, \beta^{-1}I)$$

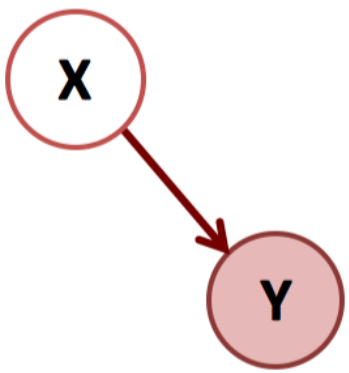
  - Marginal likelihood:

$$\begin{aligned} p(Y|W) &= \int_X p(Y|X, W)p(X)dX \\ &= \prod_{i=1}^{n} \mathcal{N}(y_{(i)}|0, WW^T + \beta^{-1}I) \end{aligned}$$

Gaussian process latent variable model



Arrow represents **conditional distribution.**



Marginalize W by **Integrating** Out.

- Dual probabilistic PCA
  - Prior distribution:

$$p(W) = \prod_{j=1}^{d} \mathcal{N}(\mathbf{w}_{(j)}|0, I)$$

  - Likelihood:

$$p(Y|X, W) = \prod_{i=1}^{n} \mathcal{N}(y_{(i)}|W\mathbf{x}_{(i)}, \beta^{-1}I)$$

  - Marginal likelihood:

$$\begin{aligned} p(Y|X) &= \int_W p(Y|X, W)p(W)dW \\ &= \prod_{j=1}^{d} \mathcal{N}(y_j|0, XX^T + \beta^{-1}I) \end{aligned}$$

Gaussian process latent variable model

- Dual probabilistic PCA
  - Marginal likelihood

$$p(Y|X) = \prod_{j=1}^{d} \mathcal{N}(y_{(j)}|0, XX^T + \beta^{-1}I)$$

  - Log marginal likelihood

$$\ln p(Y|X) = \frac{d}{2} \ln |C| - \frac{1}{2} \text{track}(C^{-1}YY^T)$$

    where $C = XX^T + \beta^{-1}I$
  - Derivatives

$$\frac{\partial \ln p(Y|X)}{\partial X} = n(-dC^{-1}X + C^{-1}YY^TC^{-1}X)$$

  - Solution

$$\frac{\partial \ln p(Y|X)}{\partial X} = 0 \Rightarrow \hat{X} = U_q L V^T$$

    where U_q and Λ_q are first q eigenvectors and eigenvalues of YY^T, $L = (\Lambda_q - \beta^{-1}I)^{1/2}$, and V is an arbitrary rotation matrix.

Gaussian process latent variable model

- Dual Probabilistic PCA
  - Solution

$$\frac{\partial \ln p(Y|X)}{\partial X} = 0 \Rightarrow \hat{X} = U_q L V^T$$

    where U_q and Λ_q are first q eigenvectors and eigenvalues of YY^T, $L = (\Lambda_q - \beta^{-1} I)^{1/2}$, and V is an arbitrary rotation matrix.
  - Note that since YY^T is a matrix inner product, one can derive kernel PCA from here.

Gaussian process latent variable model

Table 1: Summaries of PPCA and DPPCA

	Prior	Optimizing	Product over	Variance
PPCA	X	W	data n	$WW^T + \beta^{-1}I$
DPPCA	W	X	dimension d	$XX^T + \beta^{-1}I$

Gaussian process latent variable model

- Gaussian process (GP) prior:

$$p(Y|X) = \mathcal{N}(Y|0, K)$$

  where $[K]_{(i,j)} = k(\mathbf{x}_i, \mathbf{x}_j)$ is the kernel matrix.
- Recall DPPCA:

$$p(Y|X) = \prod_{i=1}^{n} \mathcal{N}(y_{(i)}|0, XX^T + \beta^{-1}I)$$

- $XX^T \in \mathbb{R}^{n \times n}$ can be changed to a kernel matrix $K \in \mathbb{R}^{n \times n}$ using a kernel trick, i.e., $XX^T \Rightarrow \Phi(X)\Phi(X)^T = K$ where $\Phi(\mathbf{x}_i)\Phi(\mathbf{x}_j)^T = k(\mathbf{x}_i, \mathbf{x}_j)$.

Gaussian process latent variable model

- Dual kernel probabilistic PCA (=GPLVM)
  - So far, we have seen that DPPCA can be extended to nonlinear embedding using nonlinear kernel function.
  - Dual probabilistic PCA likelihood:

$$p(Y|X) = \prod_{i=1}^{n} \mathcal{N}(y_{(i)}|0, XX^T + \beta^{-1}I)$$

  - GPLVM likelihood:

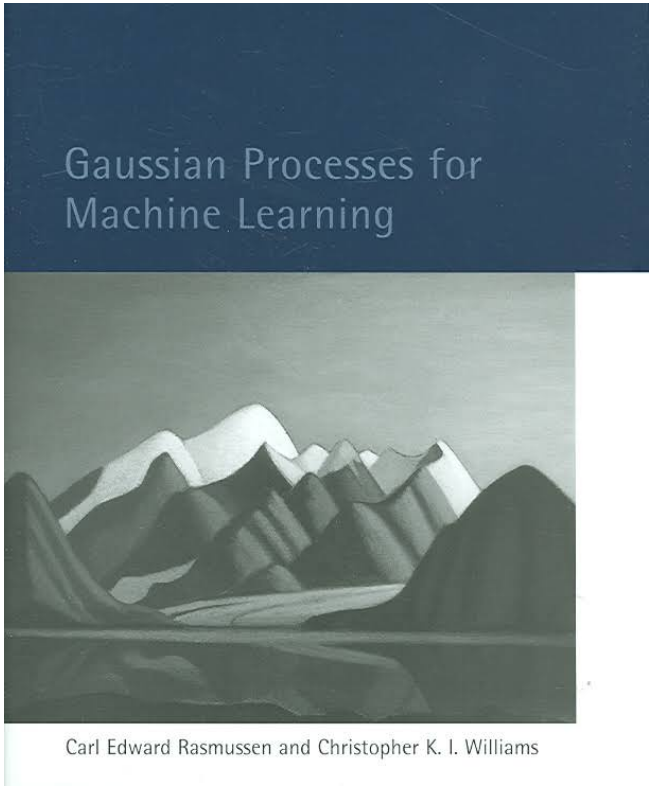
$$p(Y|X) = \prod_{i=1}^{n} \mathcal{N}(y_{(i)}|0, K)$$

Gaussian process latent variable model

- So far, we have seen that dual probabilistic PCA (DPPCA) can be extended to nonlinear embedding using a kernel trick.
- Likelihood of DPPCA:
  - $p(Y|X) = \prod_{i=1}^{d} \mathcal{N}(y_i|0, K)$
  - $\ln p(Y|X) = -\frac{d}{2} \ln |K| - \frac{nd}{2} \ln 2\pi - \frac{1}{2}\text{trace}(K^{-1}YY^T)$
- Partial derivatives:
  - $\frac{\partial L}{\partial x} = \frac{\partial L}{\partial K}\frac{\partial K}{\partial x}$
  - $\frac{\partial L}{\partial \theta} = \frac{\partial L}{\partial K}\frac{\partial K}{\partial \theta}$
- The solution for X can be optimized with a nonlinear optimizer such as scaled conjugate gradients.
- However, computing the inverse of $K \in \mathbb{R}^{n \times n}$ requires $O(n^3)$ computational complexity, making GPLVM impractical for large datasets.

Questions?

Backup slides



References i


C. E. Rasmussen and C. K. Williams.
Gaussian processes for machine learning, volume 1.
MIT press Cambridge, 2006.