Deep Gaussian Processes

Andreas C. Damianou and Neil D. Lawrence

AISTATS 2013

Presented by Esther Salazar
Duke University

August, 2013

Standard GP modelling

Let $\boldsymbol{X} \in \mathbb{R}^{N \times Q}$ and $\boldsymbol{Y} \in \mathbb{R}^{N \times D}$ a set of training input-output matrices, respectively. We seek to estimate the unobserved *latent* function $f = f(\boldsymbol{x})$ responsible for generating $\boldsymbol{Y}$ given $\boldsymbol{X}$

$$\mathbf{y}_n = f(\mathbf{x}_n) + \epsilon_n, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \sigma_\epsilon^2 \mathbf{I}),$$

and f is drawn from a GP $f(\boldsymbol{x}) \sim \mathcal{GP}(\mathbf{0}, k(x, x'))$, for example $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\sigma_{se})^2 \exp\left(-\frac{(x_i - x_j)^2}{2l^2}\right)$

The GP latent variable model provides an elegant solution by treating $\boldsymbol{X}$ as latent variables employing a product of D independent GPs as prior

Generative procedure

$$y_{nd} = f_d(\boldsymbol{x}_n) + \epsilon_{nd}$$

where $\boldsymbol{F} = \{\boldsymbol{f}_d\}_{d=1}^D$ with $f_{nd} = f_d(\boldsymbol{x}_n)$. Given a finite data set, the GP process priors take the form

$$p(\mathbf{F}|\mathbf{X}) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{f}_d|\mathbf{0}, \mathbf{K}_{NN})$$

and then obtain the likelihood $p(\mathbf{Y}|\mathbf{X}) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{y}_d|\mathbf{0}, \mathbf{K}_{NN} + \sigma_\epsilon^2 \mathbf{I})$

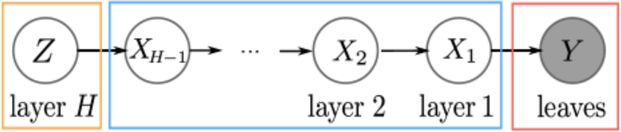
Deep Gaussian processes

Deep GP is a graphical model with three kinds of nodes:

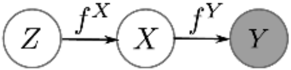
The leaf nodes: $\boldsymbol{Y} \in \mathbb{R}^{N \times D}$ (observed)

Intermediate latent space: $\boldsymbol{X}_h \in \mathbb{R}^{N \times Q_h}$, $h = 1 \ldots, H - 1$,
H is the number of hidden layers

Parent latent node: $\boldsymbol{Z} = \boldsymbol{X}_H \in \mathbb{R}^{N \times Q_z}$ (can be unobserved and potentially constrained with a prior)



Two hidden layer hierarchy



Generative process:

$$y_{nd} = f_d^Y(\mathbf{x}_n) + \epsilon_{nd}^Y, \quad d = 1, \ldots, D, \quad \mathbf{x}_n \in \mathcal{R}^Q$$
$$x_{nq} = f_q^X(\mathbf{z}_n) + \epsilon_{nq}^X, \quad q = 1, \ldots, Q, \quad \mathbf{z}_n \in \mathcal{R}^{Q_z}$$

where $f^Y \sim \mathcal{GP}(0, k^Y(\boldsymbol{X}, \boldsymbol{X}))$ and $f^X \sim \mathcal{GP}(0, k^X(\boldsymbol{Z}, \boldsymbol{Z}))$

Note that each layer adds a significant number of model parameters $\boldsymbol{X}_h$ since the size of each layer has to be a priori defined

Strategy: They seek to variationally marginalise out the whole latent space to significantly reduce the number of model parameters

The first step is to define automatic relevance determination (ARD) covariance functions for the GPs:

$$k\left(\mathbf{x}_i, \mathbf{x}_j\right) = \sigma_{ard}^2 e^{-\frac{1}{2}\sum_{q=1}^{Q} w_q \left(x_{i,q} - x_{j,q}\right)^2}$$

This function assumes a different weight w_q for each latent dimension, also can be used to “switch off” irrelevant dimensions by driving their weights to zero

Bayesian training and variational approximation

A Bayesian training requires optimization of

$$\log p(\boldsymbol{Y}) = \log \int_{\boldsymbol{X},\boldsymbol{Z}} p(\boldsymbol{Y}|\boldsymbol{X})p(\boldsymbol{X}|\boldsymbol{Z})p(\boldsymbol{Z})$$

In this paper, the authors take the general case where $p(\boldsymbol{Z}) = \mathcal{N}(\boldsymbol{Z}|\mathbf{0}, \boldsymbol{I})$

Bayesian training and variational approximation

Variational approximation

- They apply Jensen's inequality to find a variational lower bound $\mathcal{F}_v \leq \log p(\boldsymbol{Y})$
$$\mathcal{F}_v = \int_{\mathbf{X},\mathbf{Z},\mathbf{F}^Y,\mathbf{F}^X} \mathcal{Q} \log \frac{p(\mathbf{Y},\mathbf{F}^Y,\mathbf{F}^X,\mathbf{X},\mathbf{Z})}{\mathcal{Q}},$$
- Expand the joint distribution
$$p(\mathbf{Y},\mathbf{F}^Y,\mathbf{F}^X,\mathbf{X},\mathbf{Z}) = \\ p(\mathbf{Y}|\mathbf{F}^Y)p(\mathbf{F}^Y|\mathbf{X})p(\mathbf{X}|\mathbf{F}^X)p(\mathbf{F}^X|\mathbf{Z})p(\mathbf{Z}),$$
- Augment the probability space with K auxiliary inputs $\tilde{\boldsymbol{X}} \in \mathbb{R}^{K\times Q}$ and $\tilde{\boldsymbol{Z}} \in \mathbb{R}^{K\times Q_Z}$, also $\boldsymbol{U}^Y \in \mathbb{R}^{K\times D}$ and $\boldsymbol{U}^X \in \mathbb{R}^{K\times Q}$. Then, the augmented probability space is:
$$p(\mathbf{Y},\mathbf{F}^Y,\mathbf{F}^X,\mathbf{X},\mathbf{Z},\mathbf{U}^Y,\mathbf{U}^X,\tilde{\mathbf{X}},\tilde{\mathbf{Z}}) = \\ p(\mathbf{Y}|\mathbf{F}^Y)p(\mathbf{F}^Y|\mathbf{U}^Y,\mathbf{X})p(\mathbf{U}^Y|\tilde{\mathbf{X}}) \\ \cdot p(\mathbf{X}|\mathbf{F}^X)p(\mathbf{F}^X|\mathbf{U}^X,\mathbf{Z})p(\mathbf{U}^X|\tilde{\mathbf{X}})p(\mathbf{Z}) \qquad (9)$$

Bayesian training and variational approximation

- The variational distribution $\mathcal{Q}$ is

$$\begin{aligned}\mathcal{Q} =& p(\mathbf{F}^Y|\mathbf{U}^Y,\mathbf{X})q(\mathbf{U}^Y)q(\mathbf{X})\\ &\cdot p(\mathbf{F}^X|\mathbf{U}^X,\mathbf{Z})q(\mathbf{U}^X)q(\mathbf{Z}).\end{aligned}$$

where
$$q(\mathbf{X}) = \prod_{q=1}^{Q} \mathcal{N}(\boldsymbol{\mu}_q^X, \mathbf{S}_q^X),\ q(\mathbf{Z}) = \prod_{q=1}^{Q_Z} \mathcal{N}(\boldsymbol{\mu}_q^Z, \mathbf{S}_q^Z)$$

- Finally, the lower bound can be written as

$$\mathcal{F}_v = \int \mathcal{Q} \log \frac{p(\mathbf{Y}|\mathbf{F}^Y)p(\mathbf{U}^Y)p(\mathbf{X}|\mathbf{F}^X)p(\mathbf{U}^X)p(\mathbf{Z})}{\mathcal{Q}'}, \text{where } \mathcal{Q}' = q(\mathbf{U}^Y)q(\mathbf{X})q(\mathbf{U}^X)q(\mathbf{Z})$$

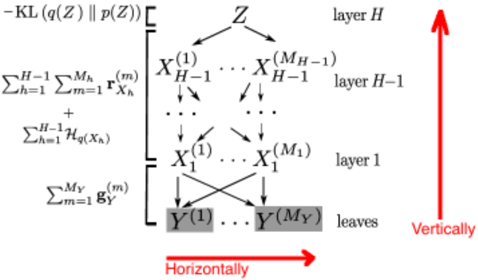
and also as

$$\boxed{\mathcal{F}_v = \mathbf{g}_Y + \mathbf{r}_X + \mathcal{H}_{q(\mathbf{X})} - \mathrm{KL}\left(q(\mathbf{Z}) \parallel p(\mathbf{Z})\right)}$$

$$\begin{aligned}\mathbf{g}_Y &= g(\mathbf{Y}, \mathbf{F}^Y, \mathbf{U}^Y, \mathbf{X})\\ &= \left\langle \log p(\mathbf{Y}|\mathbf{F}^Y) + \log \frac{p(\mathbf{U}^Y)}{q(\mathbf{U}^Y)} \right\rangle_{p(\mathbf{F}^Y|\mathbf{U}^Y,\mathbf{X})q(\mathbf{U}^Y)q(\mathbf{X})}\end{aligned}$$

$$\begin{aligned}\mathbf{r}_X &= r(\mathbf{X}, \mathbf{F}^X, \mathbf{U}^X, \mathbf{Z})\\ &= \left\langle \log p(\mathbf{X}|\mathbf{F}^X) + \log \frac{p(\mathbf{U}^X)}{q(\mathbf{U}^X)} \right\rangle_{p(\mathbf{F}^X|\mathbf{U}^X,\mathbf{Z})q(\mathbf{U}^X)q(\mathbf{X})q(\mathbf{Z})}\end{aligned}$$

Extending the hierarchy


$-\mathrm{KL}\left(q(Z)\,\|\,p(Z)\right)$
$\sum_{h=1}^{H-1}\sum_{m=1}^{M_h}\mathbf{r}_{X_h}^{(m)}$
$+$
$\sum_{h=1}^{H-1}\mathcal{H}_{q(X_h)}$
$\sum_{m=1}^{M_Y}\mathbf{g}_Y^{(m)}$
Z
layer H
$X_{H-1}^{(1)}\ \cdots\ X_{H-1}^{(M_{H-1})}$
layer $H{-}1$
$X_1^{(1)}\ \cdots\ X_1^{(M_1)}$
layer 1
$Y^{(1)}\ \cdots\ Y^{(M_Y)}$
leaves
Vertically
Horizontally

Variational bound for the most general version

$$\mathcal{F}_v = \sum_{m=1}^{M_Y}\mathbf{g}_Y^{(m)} + \sum_{h=1}^{H-1}\sum_{m=1}^{M_h}\mathbf{r}_{X_h}^{(m)} + \sum_{h=1}^{H-1}\mathcal{H}_{q(\mathbf{X}_h)}$$
$$-\,\mathrm{KL}\left(q(\mathbf{Z})\,\|\,p(\mathbf{Z})\right).$$

Experiments: Toy regression problem

- Two layers. The first GP employed a covariance function which was the sum of a linear and a quadratic exponential kernel
- 10 dimensional samples were generated
- 25 datapoints were randomly selected for the training set and left the rest for test

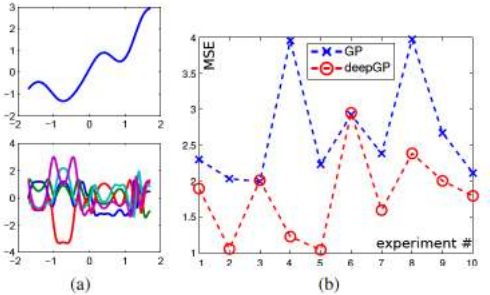


Figure 3: (a) shows the toy data created for the regression experiment. The top plot shows the (hidden) warping function and bottom plot shows the final (observed) output. (b) shows the results obtained over each experiment repetition.