Gaussian Process Models for Multi-Task Learning

Young Researchers' Meeting

Ayman Boustati

University of Warwick

June 13, 2017

Notation

- x is a scalar
- $\mathbf{x}$ is a column vector
- $\mathbf{X}$ is a matrix

Gaussian Processes

For Machine Learning

Definition

A Gaussian process is a collection of random variables any finite number of which is jointly Gaussian.

A Gaussian process $f(\mathbf{x})$ is denoted by:

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

Where $m(\mathbf{x})$ is a mean function and $k(\mathbf{x}, \mathbf{x}')$ is the covariance function or kernel, encoding our belief about the functional form of $f(\mathbf{x})$.

Gaussian Processes

The Covariance Function

Definition

A covariance function is a function that describes covariance of a random process.

A covariance function is a symmetric and positive semi-definite kernel:

$$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$$

Example

- Linear: $k_{Lin}(\mathbf{x}, \mathbf{x}') = \sum_{d=1}^{D} \sigma_d^2 x_d x_d'$
- Squared Exponential: $k_{SE}(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp(-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{2l^2})$
- Periodic: $k_{Per}(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp(-\frac{\sin^2(b\|\mathbf{x}-\mathbf{x}'\|)}{2l^2})$

Gaussian Processes

For Machine Learning

Given a set of training input-output pairs $(\mathbf{x}_i, y_i)$ for $i \in \{1, \ldots, N\}$, where $\mathbf{x}_i$'s are arranged in a design matrix $\mathbf{X}$ and y_i's arranged in a column vector $\mathbf{y}$. We can model the relationship between the inputs and outputs as follows:

$$y_i = f(\mathbf{x}_i) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$$

Where f is an unobserved, latent function.

Gaussian Processes

For Machine Learning

We can set a Gaussian process prior on the latent function f:

$$f(\mathbf{x}) \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}'))$$

Using Bayes' rule, we can obtain the posterior of f on a test input $\mathbf{x}^*$ as follows:

$$p(f(\mathbf{x}^*)|\mathcal{D}, \mathbf{x}^*, \boldsymbol{\theta}) = \mathcal{N}(\bar{f}_*, \mathrm{cov}(f_*))$$

Where,

$$\bar{f}_* = \mathbf{k}(\mathbf{x}^*, \mathbf{X})^T[\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_\epsilon^2 \mathbf{I}]^{-1}\mathbf{y}$$

$$\mathrm{cov}(f_*) = k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}(\mathbf{x}^*, \mathbf{X})^T[\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_\epsilon^2 \mathbf{I}]^{-1}\mathbf{k}(\mathbf{x}^*, \mathbf{X})$$

Gaussian Processes

For Machine Learning

To completely specify a Gaussian process f, we need to determine the values of the hyper-parameters (kernel parameters) $\boldsymbol{\theta}$. This can be easily done by type-II maximum likelihood, i.e. maximising the marginal likelihood (aka evidence). The log-marginal likelihood is given by:

$$\log p(\mathbf{y}|\mathbf{X},\theta) = -\frac{1}{2}\mathbf{y}^T[\mathbf{K}(\mathbf{X},\mathbf{X})+\sigma_\epsilon^2\mathbf{I}]^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}(\mathbf{X},\mathbf{X})+\sigma_\epsilon^2\mathbf{I}| - \frac{n}{2}\log(2\pi)$$

One can easily use off-the-shelf optimisation packages to find the value of θ that maximises the marginal likelihood.

Multitask Learning

An Introduction

Definition

Multitask learning is a machine learning framework where ones learns two or more tasks that share the same domain (input feature space) simultaneously.

The principal aim of Multitask learning is to improve the generalisation ability of the learner by leveraging domain-specific information contained in the training signals of related tasks.

Multitask Learning

Cases

- *Isotopic* case where all tasks share the same set of training inputs.
- *Hetrotopic* case where each task is associated with a different set of training inputs.
- *Partially Hetrotopic* case where tasks share some training inputs.

Gaussian Processes

For Machine Learning

Given a set of training input-output pairs $(\mathbf{x}_i, \mathbf{y}_i)$ for $i \in \{1, \ldots, N\}$ and $\mathbf{y}_i \in \mathbb{R}^D$. $\mathbf{x}_i$'s are arranged in a design matrix $\mathbf{X}$ and $\mathbf{y_i}$'s arranged in an $N \times D$ matrix $\mathbf{Y}$. We can model the relationship between the inputs and outputs as follows:

$$\mathbf{y}_i = \mathbf{f}(\mathbf{x}_i) + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{D}_\epsilon)$$

Where $\mathbf{f}$ is an unobserved, vector-valued, latent function; $\mathbf{f} = (f_1, \ldots, f_D)^T$.

Setting a Gaussian process prior on $\mathbf{f}$ can be done by assuming that each element f_d of $(f_1, \ldots, f_D)^T$ is a different random process where:

$$\text{cov}(f_d(\mathbf{x}), f_{d'}(\mathbf{x}')) = k((\mathbf{x}; d), (\mathbf{x}'; d'))$$

To make notation simpler we can write:

$$k((\mathbf{x}; d), (\mathbf{x}'; d')) = k_{d,d'}(\mathbf{x}, \mathbf{x}')$$

Gaussian Processes

For Machine Learning

To set a Gaussian process prior on the latent function $\mathbf{f}$, we write:

$$\mathbf{f}(\mathbf{x}) \sim \mathcal{GP}(\mathbf{0}, k_{d,d'}(\mathbf{x}, \mathbf{x}'))$$

Using Bayes' rule, we can obtain the posterior of f on a test input $\mathbf{x}^*$ as follows:

$$p(\mathbf{f}(\mathbf{x}^*)|\mathcal{D}, \mathbf{x}^*, \boldsymbol{\theta}) = \mathcal{N}(\bar{\mathbf{f}}_*, \text{cov}(\mathbf{f}_*))$$

Where,

$$\bar{\mathbf{f}}_* = \mathbf{K}(\mathbf{x}^*, \mathbf{X})^T [\mathbf{K}(\mathbf{X}, \mathbf{X}) + \boldsymbol{\Sigma}]^{-1} \text{vec}(\mathbf{Y})$$
$$\text{cov}(\mathbf{f}_*) = \mathbf{K}(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{K}(\mathbf{x}^*, \mathbf{X})^T [\mathbf{K}(\mathbf{X}, \mathbf{X}) + \boldsymbol{\Sigma}]^{-1} \mathbf{K}(\mathbf{x}^*, \mathbf{X})$$

Seperable Kernels

Introduction

We consider kernels of the form:

$$k_{d,d'}(\mathbf{x}, \mathbf{x}') = k_T(d, d')k(\mathbf{x}, \mathbf{x}')$$

We can also write this as a matrix expression:

$$\mathbf{K}(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}')\mathbf{B}$$

Where $\mathbf{B}$ is a $D \times D$ symmetric and positive semi-definite matrix.

Seperable Kernels

Simplest Case

We consider the simplest case of the previous formulation, where $\mathbf{B} = \mathbf{I}$:

$$k_T(d, d') = \delta_{d,d'}$$

Where δ is the Kronecker delta. This formulation corresponds to a Gram matrix that is block diagonal. This means that the D outputs are uncorrelated; however, they still share the kernel parameters.

Seperable Kernels

Intrinsic Coregionalisation Model

In ICM, we assume $\mathbf{B}$ is a free form $D \times D$ symmetric and positive semi-definite matrix. In this case:

$$k_T(d, d') = b_{d,d'}$$

Where $b_{d,d'}$ is the element in the dth row and d'th column of $\mathbf{B}$. This formulation corresponds to a Gram matrix that is block symmetric. This means that the D outputs are correlated and the cross-covariance between the dth and the d'th outputs is given by $b_{d,d'}$.

Seperable Kernels

Linear Model of Coregionalisation

LMC is a generalised case of IMC, in LMC we write the covariance as:

$$\mathbf{K}(\mathbf{x}, \mathbf{x}') = \sum_{q=1}^{Q} \mathbf{B}_q k_q(\mathbf{x}, \mathbf{x}')$$

Where $\mathbf{B}_q$'s are known as coregionalisation matrices.

Seperable Kernels

Notes

- In the isotropic case, the Gram matrix $\mathbf{K}$ can be written as a factorisation using the Kronecker product, i.e. $\tilde{\mathbf{K}} = \mathbf{B} \bigotimes \mathbf{K}(\mathbf{X}, \mathbf{X})$.
- $\mathbf{B}$ can be reparametrised in different ways e.g. PPCA where $\mathbf{B} = \mathbf{W}^T \mathbf{W} + \mathbf{D}$, or Cholesky decomposition where $\mathbf{B} = \mathbf{L}^T \mathbf{L}$. This can be important to insure numerical stability.

References

Alvarez, M., Rosasco, L., and Lawrence, N. D. (2011).
Kernels for Vector-Valued Functions: a Review.
Technical Report, MIT.

Bonilla, E. V., Chai, K. M., and Williams, C. (2007).
Multi-task Gaussian process prediction.
In *Advances in Neural Information Processing Systems*, pages 153–160, Vancouver, Canada.

Caruana, R. (1997).
Multitask Learning.
Machine Learning, 28(1):41–75.

Rasmussen, C. E. and Williams, C. K. I. (2006).
Gaussian processes for machine learning.
Adaptive computation and machine learning. MIT Press, Cambridge, Mass.
OCLC: ocm61285753.

Skolidis, G. and Sanguinetti, G. (2011).
Bayesian Multitask Classification With Gaussian Process Priors.
IEEE Transactions on Neural Networks, 22(12):2011–2021.