Gaussian Process for Machine Learning

YongKyung Oh

2019.06.07

School of Management Engineering, UNIST

Contents

Overview and Introduction

Overview

- Perspective of Machine Learning
  - **Machine learning** is concerned with the design of algorithms enabling machines to learn. The goal of machine learning is to build computer systems that can adapt and learn from their experience. (Dietterich, 1999)
  - **Learning** is understood as automatic extraction of general rules about the population from a small sample in order to make predictions and decisions.
  - All learning algorithms perform equally well if averaged over all possible learning problems. Thus, **prior knowledge** or **prior assumptions** on the particular problem at hand like smoothness of the underlying function are indispensable for successful learning.

- Problem addressed by Machine Learning
  - Supervised Learning
  - Unsupervised Learning
  - Reinforcement Learning

Overview

Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It means having a full set of labeled data while training an algorithm (Russell, S. & Norvig, P., 2010).

- Two approaches (Williams, C. K. & Rasmussen, C. E., 2006)
  - The first is to restrict the class of functions that we consider. The first approach has an obvious problem in that we have to decide upon the richness of the class of functions considered.
  - The second approach is to give a prior probability to every possible function, where higher probabilities are given to functions that we consider to be more likely, for example because they are smoother than other functions.

- In that surely there are an uncountably infinite set of possible functions, and how are we going to compute with this set in finite time?

Introduction

"parametric methods require the observations within each group to have an approximately Normal distribution . . . if the raw data do not satisfy these conditions . . . a non-parametric method should be used" (Altman, D. G., 1990)

- Parametric Statistics
  - **Parametric statistics** are any statistical approaches based on underlying assumptions about data's distribution. Because parametric statistics are based on the normal curve, data must meet certain assumptions.
  - In parametric statistics, we agree on a function class indexed by a finite number of parameters. A distribution over these parameters induces an ensemble over functions.

- Nonparametric Statistics
  - **Nonparametric statistics** are not based on the parameters of the normal curve. Though nonparametric statistical tests have more flexibility than do parametric statistic, nonparametric approaches are not as robust.
  - In non-parametric statistics, regularities of the relationship are postulated without requiring the ensemble to be concentrated on a describable class.

Introduction

Gaussian distribution (also known as normal distribution) is assumed that during any measurement values will follow a normal distribution with an equal number of measurements above and below the mean value.

$$p(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\} \sim \mathcal{N}(x|\mu,\sigma^2)$$

For multivariate case, we can expand the variables into N-dimensional vector.

$$p(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{N/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\} \sim \mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma})$$

Gaussian process as a distribution over functions. A Gaussian process is a generalization of the Gaussian probability distribution. Just as a multivariate normal distribution is completely specified by a mean vector and covariance matrix, a GP is fully specified by a mean function and a covariance function:

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

Gaussian process regression is a nonparametric Bayesian regression method using the properties of Gaussian processes.

Introduction

Adopting a set of Gaussians confers a number of advantages and properties.

$$p(x,y) = \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} \Sigma_x & \Sigma_{xy} \\ \Sigma_{xy}^T & \Sigma_y \end{bmatrix}\right)$$

- Normalization: the density function normalizes into 1:

$$\int_x p(x|\mu_x, \Sigma_x)dx = 1$$

- Marginalization: marginal distribution of any subset of elements from a multivariate normal distribution is also normal:

$$p(x) = \int_y p(x,y)dy = \mathcal{N}(\mu_x, \Sigma_x)$$

- Conditioning: conditional distributions of a subset of the elements of a multivariate normal distribution are also normal:

$$p(x|y) = \frac{p(x,y|\mu_x,\Sigma_x)}{\int_y p(x,y|\mu_x,\Sigma_x)dy} = \mathcal{N}(\mu_x + \Sigma_{xy}\Sigma_y^{-1}(y-\mu_y), \Sigma_x - \Sigma_{xy}\Sigma_y^{-1}\Sigma_{xy}^T)$$

Gaussian Processes

Gaussian Processes

- Random Field
  - **Random field** is a mapping from an parameter space to real-valued random variables, a natural generalization of a joint distribution to an infinite index set.
  - Random field is simply a stochastic process, taking values in a Euclidean space, and defined over a parameter space of dimensionality at least one.
  - Given a parameter space X, a stochastic process u over X is a collection of random variables $\{u(x) : x \in X\}$. If X is a set of dimension N, and the random variables $u(x)$ are all vector valued of dimension d, then we call the vector valued random field u a (N, d) random field, where X is index set (e.g. time $\mathbb{R}$ or space $\mathbb{R}^3$).

- The Kolmogorov Extension Theorem says, essentially, that one can get a process on $\mathbb{R}^N$ for parameter space X being an arbitrary, non-empty index set, by specifying all finite dimensional distributions (f.d.d.) in a "consistent" way.
- In other words, the distributional properties of a (N, d) random field over X are determined by its family of f.d.d.'s.

Gaussian Processes

- Stationary Processes
  - A process is called **homogeneous** (or **stationary**) if its f.d.d.'s are invariant under simultaneous translation of their variables.
  - Suppose that u is an (N, d) random field defined over all of $\mathbb{R}^N$. Suppose furthermore that the mean function $m(x)$ is constant, and that the covariance function $K(x, x')$ is a function of the difference x, x' only. Then we say that u is homogeneous or stationary.

- Isotropic Processes
  - A stationary process is called **isotropic**, if it is also true that $K(x, x')$ is a function of the euclidean distance $|x - x'|$ only. Then we say that u is isotropic process.
  - In this case, the spectral distribution F is invariant under isotropic isomorphisms (e.g., rotations). Loosely speaking, second-order characteristics of an isotropic process are the same from whatever position and direction they are observed. This definition does not depend on the coordinate system.

Gaussian Processes

A Gaussian process (GP) is a process whose f.d.d.'s are Gaussian. There are two elementary views on Gaussian processes, the **function space view** and the **weight space view**. While the former is usually much simpler to work with, the latter allows us to relate GP models to parametric linear models rather directly.

We can now define a real valued Gaussian (random) field or Gaussian (random) process to be a random process u on a parameter set X for which the (finite dimensional) distributions of $u_{x_1} \dots u_{x_n}$ are multivariate Gaussian for each $1 \leq n < \infty$ and each $(x_1, \dots x_n) \in X^n$, where X is index set (e.g. time $\mathbb{R}$ or space $\mathbb{R}^3$).

Suppose that $u(x)$ is a random process. Since multivariate Gaussian distributions are determined by means and covariances, it is immediate that Gaussian random fields are determined by their mean and covariance functions, defined by

$$m(x) = \mathbb{E}[u(x)]$$

$$k(x, x') = \mathbb{E}[(u(x) - m(x))(u(x') - m(x'))]$$

Gaussian Processes

A Gaussian Process is a collection of random variables, any finite number of which have (consistent) joint Gaussian distributions (Rasmussen, C. E., 2003). It is fully specified by its mean function $m(x)$ and covariance function $k(x, x')$.

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

Assume that we have a training set $\mathcal{D} = \{(x_i, u_i), i = 1 : N\}$, using noise-free observations of $u_i = u(x_i)$. We will predict test output(u_*) using test set (x_*).

$$\begin{pmatrix} \mathbf{u} \\ \mathbf{u}_* \end{pmatrix} = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}_* \end{bmatrix}, \begin{bmatrix} \mathbf{K} & \mathbf{K}_* \\ \mathbf{K}_*^{\mathbf{T}} & \mathbf{K}_{**} \end{bmatrix} \right)$$

It is possible to get the conditional probability of one of the variables given the other, and this is how, in a GP, we can derive the posterior from the prior and our observations. Therefore, the posterior has the following form:

$$p(\mathbf{u}_*|\mathbf{X}_*, \mathbf{X}, \mathbf{u}) = \mathcal{N}(\mathbf{u}_*|\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$$

$$\boldsymbol{\mu}_* = \mu(\mathbf{x}_*) + \mathbf{K}_*^{\mathbf{T}}\mathbf{K}^{-\mathbf{1}}(\mathbf{y} - \mu(\mathbf{x}))$$

$$\boldsymbol{\Sigma}_* = \mathbf{K}_{**} - \mathbf{K}_*^{\mathbf{T}}\mathbf{K}^{-\mathbf{1}}\mathbf{K}_*$$

Two views on Gaussian Process

Weight-space view

Weight-space view Suppose linear regression with function value $u(\mathbf{x}) = \mathbf{x}^T\mathbf{w}$ where $\mathbf{w}$ is vector of weight. Observed target value $y = u(x) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$. Assume that we have a training set $\mathcal{D} = \{(x_i, u_i), i = 1 : N\}$ and define $X = [x_1^T; \ldots; x_n^T]$. Then,

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

Goal of linear regression is to find $\mathbf{w}$ such that $(\|\mathbf{y} - X^T\mathbf{w}\|)^2$ is minimize.

Solution is $\hat{\mathbf{w}} = (XX^T)^{-1}X\mathbf{y}$.

Weight-space view

Bayesian formulation: Put a prior over the parameters, i.e., $\mathbf{w} \sim \mathcal{N}(0, \Sigma_p)$

Finding the posterior distribution is the goal of a Bayesian method:
$p(\mathbf{w}|\mathbf{y}, X) = \frac{p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w})}{p(\mathbf{y}|X)}$.

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

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

Hence,

$$p(\mathbf{w}|\mathbf{y}, X) \sim \mathcal{N}(\bar{\mathbf{w}}, A^{-1})$$

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.

Weight-space view

For bayesian formulation, **Kernel Trick**, which is projections of inputs into feature space, can be applied. Bayesian linear model which suffers from limited expressiveness. A very simple idea to overcome this problem is to first project the inputs into some high dimensional space using a set of basis functions and then apply the linear model in this space instead of directly on the inputs themselves.

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

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

Solution that maximizes the likelihood distribution, maximum likelihood estimation (MLE) solution:

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

Function Space View

A Gaussian Process is fully specified by its mean function $m(x)$ and covariance function $k(x, x')$.

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

Assume that we have a training set $\mathcal{D} = \{(x_i, u_i), i = 1 : N\}$, using noise-free observations of $u_i = u(x_i)$. We will predict test output(u_*) using test set (x_*).

$$\begin{pmatrix} \mathbf{u} \\ \mathbf{u}_* \end{pmatrix} = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}_* \end{bmatrix}, \begin{bmatrix} \mathbf{K} & \mathbf{K}_* \\ \mathbf{K}_*^{\mathbf{T}} & \mathbf{K}_{**} \end{bmatrix} \right)$$

It is possible to get the conditional probability of one of the variables given the other, and this is how, in a GP, we can derive the posterior from the prior and our observations. Therefore, the posterior has the following form:

$$p(\mathbf{u}_*|\mathbf{X}_*, \mathbf{X}, \mathbf{u}) = \mathcal{N}(\mathbf{u}_*|\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$$

$$\boldsymbol{\mu}_* = \mu(\mathbf{x}_*) + \mathbf{K}_*^{\mathbf{T}}\mathbf{K}^{-\mathbf{1}}(\mathbf{u} - \mu(\mathbf{x}))$$

$$\boldsymbol{\Sigma}_* = \mathbf{K}_{**} - \mathbf{K}_*^{\mathbf{T}}\mathbf{K}^{-\mathbf{1}}\mathbf{K}_*$$

Function Space View

Assume that the observation is noisy: $y = u(x) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma)$.

$$cov(y_n, y_m) = k(x_n, x_m) + \sigma_y^2 \delta_{nm}$$
$$cov(\mathbf{y}|\mathbf{X}) = \mathbf{K} + \sigma_y^2 \mathbf{I_N} \approxeq \mathbf{K_y}$$

Assume that we have a training set $\mathcal{D} = \{(x_i, u_i), i = 1 : N\}$, using noisy observations of $y_i = y(x_i)$. We will predict test output(u_*) using test set (x_*).

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{u_*} \end{pmatrix} = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}_* \end{bmatrix}, \begin{bmatrix} \mathbf{K_y} & \mathbf{K}_* \\ \\ \mathbf{K}_*^\mathbf{T} & \mathbf{K}_{**} \end{bmatrix} \right)$$

By the standard rule for conditional Gaussians, the posterior has the following form. For simplicity, assume zero-mean distribution.

$$p(\mathbf{u}_*|\mathbf{X}_*, \mathbf{X}, \mathbf{u}) = \mathcal{N}(\mathbf{u}_*|\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$$
$$\boldsymbol{\mu}_* = \mu(\mathbf{x}_*) + \mathbf{K}_*^\mathbf{T} \mathbf{K}_\mathbf{y}^{-1}(\mathbf{y})$$
$$\boldsymbol{\Sigma}_* = \mathbf{K}_{**} - \mathbf{K}_*^\mathbf{T} \mathbf{K}_\mathbf{y}^{-1} \mathbf{K}_*$$

Function Space View

Estimate GP prior for observations. Recall that we have some points x for which for which we have observed the outcome $u(x)$. There are some poinst x_* for which we would like to estimate $u(x_*)$. We can sample from the joint gaussian distribution of u and u_*.

$$u_* \sim \mathcal{N}(\mu_*, \Sigma_*)$$

We have a univariate distribution $x \sim \mathcal{N}\left(\mu, \sigma^2\right)$, you can express this in relation to standard normals, i.e. as $x \sim \mu + \sigma(\mathcal{N}(0,1))$ We need the equivalent way to express our multivariate normal distribution in terms of standard normals:

$$u_* \sim \mu + \mathbf{L}\mathcal{N}(0, I)$$

where $\mathbf{L}\mathbf{L}^T = \Sigma_*$, i.e. the square root of our covariance matrix. We can use something called a Cholesky decomposition to find this.

For noisy observation with zero mean case:

$$p(\mathbf{u}_*|\mathbf{X}_*, \mathbf{X}, \mathbf{u}) = \mathcal{N}(\mathbf{u}_*|\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$$

We apply the Cholesky decompostion: $\mathbf{K_y} = \mathbf{K} + \sigma_y^2\mathbf{I_N} = \mathbf{L}\mathbf{L}^T$.

Function Space View

Performance of gaussian process prediction is depend on the kernel. Log-likelihood is used to train/optimize the kernel parameters(θ).

For noise-free observation with zero mean case: :

$$L = \log \mathcal{N}(\mathbf{u}|0, \mathbf{K}) = \log p(\mathbf{u}|\mathbf{X}, \theta) = -\frac{1}{2}\log|\mathbf{K}| - \frac{1}{2}\mathbf{u}^{\mathbf{T}}\mathbf{K}^{-\mathbf{1}}\mathbf{u} - \frac{N}{2}\log(2\pi)$$

For noisy observation with zero mean case:

$$L = \log \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K_y}) = \log p(\mathbf{y}|\mathbf{X}, \theta) = -\frac{1}{2}\log|\mathbf{K_y}| - \frac{1}{2}\mathbf{y}^{\mathbf{T}}\mathbf{K_y}^{-\mathbf{1}}\mathbf{y} - \frac{N}{2}\log(2\pi)$$

Train to minimize the objective function ($J(\theta)$) using gradient optimizer.

$$J(\theta) = \log p(\mathbf{y}|\mathbf{X}, \theta)$$

$$\frac{\partial}{\partial\theta_j}\log p(\mathbf{y}|\mathbf{X}) = -\frac{1}{2}tr(\mathbf{K_y}^{-\mathbf{1}}\frac{\partial\mathbf{K_y}}{\partial\theta_j}) + \frac{1}{2}\mathbf{y}^{\mathbf{T}}\mathbf{K_y}^{-\mathbf{1}}\frac{\partial\mathbf{K_y}}{\partial\theta_j}\mathbf{K_y}^{-\mathbf{1}}\mathbf{y}$$

Reproducing Kernel Hilbert Spaces

Kernel

Mapping function betweeing input space and feature space. e.g. mapping function $\Phi(x) : \mathbb{R}^2 -> \mathbb{R}^6$

$$\Phi : (x_1, x_2) \rightarrow (x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, 1)$$

Kernel: The kernel or null space of some linear transformation, between two vector spaces is the set of all vectors. In essence, the kernel is a collection of all elements that are sent to zero by the transformation.

Kernel Trick: $K(x_i, x_j) = \Phi(x_i)^T \Phi(x_j)$ Assume $\Phi(x)$ is Linear transformation and standard matrix A. We can define $\Phi(x) = Ax$.

$$K(x_i, x_j) = \Phi(x_i)^T \Phi(x_j) = x_i^T A^T A x_j$$

Output of K is scalar. It means $K(x_i, x_j) = K(x_j, x_i)$. Therefore, $K(x_i, x_j)$ should be semi-definite matrix and symmetric matrix.

Kernel

Mercer's theorem Suppose $K(s,t)$ is a symmetric (that is, $K(s,t) = K(t,s)$), continuous, and nonnegative definite kernel function on $[a,b] \times [a,b]$. Mercer's theorem asserts that there is an orthonormal set of eigenfunctions $\Phi_j(x)$ and eigenvalues λ_j such that

$$K(s,t) = \sum_{j}^{\infty} \lambda_j \Phi_j(s) \Phi_j(t),$$

where the values and functions satisfy the integral eigenvalue equation

$$\lambda_j \Phi_j(s) = \int_a^b K(s,t) \Phi_j(t)$$

For example, assume Gaussain kerenl with $2\sigma^2 = 1$,

$$\begin{aligned} K(x_1, x_2) &= exp\left\{-(x_1 - x_2)^2\right\} \\ &= exp(-x_1) exp(-x_2) exp(2x_1 x_2) \end{aligned}$$

By adapting taylor series expansion: $exp(2x_1x_2) = \sum_{k=0}^{\infty} \frac{2^k x_1^k x_2^k}{k!}$

It can be interpreted that any input space can be mapping with (infinite) feature space.

Reproducing Kernel Hilbert Spaces

A Hilbert space H is a complete inner product space. We will see that a reproducing kernel Hilbert space (RKHS) is a Hilbert space with extra structure that makes it very useful for statistics and machine learning.

Assume function $f : \mathcal{X} \to \mathbb{R}$ which is $\|f\|_p \equiv \left(\int_{\mathcal{X}} |f(x)|^p dx\right)^{\frac{1}{p}} < \infty$ in L^pspace. In the case of $p = 2$,

$$\langle f, g \rangle \equiv \left(\int_{\mathcal{X}} f(x)g(x)dx\right)^{\frac{1}{2}}$$

Reproducing Property of Kernel: Consider Hilbert space H of functions $f : \mathcal{X} \to \mathbb{R}$ and kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that $k(\cdot, x) \in H, \forall x \in \mathcal{X}$. The reproducing property is defined by,

$$\langle f, k(\cdot, x) \rangle_H = f(x), \forall x \in \mathcal{X}$$

Reproducing Kernel Hilbert Spaces

Reproducing kernel Hilbert space Let H be a Hilbert space of real functions f defined on an index set $\mathcal{X}$. Then H called a reproducing kernel Hilbert space endowed with an inner product $\langle \cdot, \cdot \rangle_H$ and norm $\|f\|_H = \sqrt{\langle f, f \rangle_H}$, if there exists a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ with the following properties:

1. for every x, $k(x, x')$ as a function of x') belongs to H
2. k has the reproducing property, $\langle f, k(\cdot, x) \rangle_H = f(x), \forall x \in \mathcal{X}$

- For every positive definite function $k(\cdot, \cdot)$ on there exists a unique RKHS, and vice versa. The RKHS norm is in fact solely a property of the kernel and is invariant under this change of measure.
- The key intuition behind the RKHS formalism is that the squared norm $\|f\|_H^2$ can be thought of as a generalization to functions of the n-dimensional quadratic form $\mathbf{f}^T K^{-1} \mathbf{f}$.

Reproducing Kernel Hilbert Spaces

Consider a real positive semidefinite kernel $k(x, x')$ with with an eigenfunction expansion $k(x, x') = \sum_{i=1}^{N} \lambda_i \Phi_i(x) \Phi_i(x')$ relative to a measure μ. Hilbert space between functions $f(x) = \sum_{i=1}^{N} f_i \Phi_i(x)$ and $g(x) = \sum_{i=1}^{N} g_i \Phi_i(x)$ is defined as

$$\langle f, g \rangle_H = \sum_{i=1}^{N} \frac{f_i g_i}{\lambda_i}$$

$$\langle f(\cdot), k(\cdot, x) \rangle_H = \sum_{i=1}^{N} \frac{f_i \lambda_i \Phi_i(x)}{\lambda_i} = f(x)$$

$$\langle k(\cdot, x'), k(\cdot, x) \rangle_H = \sum_{i=1}^{N} \frac{\lambda_i \Phi_i(x) \Phi_i(x')}{\lambda_i} = k(x, x')$$

Duality between RKHS and Gaussian Process

$u(x)$ is zero mean GP with covariance function K. In other words, We sample the coefficients u_i in the eigenexpansion $u(x) = \sum_{i=1}^{N} u_i \Phi_i(x)$ from $\mathcal{N}(0, \lambda_i)$.

$$\mathbb{E}[\|u\|_K^2] = \sum_{i=1}^{N} \frac{\mathbb{E}[u_i^2]}{\lambda_i} = \sum_{i=1}^{N} 1 = N \to \infty \quad \text{when} \quad N \to \infty$$

A better intuition about H_K is that it will turn out to contain expected values of $u(x)$ conditioned on a finite amount of information, thus the posterior mean functions we are interested in.

$$\langle u(x), u(x')) \rangle_{GP} = \mathbb{E}[u(x)u(x')] = k(x, x') = \langle k(\cdot, x'), k(\cdot, x) \rangle_K$$

For most purposes, we can regard H_{GP} as RKHS with RK K. The space H_{GP} is important in the context of inference on GP models we are interested in, because it contains exactly the random variables we condition on or would like to predict in situations where only a finite amount of information is available.

Some of Gaussian Process Model

Gaussian Process Model

Generalized Linear Model Generalized linear model (GLM) is a flexible generalization of ordinary linear regression that allows for response variables that have error distribution models other than a normal distribution.

Weighted sum of fixed finite set of M basis functions (or link functions):

$$f(x) = \sum_{m=1}^{M} w_m \phi(x) = \mathbf{w}^T \boldsymbol{\phi}(x)$$

Place Gaussian prior on weights: $p(\mathbf{w}) = \mathcal{N}(0, \boldsymbol{\Sigma}_w$. Defines GP with finite rank M (degenerate) covariance function:

$$K(x, x') = \mathbb{E}[f(x)f(x')] = \phi^T(x)\boldsymbol{\Sigma}_w\phi(x')$$

According to Mercer's theorem, we can always decompose covariance function into eigenfunctions and eigenvalues. General GP — can specify covariance function directly rather than via set of basis functions.

Gaussian Process Model

Multilayer Perceptron (MLP)

To rid of the limitation of linear regression solution, MLP (or Neural Netwokr) applies non-linear functions such as the Sigmoid and inverse tangent functions onto linear functions. Thus a perceptron is created and took the form $h(x) = g(f(x))$ where $g(z)$ is a non-linear transformation of a linear function $f(x) = w^T x$.

Neural net with one hidden layer of N_H units, with bounded hidden layer transfer function.

$$h(x) = b + \sum_{j=1}^{N_H} v_j f(x; w_j)$$

If, v's and b zero mean independent, and weights w_j i.i.d., then central limit theorem implies $NN \to GP$ as $N_H \to \infty$. (Neal, 1996)

Gaussian Process Model

Spline models

The equivalent between Gaussian processes and smoothing splines has been shown in Kimeldorf and Wahba 1970.

Univariate cubic spline has cost functional:

$$\sum_{n=1}^{N}(f(x_n) - y_n)^2 + \lambda \int_0^1 f''(x)^2 dx$$

- Can derive a spline covariance function, and full GP machinery can be applied to spline regression
- Penalties on derivatives — equivalent to specifying the inverse covariance function

Gaussian Process Model

Kriging

Kriging is a type of regression that gives a least squares estimate of data (Remy et. al, 2011). It uses z-scores to generate an estimated surface model from the spatial description of a scattered set of data points. It is a method of interpolation for which the interpolated values are modeled by a Gaussian process governed by prior covariances. Under suitable assumptions on the priors, kriging gives the best linear unbiased prediction of the intermediate values.

The basic model is the same as for semiparametric smoothing:

$$z(x) = m(x)^T \beta + \epsilon(x)$$

where $m(x)$ is is a known feature map and $\epsilon(x)$ is a zero-mean random field with covariance function K. Kriging is a minimum mean squared error prediction method for linear functionals of $z(x)$ given observations $z = (z(x_1), \ldots z(x_n))^T$ at spatial locations x_i. While $m(x)^T \beta$ globally approximates the design space, $\epsilon(x)$ creates "localized" deviations so that the kriging model interpolate n sampled data.

Choice of Kernel and Kernel Design

Choice of Kernel and Kernel Design

RBF (Gaussian) kernel

$$k_{\text{RBF}}(\mathbf{x_1}, \mathbf{x_2}) = \exp\left(-\frac{1}{2}(\mathbf{x_1} - \mathbf{x_2})^\top \Theta^{-1}(\mathbf{x_1} - \mathbf{x_2})\right)$$

Matern kernel

$$k_{\text{Matern}}(\mathbf{x_1}, \mathbf{x_2}) = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\sqrt{2\nu}d\right) K_\nu\left(\sqrt{2\nu}d\right)$$

Where $d = (\mathbf{x_1} - \mathbf{x_2})^\top \Theta^{-1}(\mathbf{x_1} - \mathbf{x_2})$, ν is a smoothness parameter and K_ν is a modified Bessel function.

Linear kernel

$$k_{\text{Linear}}(\mathbf{x_1}, \mathbf{x_2}) = v\mathbf{x_1}^\top \mathbf{x_2}$$

Periodic kernel

$$k_{\text{Periodic}}(\mathbf{x_1}, \mathbf{x_2}) = \exp\left(\frac{2\sin^2(\pi\|\mathbf{x_1} - \mathbf{x_2}\|_1/p)}{\ell^2}\right)$$

Polynomial kernel

$$k_{\text{Poly}}(\mathbf{x_1}, \mathbf{x_2}) = (\mathbf{x_1}^\top \mathbf{x_2} + c)^d.$$

Implementation

Implementation

1. Gaussian process basic
2. Gaussian process prior and estimation without known function
3. Gaussian process training
4. Gaussian process regression with different kernel
5. Gaussian process regression on real data

Reference

Reference

- Seeger, M. (2004). Gaussian processes for machine learning. International journal of neural systems, 14(02), 69-106.
- Williams, C. K., & Rasmussen, C. E. (2006). Gaussian processes for machine learning (Vol. 2, No. 3, p. 4). Cambridge, MA: MIT Press.
- Rasmussen, C. E. (2003, February). Gaussian processes in machine learning. In Summer School on Machine Learning (pp. 63-71). Springer, Berlin, Heidelberg.
- Do, C. B. (2007). Gaussian processes. Stanford University, Stanford, CA, accessed Dec, 5, 2017.
- Nickisch, H. (2010). Bayesian inference and experimental design for large generalised linear models (Doctoral dissertation, Berlin Institute of Technology).
- Altman, D. G. (1990). Practical statistics for medical research. CRC press.
- Russell, S., & Norvig, P. (2010). Artificial Intelligence: A Modern Approach. Third Edit.
- Toby Driscoll (2011), Mercer's theorem and the Karhunen-Loeve expansion, http://www.chebfun.org/examples/stats/MercerKarhunenLoeve.html
- Chris Fonnesbeck (2017), Fitting Gaussian Process Models in Python, https://blog.dominodatalab.com/fitting-gaussian-process-models-python/
- Martin Krasser (2018), Gaussian Processes, http://krasserm.github.io/2018/03/19/gaussian-processes/

End of Presentation