10-708: Probabilistic Graphical Models, Spring 2015

20 : Gaussian Process

Lecturer: Andrew Gordon Wilson *Scribes: Yuetao Xu, Haohan Wang, Jisu Kim*

1 Three Views of Regression

1.1 Deterministic

Now we have a basic regression problem, defined as following:

- Training set of N targets (observations) $y = (y(x_1), ...y(x_N))^T$, here $(y(x_1), ...y(x_N))$ could be written as $f(x, w)$
- Observations evaluated at inputs $X = (x_1, ..., x_N)^T$
- Task: Predict value of $y(x_\star)$ at a test input $x_\star$

Deterministically, we could solve this problem with optimizing the error function defined as follows:

$$E(w) = \sum_{i=1}^{N} (f(x_i, w) - y_i)^2$$

1.2 Maximum Likelihood

As another approach, we could look at this problem probabilistically.
We could explicity account for noise in our model, with $\epsilon(x) = N(0, \sigma^2)$ defined as a noise term, as following:

$$y(x) = f(x, w) + \epsilon(x)$$

So, we can have the model as:

$$p(y|x, w, \sigma^2) = N(y(x); f(x, w), \sigma^2)$$

Then the likelihood is:

$$p(y|x, w, \sigma^2) = \prod_{i=1}^{N} N(y(x_i); f(x_i, w), \sigma^2)$$

Then we maximize the likelihood with respect to σ^2 and w
For both deterministic approach and probabilistic (Maximum likelihood) approach, regularization could be considered to avoid overfitting.

1.3 Bayesian

From the basic Bayesian approach, which is

$$posterior = \frac{likelihood \times prior}{marginallikelihood}$$

We have the following form for our model,

$$p(w|y, X, \sigma^2) = \frac{p(y|X, w, \sigma^2)p(w)}{p(y|X, \sigma^2)}$$

Thus, we can have the predictive distribution as following:

$$py|x_\star, y, X = \int p(y|x_\star, w)p(w|y, X)dw$$

It is worth noting that Bayesian approach could automatically calibrated the complexity of models, so it does not have the problem of overfitting.
It is also interesting to notice that this problem is solved as a optimization problem in Maximum Likelihood approach, but as a marginalization problem in Bayesian approach.

1.4 Model Selection and Marginal Likelihood

The ability for a model to learn from data depends on its:

- Support: what solutions we think are priori possible
- Inductive biases: what solutions we think a priori likely

There is a trade off of these two abilities. As in Figure 1, it shows how the evidence discourages overcomplex models, and can be used to select the most probable model. We can see that when we favor for the support, we are enlarging the space which the model covers, and it will result in a complex model. Instead, when we favor inductive bias, we are shrinking the data sets that the model covers, and it will result in a simple model. Complex models achieve modest evidence while simple models reach high evidences, but only for a limited set of data.

1.5 Occam's Razor Example

1.5.1 A very intuitive example

Andrew Willison has drawn a very simple and intuitive example to explain Occam's Razor, as showed and explained in Figure 2. This example shows us how Occam's Razor works, it favors the simpler model that could explain the data.

1.5.2 A simple regression example

As another example, we try to figure out what is next number of this sequence: $-1, 3, 7, 11, ...$
If we use y_x to denote the xth number, where $x = 1, 2, 3, ...,$
Hypothesis 1: $y_x = 5x - 4$
Hypothesis 2: $y_x = \frac{1}{11}x^3 + \frac{9}{11}x^2 + \frac{23}{11}$
Then, we can get that $P(H_1|D) \approx 10^{-1}$, while $P(H_2|D) \approx 2.5 \times 10^{-12}$. Thus, we should favor Hypothesis 1.

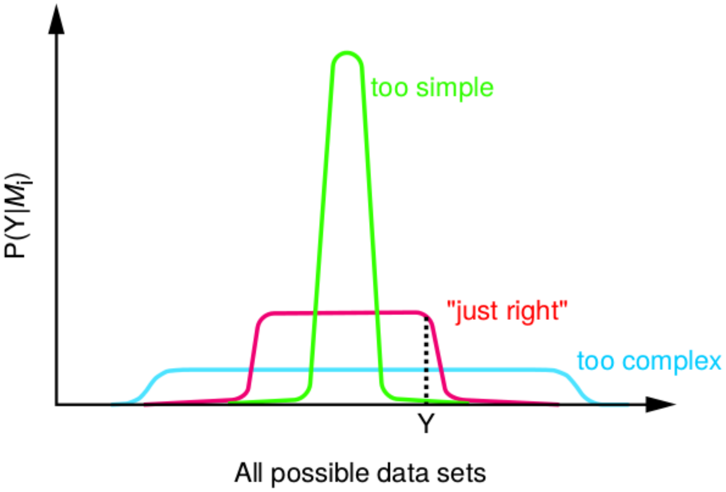


Figure 1: Relationship between evidence and complexity of models

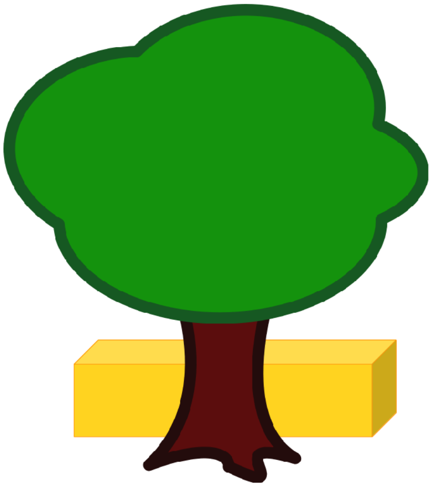


Figure 2: A very simple and intuitive example to explain Occam's Razor: Box behind a tree. When someone finds such a scene in daily life, probably he/she will believe that is a regular-shaped box behind the tree. However, many hypothesis could be made to describe this scene. For example, some stupid guy could believe that there are two boxes behind the tree, but the gap between these gaps are blocked by the tree, so we cannot tell, or some crazy guys could believe that there is a tree-shape box blocked by the tree. However, our experience tells that the the most believing hypothesis is the simplest one that can explain this scene, i.e, there is only one regular-shaped box behind the tree.

2 From Linear Model to Gaussian Process

2.1 Linear Model

Let's start from a simple linear model:

$$f(x) = a_0 + a_1 x$$

where x is the input (and a scalar), a_0 is the bias and a_1 is the slope. We put a prior belief on the parameters a_0 and a_1, so $a_0, a_1 \sim \mathcal{N}(0, 1)$.

We are interested in the function value $f(x)$, for any given x, not the parameters. First, let's consider its expectation, $\mathbb{E}[f(x)]$:

$$\mathbb{E}[f(x)] = \mathbb{E}[a_0 + a_1 x] = \mathbb{E}[a_0] + \mathbb{E}[a_1]x = 0$$

Then we consider the covariance between two function values $f(x_a)$ and $f(x_b)$, for any pair of x_a and x_b:

$$\begin{aligned}\mathrm{Cov}[f(x_a), f(x_b)] =& \mathbb{E}[f(x_a)f(x_b)] - \mathbb{E}[f(x_a)]\,\mathbb{E}[f(x_b)] \\ =& \mathbb{E}[(a_0 + a_1 x_a)(a_0 + a_1 x_b)] - 0 \\ =& \mathbb{E}[a_0^2] + \mathbb{E}[a_0 a_1](x_a + x_b) + \mathbb{E}[a_1^2]x_a x_b \\ =& 1 + 0 + 1 \cdot x_a x_b \\ =& 1 + x_a x_b\end{aligned}$$

With the expectation function $\mathbb{E}[f(x)]$ and the covariance function $\mathrm{Cov}[f(x_a), f(x_b)]$, we can define a joint Gaussian distribution for the set of function values $f(x_1), \cdots, f(x_N)$ of any finite set of input values $x_1 \cdots x_N$.

$$\begin{aligned}f(x_1), \cdots, f(x_N) \sim& \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}) \\ \boldsymbol{\mu}_i =& \mathbb{E}[x_i] = 0 \\ \mathbf{K}_{ij} =& \mathrm{Cov}[f(x_i, x_j)] = 1 + x_i x_j\end{aligned}$$

By definition, $f(x)$ is a Gaussian process.

2.2 Definition of Gaussian Process

A Gaussian process (GP) is a collection of random variables, e.g. $\{f(\mathbf{x}) | \, \mathbf{x} \in \mathbb{R}^d\}$, any finite number of which have a joint Gaussian distribution, e.g. $f(\mathbf{x}_1) \cdots f(\mathbf{x}_N)$. $f(\mathbf{x}) \sim \mathcal{GP}(m, k)$ denotes that $f(\mathbf{x})$ is a Gaussian process that is controlled by the mean function m and the covariance function k. From the definition we know that, for any finite set of input $\mathbf{x}_1 \cdots \mathbf{x}_N$,

$$\begin{aligned}f(\mathbf{x}_1) \cdots f(\mathbf{x}_N) \sim& \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}) \\ \boldsymbol{\mu}_i =& m(\mathbf{x}_i) \\ \mathbf{K}_{ij} =& k(\mathbf{x}_i, \mathbf{x}_j)\end{aligned}$$

In the previous example, $m(x_i) = 0$ and $k(x_i, x_j) = 1 + x_i x_j$.

2.3 Linear Basis Function Models

We can project $\mathbf{x}$ into feature space by a function $\boldsymbol{\phi}(\mathbf{x})$. This projection allows us to apply linear model to non-linear features instead of the input directly. For example, let $\boldsymbol{\phi}(x) = (1, x, x^2, x^3, \cdots)^T$, then we can do polynomial regression. Therefore a linear model with such projection can be written as:

$$\begin{aligned} f(\mathbf{x}) &= \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) \\ \mathbf{w} &\sim \mathcal{N}(0, \Sigma_w) \end{aligned}$$

$f(\mathbf{x})$ is a Gaussian process, since

$$\begin{aligned} f(\mathbf{x}) &\sim \mathcal{GP}(m, k) \\ m(\mathbf{x}) &= \mathbb{E}[f(x)] = \mathbb{E}[\mathbf{w}^T]\, \boldsymbol{\phi}(\mathbf{x}) = 0 \\ k(\mathbf{x}_i, \mathbf{x}_j) &= \mathrm{Cov}[f(\mathbf{x}_i), f(\mathbf{x}_j)] = \boldsymbol{\phi}(\mathbf{x}_i)^T \Sigma_w\, \boldsymbol{\phi}(\mathbf{x}_j) \end{aligned}$$

As long as the the prior belief of $\mathbf{w}$ has zero mean, $m(\mathbf{x}) = 0$. Therefore in this case k alone defines the Gaussian process. In the following sections we assume $\mathbf{w}$ is zero-meaned.

2.4 Inference

With k defined, training samples $\mathbf{X}$ and $\mathbf{y}$ observed, we are interested in inferring $\mathbf{f}_*$, given a set of new inputs $\mathbf{X}_*$. Note that $\mathbf{y}$ is the observation value with noise introduced, therefore we assume $\mathbf{y} = \mathbf{f} + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon}$ is the noise and $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2 I)$. As a result, $\mathbf{y} \sim \mathcal{N}(0, \mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 I)$, where $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. To infer $\mathbf{f}_*$ given observation $\mathbf{X}$, $\mathbf{y}$ and query $\mathbf{X}_*$, we write down the joint distribution of $\mathbf{y}$ and $\mathbf{f}_*$. By the definition of Gaussian process, it is a Gaussian distribution. Therefore, we have:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \begin{bmatrix} \mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 I & \mathbf{K}(\mathbf{X}, \mathbf{X}_*) \\ \mathbf{K}(\mathbf{X}_*, \mathbf{X}) & \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix})$$

Therefore the conditional distribution $p(\mathbf{f}_* \mid \mathbf{y})$ can be written as:

$$\begin{aligned} \mathbf{f}_* \mid \mathbf{X}_*, \mathbf{X}, \mathbf{y} &\sim \mathcal{N}(\bar{\mathbf{f}}_*, \mathrm{cov}(\mathbf{f}_*)) \\ \bar{\mathbf{f}}_* &= \mathbf{K}(\mathbf{X}_*, \mathbf{X})[\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 I]^{-1}\, \mathbf{y} \\ \mathrm{cov}(\mathbf{f}_*) &= \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) - \mathbf{K}(\mathbf{X}_*, \mathbf{X})[\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 I]^{-1}\, \mathbf{K}(\mathbf{X}, \mathbf{X}_*) \end{aligned}$$

From the equation we can see that we do not need to explicitly compute $\mathbf{w}$ and $\boldsymbol{\phi}$. This allows us to use infinite number of features, as long as we provide a kernel function k.

2.5 Example: RBF Kernel

RBP Kernel is an example of such kernel which has infinite features. Each feature i can be written as:

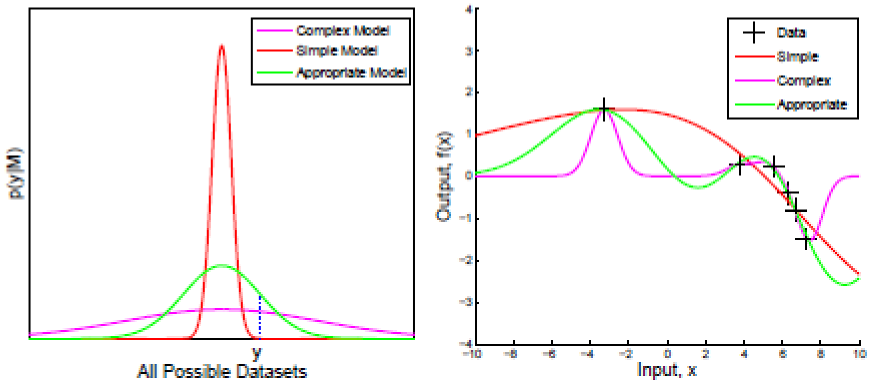
$$\boldsymbol{\phi}_i(\mathbf{x}) = \sigma^2 \exp(-\frac{||\, \mathbf{x} - \mathbf{c}_i \,||_2^2}{2l^2})$$

Suppose there are J features, the kernel function can be written as:

$$k(\mathbf{x}_p, \mathbf{x}_q) = \frac{\sigma^2}{J} \sum_{i=1}^{J} \boldsymbol{\phi}_i(\mathbf{x}_p)\, \boldsymbol{\phi}_i(\mathbf{x}_q)$$

Let $J \to \infty$ and let $\mathbf{c}$ distributed at a constant interval $\Delta\, \mathbf{c} \to \mathbf{0}$, we get the RBF kernel:

$$k_{RBF}(\mathbf{x}_i, \mathbf{x}_j) = a^2 \exp(-\frac{||\, \mathbf{x}_i - \mathbf{x}_j \,||_2^2}{2l^2})$$



(a) (b)

Figure 3: Panel (a) shows schematic of the behavior of the evidence for different model complexities. A simple model can only account for a limited range of possible sets of target values, but since the marginal likelihood must normalize to unity, the data sets which the model does account for have a large value of the marginal likelihood. A complex model is the converse. Panel (b) shows output $f(x)$ for different model complexities. [Originally from Lecture #21 slides, page 38]

The intuition behind RBF kernel is that nearby inputs are more correlated. l controls how the correlation decays with respect to the nearness (L2 distance). A large l indicates a slower decay, so further points will have more correlation. Therefore the resulting $\mathbf{f}$ will be more smoothed. On the other hand, small l indicates a rapid decay, so only the nearest inputs are correlated. Hence the resulting $\mathbf{f}$ vibrates more violently.

3 More on Gaussian Process

3.1 Learning and Model Selection

Posterior $p(\mathcal{M}_i|\,\mathbf{y})$ is factorized as

$$p(\mathcal{M}_i|\,\mathbf{y}) = \frac{p(\mathbf{y}\,|\mathcal{M}_i)p(\mathcal{M}_i)}{p(\mathbf{y})}.$$

The marginal likelihood $p(\mathbf{y}\,|\mathcal{M}_i)$ is also called as evidence of model, and is given by

$$p(\mathbf{y}\,|\mathcal{M}_i) = \int p(\mathbf{y}\,|\,\mathbf{f}, \mathcal{M}_i)p(\mathbf{f})d\,\mathbf{f}\,.$$

Figure 3 illustrates why the evidence doesn't simply favor the models that fit the training data the best.

3.1.1 Learning as Likelihood Optimization over Hyperparameters

For learning θ, we can integrate away the entire Gaussian process $f(x)$ to obtain the marginal likelihood, as a function of kernel hyperparameters θ alone:

$$p(\mathbf{y}\,|\theta, \mathbf{X}) = \int p(\mathbf{y}\,|\,\mathbf{f}, \mathbf{X})p(\mathbf{f}\,|\theta, \mathbf{X})d\,\mathbf{f}\,.$$

Since we have already seen that $\mathbf{y}\,|\theta, \mathbf{X} \sim \mathcal{N}(0,\ K_\theta + \sigma^2 I)$, log marginal likelihood is

$$\log p(\mathbf{y}\,|\theta, \mathbf{X}) = \overbrace{-\frac{1}{2}\mathbf{y}^T(K_\theta + \sigma^2 I)^{-1}\mathbf{y}}^{\text{model fit}} - \overbrace{\frac{1}{2}\log|K_\theta + \sigma^2 I|}^{\text{complexity penalty}} - \frac{N}{2}\log(2\pi).$$

Maximizing above likelihood is used as an extremely powerful mechanism for kernel learning.

3.1.2 Learning by fully Bayesian Treatment

For inferring f_*, a fully Bayesian treatment would integrate away kernel hyperparameters θ:

$$p(\mathbf{f}_*\,|\,\mathbf{X}_*, \mathbf{X}, \mathbf{y}) = \int p(\mathbf{f}_*\,|\,\mathbf{X}_*, \mathbf{X}, \mathbf{y}, \theta)p(\theta|\,\mathbf{y})d\theta.$$

For example, MCMC can be used to find

$$p(\mathbf{f}_*\,|\,\mathbf{X}_*, \mathbf{X}, \mathbf{y}) \approx \frac{1}{J}\sum_{i=1}^{J} p(\mathbf{f}_*\,|\,\mathbf{X}_*, \mathbf{X}, \mathbf{y}, \theta^{(i)}),\ \theta^{(i)} \sim p(\theta|\,\mathbf{y})$$

For non-Gaussian noise model case, f cannot be integrated away, and the strong dependencies between f and θ makes sampling extremely difficult. In Andrew's experience, the most effective solution is to use a deterministic approximation for the posterior $p(\mathbf{f}\,|\,\mathbf{y})$ which enables one to work with an approximate marginal likelihood.

3.2 Gaussian Process: Covariance Kernel and Graphical Model

3.2.1 Gaussian Process Covariance Kernels

There are several different choices for the covariance kernel in Gaussian Process. Let $\tau = x - x'$:

$$\begin{aligned}
k_{SE}(\tau) &= \exp\left(-0.5\tau^2/l^2\right)\\
k_{MA}(\tau) &= a\left(1 + \frac{\sqrt{3}\tau}{l}\right)\exp\left(-\frac{\sqrt{3}\tau}{l}\right)\\
k_{RQ}(\tau) &= \left(1 + \frac{\tau^2}{2\alpha l^2}\right)^{-\alpha}\\
k_{PE}(\tau) &= \exp\left(-2\sin^2(\pi\tau\omega)/l^2\right)
\end{aligned}$$

3.2.2 Gaussian Process Graphical Model

A graphical model representation of a GP is given in Figure 4. This figure also illustrates why addition of further inputs x_* and unobserved targets $\mathbf{y}_*$ does not change the distribution of any other variables.

3.3 A Example of $\mathbf{CO_2}$ Prediction

The data consists of monthly average atmospheric CO_2 concentrations derived from air samples collected at the Mauna Loa Observatory, Hawaii, between 1958 and 2003. The data is shown in Figure 5. Our goal is to model the CO_2 concentration as a function of time x.

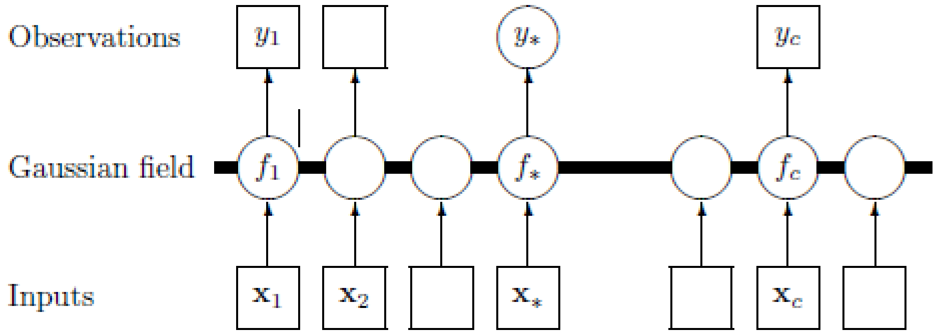


Figure 4: Graphical model for a GP for regression. Squares are observed variables, circles are latent. and the think horizontal bar is a set of fully connected nodes. Note that each $\mathbf{y}_i$ is conditionally independent given $\mathbf{f}_i$. Because of the marginalization property of GP, addition of further inputs x_* and unobserved targets $\mathbf{y}_*$ does not change the distribution of any other variables. [Originally Figure 2.3 from Rasmussen and Williams (2006)]

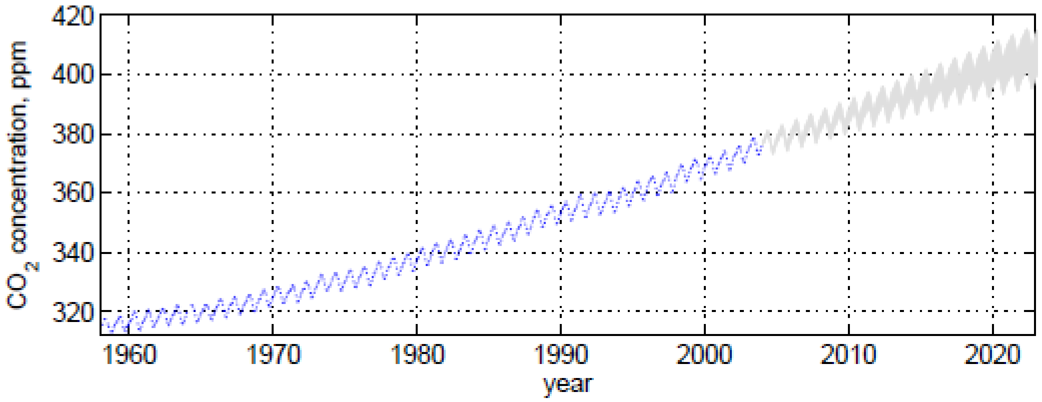


Figure 5: The 545 observations of monthly averages of the atmospheric concentration of CO_2 made between 1958 and the end of 2003, together with 95% predictive confidence region for a Gaussian process regression model, 20 years into the future. Rising trend and seasonal variations are clearly visible. Note also that the confidence interval gets wider the further the predictions are extrapolated. [Originally Figure 5.6 from Rasmussen and Williams (2006)]

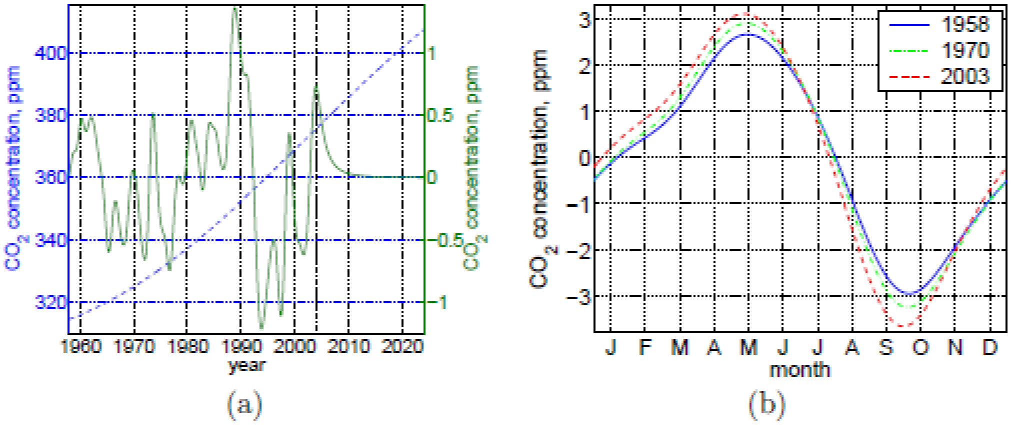


Figure 6: Panel (a): long term trend, dashed, left hand scale; superimposed is the medium term trend, full line, right hand scale. Panel (b) shows the seasonal variation over the year for three different years. [Originally Figure 5.7 from Rasmussen and Williams (2006)]

Several different features are encoded into different covariance functions:

- Long rising trend: $k_1(x_p, x_q) = \theta_1^2 \exp\left(-\frac{(x_p-x_q)^2}{2\theta_2^2}\right)$[1]

- Quasi-periodic seasonal changes:

$$\begin{aligned} k_2(x_p, x_q) &= k_{RBF}(x_p, x_q) k_{PER}(x_p, x_q) \\ &= \theta_3^2 \exp\left(-\frac{(x_p - x_q)}{2\theta_4^2} - \frac{2\sin^2(\pi(x_p - x_q))}{\theta_5^2}\right) \end{aligned}$$

- Multi-scale medium term irregularities: $k_3(x_p, x_q) = \theta_6^2\left(1 + \frac{(x_p-x_q)^2}{2\theta_8\theta_7^2}\right)^{-\theta_8}$

- Correlated and i.i.d. noise: $k_4(x_p, x_q) = \theta_9^2 \exp\left(-\frac{(x_p-x_q)^2}{2\theta_{10}^2}\right) + \theta_{11}^2\delta_{pq}$.

And then these covariance functions are combined together:

$$k_{total}(x_p, x_q) = k_1(x_p, x_q) + k_2(x_p, x_q) + k_3(x_p, x_q) + k_4(x_p, x_q).$$

The mean predictions for long term trend and seasonal variation is illustrated in Figure 6.

In this GP example, confidence in the extrapolation is high, which suggests that model is well specified. The learned hyperparameters θ can be interpreted to learn information about our dataset. Also, this dataset is well studied and a lot of interesting pattern recognition has been done by human. We would like to automate this modeling procedure.

[1] Despite its clear rising linear trend, we are not using linear trend. This is since when test data is very far from training data, we want function to eventually return to 0.

3.4 Non-Gaussian Likelihoods

Suppose our noise model is non-Gaussian. For inferring $\mathbf{f}_*$, we can no longer analytically integrate away the Gaussian process. But we can use a simple Monte carlo sum:

$$\begin{aligned} p(\mathbf{f}_* \,|\, \mathbf{y}, \mathbf{X}, x_*) &= \int p(\mathbf{f}_* \,|\, \mathbf{f}, x_*) p(\mathbf{f} \,|\, \mathbf{y}) d\,\mathbf{f} \\ &\approx \frac{1}{J} \sum_{j=1}^{J} p(\mathbf{f}_* \,|\, \mathbf{f}^{(j)}, x_*), \;\; \mathbf{f}^{(j)} \sim p(\mathbf{f} \,|\, \mathbf{y}). \end{aligned}$$

To sample from $p(\mathbf{f} \,|\, \mathbf{y})$, we can use slice sampling. See Elliptical slice sampling, Murray et.al. AISTATS 2010.

For learning hyperparameters θ, it's easy to implement Gibbs sampling:

$$\begin{aligned} p(f|\,\mathbf{y}, \theta) \propto p(\mathbf{y}\,|f) p(f|\theta) \\ p(\theta|f, \mathbf{y}) \propto p(f|\theta) p(\theta). \end{aligned}$$

But this won't work because of strong correlations between $\mathbf{f}$ and θ. There are several strategies as listed below:

- Transform into a whitened space, $f = L\nu$, and sample from ν and θ, which decouples correlations.
- Use a deterministic approach to approximately integrate away f to get a marginal likelihood $p(\mathbf{y}\,|\theta)$:

$$p(\mathbf{y}\,|\theta) = \int p(\mathbf{y}\,|f) p(f|\theta) df.$$

- The Laplace approximation, e.g., approximates $p(f|\,\mathbf{y})$ as a Gaussian.