

STATS 211: Statistical Methods II

Lecture 6: Gaussian process models

Babak Shahbaba

UCI, Winter 2012

Gaussian process models

- To introduce this concept, we start with a simple linear regression model.
- Recall that we presented a linear regression model as

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots \beta_p x_{ip} + \epsilon_i$$

- Using normal priors (with mean zero, and in general, different variances) for β 's

$$\beta_j | \sigma_j \sim N(0, \sigma_j^2) \quad j = 0, \dots, p$$

Gaussian process models

- In prior, β has a $(p + 1)$ dimensional multivariate normal distribution

$$\beta | \Sigma_\beta \sim N(0, \Sigma_\beta)$$

- ϵ also has an n dimensional multivariate normal distribution

$$\epsilon | \Sigma_\epsilon \sim N(0, \Sigma_\epsilon)$$

- To obtain the distribution of y we multiply β by the matrix x and add ϵ to it.
- Based on the properties of multivariate normal distribution, the resulting distribution would still be multivariate normal $N(0, C)$ where

$$C = x \Sigma_\beta x^T + \Sigma_\epsilon$$

Gaussian process models

- This gives us the prior distribution on the function $y(x)$.
- Since any finite subset of $y(x)$ (e.g., for the n observed cases) would have a Gaussian distribution, the prior distribution on $y(x)$ is a *Gaussian process*.
- Similar to the Gaussian distribution, the Gaussian process is also defined by its mean (here, the mean is 0 in prior) and its covariance function C .
- For the above linear model, the elements of C are

$$C_{ij} = \text{Cov}(y_i, y_j) = \sigma_0^2 + \sum_{u=1}^p x_{iu}x_{ju}\sigma_u^2 + \delta_{ij}\sigma_\epsilon^2$$

where δ_{ij} is equal to 1 if $i = j$, and 0 otherwise.

Gaussian process models

- Setting up the model this way, we are putting the prior directly on the relationship between x and y as opposed to on some parameters that represent this relationship (i.e., we cut out the middleman).
- This is specially useful if our objective is to predict future cases as opposed to making inference about the relationship between x and y .
- Note that the prior here is implicit and reflects our choice of the functional form.
- In the above example, we are assuming the relationship is linear. In general, we could use other covariance functions, C , to create nonlinear relationship.

Gaussian process for nonlinear regression

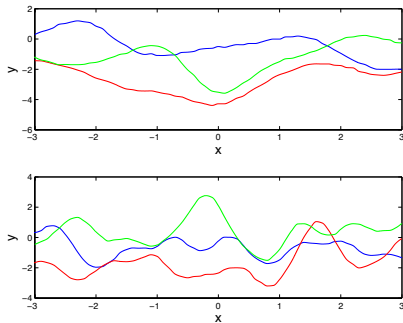
- For example, the following covariance function is very useful and includes a wide range of smooth nonlinear functions:

$$\text{Cov}(y_i, y_j) = \lambda^2 + \eta^2 \exp \left(- \sum_{u=1}^p \rho_u^2 (x_{iu} - x_{ju})^2 \right) + \delta_{ij} \sigma_\epsilon^2$$

- The constant part is used to make sure the model fit functions where the mean of y is not zero (the x matrix does not have a vector of 1's anymore). However, it is better to center y before analysis so we don't have to use a large constant.
- There is one ρ for each predictor. These parameters play an important role in identifying which predictor is relevant to the prediction task (see Neal, 1998, for details).
- The noise parameter, σ_ϵ^2 (also called *jitter*), is essential to improve the computation.

The effect of parameters in the covariance function

- By using different η , ρ 's, λ and σ_ϵ , we can generate a large variety of functions. The following graphs shows samples from two different priors: the first graph is based on a prior with $\eta = 1$, $\rho = 1$, $\lambda = 1$, and $\sigma_\epsilon = 0.01$, and in the second one we use the same priors except $\rho = 2$. As we mentioned before, it is better not to sample directly from multivariate normal and use the Cholesky decomposition instead.



Prediction

- As mentioned above, using a Gaussian process prior is especially useful if our goal is predicting future cases for which we only know the value of predictors, \tilde{x} .
- Assume that we have observed (x, y) for n cases, and we want to predict \tilde{y} for a new observation with predictor values \tilde{x} .
- Since the covariance function depends on x , we can find C_{n+1} for n the training cases and the new observation, i.e., for $\begin{pmatrix} x \\ \tilde{x} \end{pmatrix}$. To avoid confusion we denote the covariance matrix for just the training cases as C_n .
- We can write down C_{n+1} as follows:

$$C_{n+1} = \begin{pmatrix} C_n & K \\ K^T & v \end{pmatrix}$$

where K is the $n \times 1$ covariance vector between \tilde{y} and the n observed y . v is the prior variance of \tilde{y} obtained based on the covariance function C .

Prediction

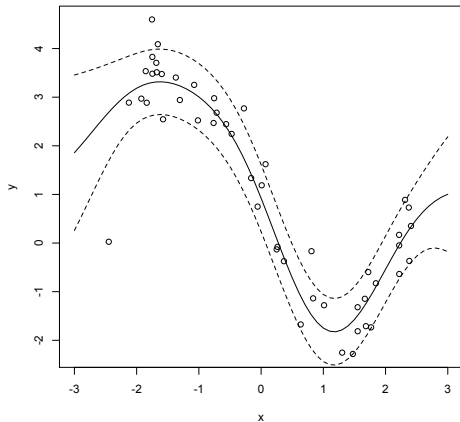
- Based the above setting, we can obtain the posterior predictive distribution for the new case.
- This distribution is also Gaussian with the following mean and variance:

$$\begin{aligned}E(\tilde{y}|y) &= K^T C_n^{-1} y \\ \text{Var}(\tilde{y}|y) &= v - K^T C_n^{-1} K\end{aligned}$$

- If we need a point estimate, we can use $E(\tilde{y}|y)$.
- We use the Cholesky decomposition to obtain the inverse of C .

Example

- The following example shows a Gaussian process model trained on 100 data points uniformly sampled from -2 to 2 .



Example

- For the above model, we used the following covariance function:

$$\text{Cov}(y_i, y_j) = 2 + \exp(-0.5(x_i - x_j)^2) + \delta_{ij} \times 0.1$$

- The solid line is expected function based on a grid test points between -3 and 3.
- The dashed lines show the 95% interval for predictions.

Hyperparameters

- In reality, we might not have enough information to fix the parameters of the covariance functions.
- In general, we would treat these parameters (e.g., η , ρ 's, λ and σ_ϵ) as hyperparameters.
- Therefore, we need to use MCMC simulations in order to obtain samples from the posterior distributions of these hyperparameters, and as usual, we integrate over these posterior distributions to obtain the posterior predictive probabilities (not discussed in this class).