Approximating the Likelihood for the Hyper-parameters in Gaussian Process Regression

Advisor: Professor Radford Neal

Chunyi Wang
Department of Statistics,
University of Toronto

Graudate Student Research Day
April 28th, 2011

Gaussian Process Regression: Model

We observe n training cases $(x_1, y_1), ..., (x_n, y_n)$ where x_i is a vector of inputs of length p, and y_i is the corresponding scalar response, which we assume is a function of the inputs plus some noise:

$$y_i = f(x_i) + \epsilon_i$$

where $\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$

In a Gaussian Process Regression model, the prior mean of the function f is 0, and the covariance of the response is

$$\text{Cov}(y_i, y_j) = k(x_i, x_j) + \sigma^2 \delta_{ij}$$

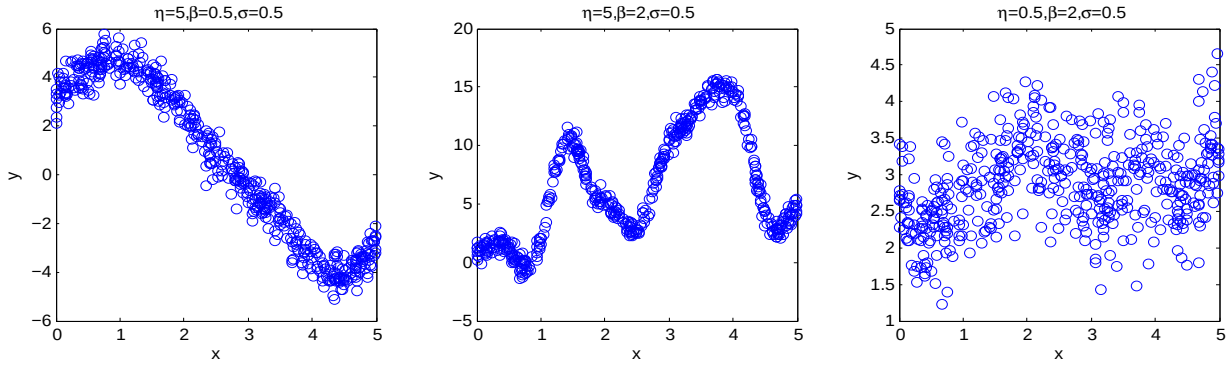
Gaussian Process Regression: Covariance Function

Any covariance function that leads to non-negative definite covariance matrices is allowed, such as the squared exponential:

$$k(x_i, x_j) = \eta^2 \exp(-\beta^2 ||x_i - x_j||^2)$$

η, β are unknown parameters that are estimated from the data.

Illustration of GP data with different hyper-parameter values:



Gaussian Process Regression: Prediction

We wish to predict the response y_*, for a test case x_* based on the training cases.The predictive distribution for the response y_* is Gaussian:

$$E[y_*|y] = k^T C^{-1} y$$

$$Var[y_*|y] = v - k^T C^{-1} k$$

where C is the covariance matrix for the training responses, k is the vector of covariances between y_* and each of y_i, and v is the prior variance of y^*, [*i.e.* $\mathrm{Cov}(y^*, y^*)$].

To do this in the Bayesian framework, we obtain a random sample from the posterior density for the hyper-parameter θ:

$$\pi(\theta|y) \propto (2\pi)^{-\frac{n}{2}} \det(C)^{-1} \exp\left(-\frac{1}{2} y^T C^{-1} y\right) \pi(\theta)$$

where $\pi(\theta)$ is the prior for θ.

Complexity for the GP Regression Model

The posterior density is

$$\pi(\theta|y) \propto (2\pi)^{-\frac{n}{2}} \det(C)^{-1} \exp\left(-\frac{1}{2} y^T C^{-1} y\right) \pi(\theta)$$

The time needed to perform the following major computations are (asymptotically, with an implementation-specific constant coefficient):

C	pn^2
$\det(C)$	n^3
C^{-1}	n^3
$y^T C^{-1} y$	n^2

In practice we compute C (pn^2), and the Cholesky decomposition of C (n^3), then we can cheaply obtain $\det(C)$ and $y^T C^{-1} y$.

Markov Chain Monte Carlo Methods

We construct a ergodic Markov Chain with transition $T(x'|x)$ which leaves the target distribution $\pi(x)$ invariant, *i.e.*

$$\int \pi(x)T(x'|x)dx = \pi(x')$$

Metropolis algorithm: propose to move from x to x^* (according to a proposal distribution $S(x^*|x)$), accept the proposal with probability $\min[1, \pi(x^*)/\pi(x)]$. This satisfies the detailed balance condition

$$\pi(x)T(x'|x) = \pi(x')T(x|x')$$

and thus the chain (called reversible) will leave the target distribution π invariant.

MCMC with Temporary Mapping

We can combine three stochastic mappings $\hat{T}, \bar{T}$ and $\check{T}$ to form the transition $T(x'|x)$, as follows:

$$x \xrightarrow{\hat{T}} y \xrightarrow{\bar{T}} y' \xrightarrow{\check{T}} x'$$

where $x \in \mathcal{X}$ is the original sample space and $y \in \mathcal{Y}$ is a temporary space.
To leave the target distribution π invariant these mappings have to satisfy

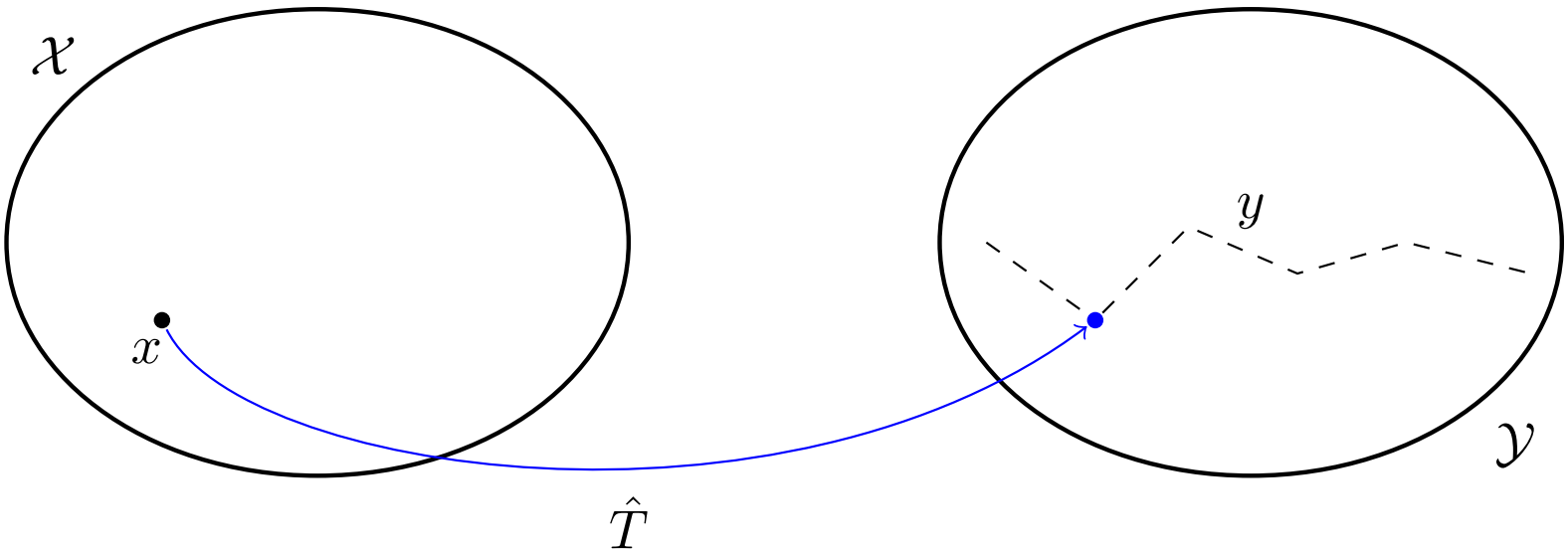
$$\int \pi(x)\hat{T}(y|x)dx = \rho(y)$$

$$\int \rho(y)\bar{T}(y'|y)dy = \rho(y')$$

$$\int \rho(y')\check{T}(x'|y')dy' = \pi(x')$$

Mapping to a Discretizing Chain

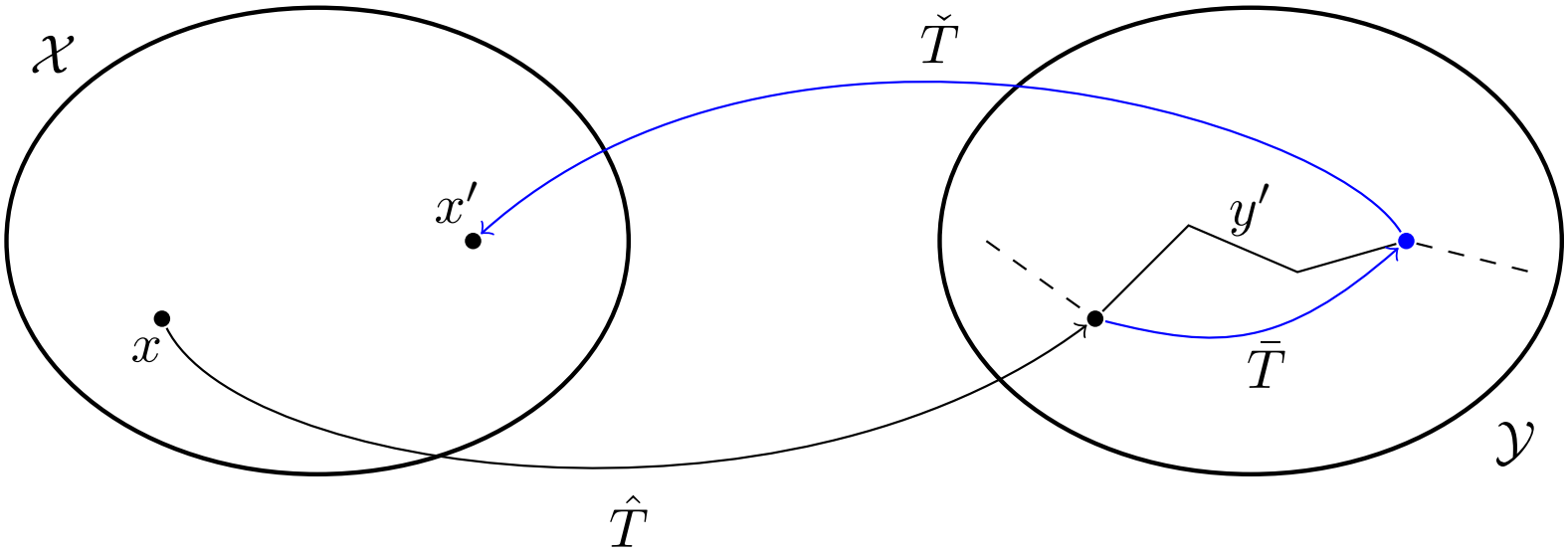
Suppose we have a Markov Chain which leaves a distribution π^* invariant. We can map to a space of realizations of such a chain. The current state x is mapped to a chain with one time step (whose value is x) 'marked'.



We don't actually compute everything beforehand, but simulate new states (and save them for future re-use) when needed.

Mapping to a Discretizing Chain - Continued

We then attempt to "move" the marker along the chain to another state (whose value is x'), with acceptance probability $\min[1, \frac{\pi(x')/\pi^*(x')}{\pi(x)/\pi^*(x)}]$. We can do multiple such updates in this space before mapping back to the original space.



(Solid line segments are the updates that are actually simulated, while the dashed segments are not).

Approximation: Dimension Reduction

There are mainly two classes of approximation methods. One class of approximations is based on reducing the dimension of the data.

- Subset of data (SoD): π^* is the "posterior" given only a subset (of m observations) of $(x_1, y_1), ..., (x_n, y_n)$. Need time proportional to pm^2 to compute C^*, and m^3 to invert C^*.
- Linear combination of responses: Let $\tilde{y} = Ay$ where A is of rank m. $\tilde{y}$ is also Gaussian, with lower dimension. π^* is the posterior based on the covariance matrix for $\tilde{y}$, $\tilde{C} = ACA^T$, of rank m.
- Others: SoR, Bayesian Committe Machine, etc...

Approximation: Diagonal Plus Low Rank

The other class is based on approximating the covariance matrix C by the sum of a diagonal matrix and a matrix of low rank.

C is usually of the form $\sigma^2 I + C^0$, where C^0 is non-negative definite. If C^0 can be approximated by some lower rank matrix $\hat{C}^0$, then with the matrix inversion lemma and the matrix determinant lemma:

$$(D + UWV^T)^{-1} = D^{-1} - D^{-1}U(W^{-1} + V^T D^{-1} U)^{-1} V^T D^{-1}$$

$$\det(D + UWV^T) = \det(W^{-1} + V^T D^{-1} U) \det(W) \det(D)$$

the computation can be reduced. Thus we can approximate the likelihood by substitute C with $\hat{C} = \hat{C}^0 + \sigma^2 I$ in the posterior:

$$(2\pi)^{-\frac{n}{2}} \det(\hat{C})^{-1} \exp\left(-\frac{1}{2} y^T \hat{C}^{-1} y\right)$$

Approximation: Diagonal Plus Low Rank - Continued

- Eigen-method: $\hat{C} = \sigma^2 I + B\Lambda_m B^T$, where Λ_m is the diagonal matrix with eigenvalues $\lambda_1 \geq \lambda_2, ..., \geq \lambda_m$ of C on its diagonal, and B is an $n \times m$ matrix whose columns are the corresponding orthonormal eigenvectors. Need to compute C (pn^2) and the first m eigenvalues and eigenvectors of C $(mn^2$, with a large constant factor).
- Nyström methods: $\hat{C} = \sigma^2 I + C^0_{(n,m)}[C^0_{(m,m)}]^{-1}C^0_{(m,n)}$ where $C^0_{(n,m)}$ is a $n \times m$ matrix, whose m columns are m randomly selected columns from C^0. Need to compute $C^0_{(n,m)}$ (pmn), then find the Cholesky decomposition of some $m \times m$ matrix, (m^3).

Example: Use SoD to form the π^*

We generate a synthetic dataset as follows:

$$y = 3\sin(x^2) + 2\sin(1.5x + 1) + \epsilon$$

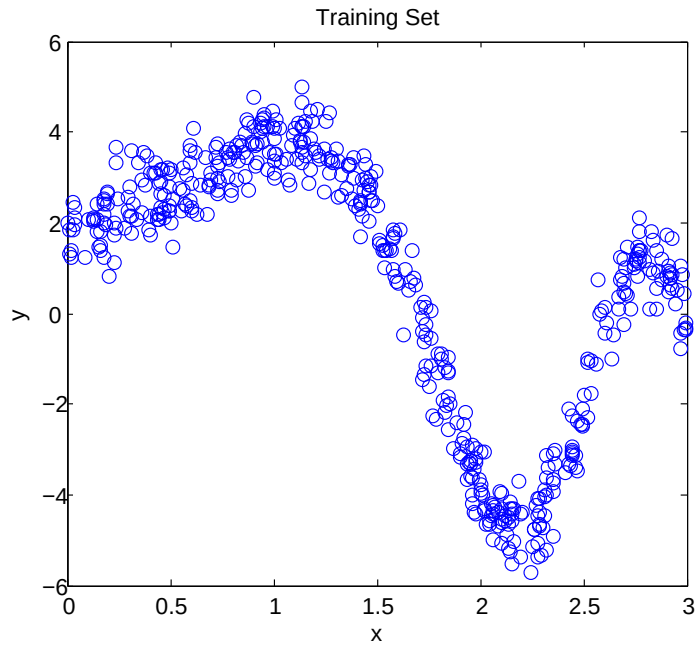
where $x \sim \text{Unif}(0, 3)$ and $\epsilon \sim N(0, 0.5^2)$. We generated 500 observations as the training set, and another 300 for the testing set.

We use the a squared exponential covariance function:

$$10^2 + \eta^2 \exp\left(-\frac{(x - x')^2}{\beta^2}\right) + \delta_{.,'}\sigma^2$$

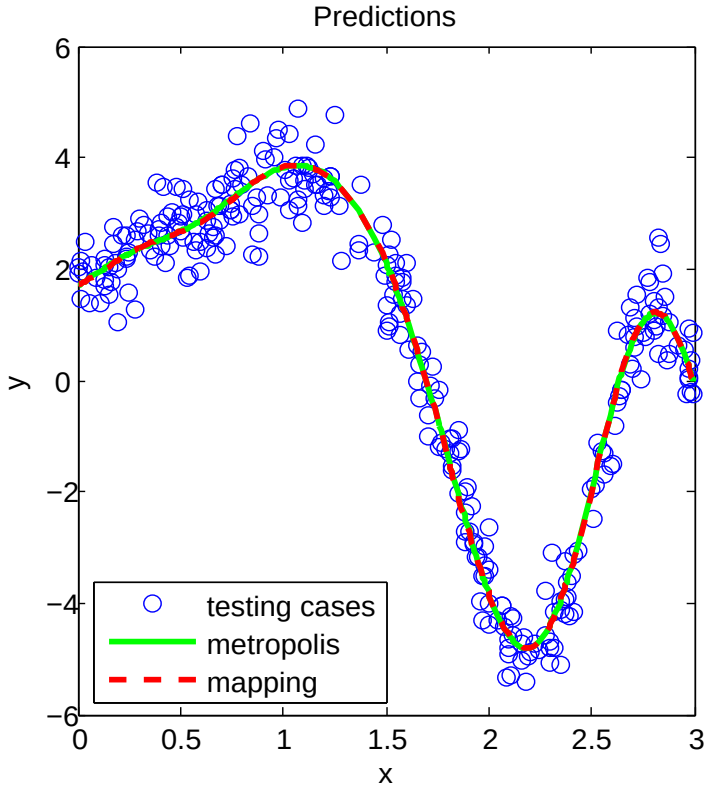
and the priors are

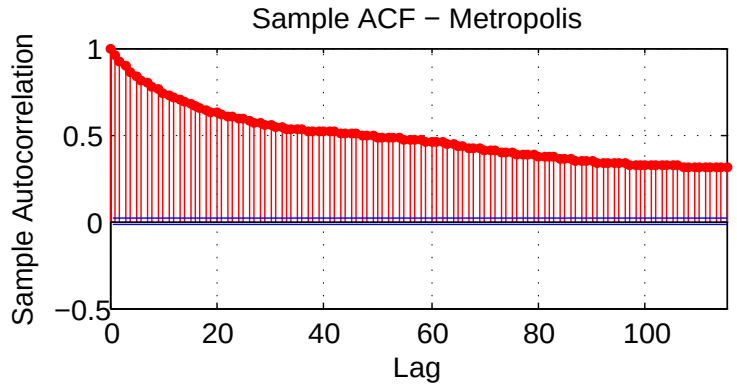
$$\log \eta^2 \sim N(3, 3^2)$$
$$\log \beta^2 \sim N(2, 3^2)$$
$$\log \sigma^2 \sim N(0, 3^2)$$

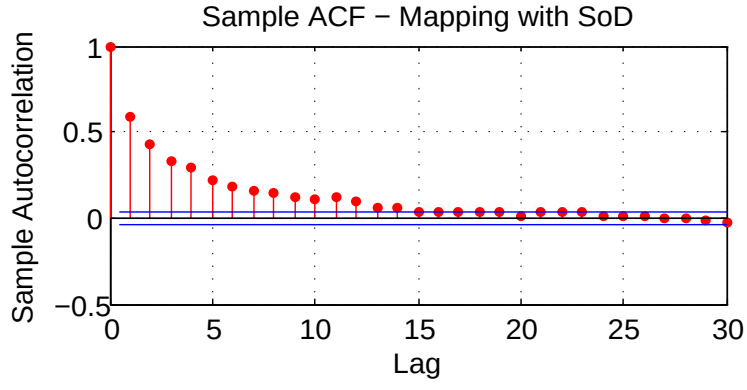


Example: Use SoD to form the π^* - Continued

The first 50 observations are used as the subset to form the π^* to implement the MCMC (with "mapping"), and compare the results to a Metropolis MCMC. The sample ACFs are adjusted so that they reflect the same amount of evaluations of $\pi(x)$.







References

1. Neal, R. M. (1998) *Constructing Efficient MCMC Methods Using Temporary Mapping and Caching*, Talk at Columbia University, December 2006
2. Neal, R. M. (1998) *Regression and Classification Using Gaussian Process Priors* Bayesian Statistics 6, pp. 475-501 Oxford University Press
3. Neal, R. M. (2008) *Approximate Gaussian Process Regression Using Matrix Approximations and Linear Response Combinations* Tech. Report (Draft), Dept. of Statistics, University of Toronto
4. Quiñonero-Candela, J., Rasmussen, C.E. and Williams, C. K. I. (2007) *Approximation Methods for Gaussian Process Regression* Tech. Report MSR-TR-2007-124, Microsoft Research
5. Rasmussen, C. E. and Williams, C. K. I. (2006) *Gaussian Processes for Machine Learning*, The MIT Press.