vignette

2024-12-08

R Markdown

Hidden Markov Model [**?**]is widely used for modeling series and it's logically clear for Bayesian inference, for it's forward generating process is explicit. A series of latent status is derived from a hidden Markov process and the data we see comes from a distribution parameterized by the latent status and other parameters invariant to state. The psoterior distribution given a series would be

$$p(\mathbf{X}, \mathbf{Z}|\theta) = p(z_1|\pi) \left[\prod_{n=2}^{N} p(z_n|z_{n-1}, \mathbf{A}) \right] \prod_{m=1}^{N} p(x_m|z_m, \phi)$$

Where $\mathbf{X}$ is the vector of the sequential data, $\mathbf{Z}$ is the vector of the sequential latent status. $\mathbf{A}$ and ϕ are parameters governing the whole model, π is the marginal distribution for starting states. θ is the collection of model parameters.

Given a proper prior of the model parameters, we can derive a posterior easily and we can do Bayesian inference by sampling methods. One potential choice is Hamiltonian Monte Carlo which has been well optimized in package **stan**. A **R** interface **Rstan**[**?**] is already developed, which makes it easier and faster than rewriting the Monte Carlo algorithms manually.

Model construction First we need a class to represent models. Given the model parameters θ, it should contain some basic functionalities and variables of a model instance. synthetic data generator Given a model instance, the synthetic data generator would first use the Markov Chain to generate a series of latent status and then sample from the specified distribution to get the data $\mathbf{X}$. The length of the sequence, the transition matrix and the parameters for the sampling distribution should be given. Also the starting state should come from the given marginal distribution π prior To do Bayesian inference we need priors for all the model parameters mentioned above. It takes the values of model parameters and then gives a evaluation of density value at the point. log posterior evaluation To do Bayesian inference we need a posterior distribution evaluator at any given values of the model parameters. The parameters are the input and the density value is the output. HMC sampler With **Rstan** interface and code written in Stan in a separated file, we can do the sampling process automatically. Based on the samples, we can give the maximum a posteriori estimation of the model parameters and also the uncertainty quantification.