MCMC Using Ensembles of States with Application to Gaussian Process Regression

Radford M. Neal, University of Toronto
Dept. of Statistics and Dept. of Computer Science

`http://www.cs.utoronto.ca/~radford/`

Outline of This Talk

- Very brief introduction to Markov chain sampling.

- The idea of temporarily mapping to another space.
  – Can be used in many ways. I'll look at just one...

- Ensemble MCMC.
  – map from $\mathcal{X}$ to $\mathcal{X}^K$, and back.
  – looking at K points does some averaging (integration) explicitly.
  – Need computational short-cut for computing at K points...

- Ensemble MCMC for problems with "fast" and "slow" variables.
  – can quickly recompute the density if only "fast" variables change.

- Application to Gaussian process regression.
  – overall scale of covariance matrix and noise variance can be "fast".

Markov Chain Sampling

We wish to sample from some distribution for $x \in \mathcal{X}$ that has probability/density function $\pi(x)$. But obtaining points drawn independently from π is too hard.

Instead we sample using a Markov chain with transition probabilities/densities (from x to x') denoted by $T(x'|x)$, for which π is an invariant distribution:

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

If the chain is ergodic — ie, it will eventually visit all parts of the space from any starting point — it will converge to this (unique) invariant distribution.

We can obtain a sample of dependent points drawn approximately from π by simulating one or more chains for a suitably-long time, and then discarding the early parts of the chains. We can use this sample to estimate expectations or quantiles of functions of x.

Note that we can combine several transitions that leave π invariant by applying them in turn, or choosing one randomly at each step.

The Metropolis Algorithm

In the *Metropolis algorithm* a Markov chain transition from state x to state x' operates as follows:

1) A "candidate", x^*, is proposed according to some probabilities $S(x^*|x)$, satisfying the symmetry condition that $S(x|x^*) = S(x^*|x)$.

2) This candidate, x^*, is accepted as the next state with probability

$$\min\left[1,\ \pi(x^*)/\pi(x)\right]$$

   If x^* is accepted, then $x' = x^*$. If x^* is instead rejected, then $x' = x$.

One can show that these transitions have π as an invariant distribution. Whether they are ergodic depends on the particular π and S.

Using a suitable proposal distribution, S, is crucial. Typical choices:

- Change all of x at once: $x^* \sim N(x, \sigma I)$.
- Change one element of a multidimensional x at a time: $x_i^* \sim N(x_i, \sigma^2)$.

The proposal width, σ, needs to be neither too big (almost all candidates rejected) nor too small (chain moves too slowly).

Transitions that Temporarily Map to Another Space

One way to define the transitions $T(x'|x)$ is via three other stochastic mappings, $\hat{T}$, $\bar{T}$, and $\check{T}$, as follows:

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

Starting from x, we obtain a value in the temporary space by sampling from $\hat{T}(y|x)$. The target distribution for $y \in \mathcal{Y}$ has probability/density function $\rho(y)$. We require that

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

$\bar{T}(y'|y)$ defines a transition on the temporary space that leaves ρ invariant:

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

Finally, $\check{T}$ gets us back to the original space. It must satisfy

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

The overall transition, $T(x'|x)$, leaves π invariant, and so can be used for Markov sampling of π.

Mapping to an Ensemble of Points

Let the temporary space be a collection of K points from $\mathcal{X}$ — ie, $\mathcal{Y} = \mathcal{X}^K$. We write a state in $\mathcal{Y}$ as $y = (x^{(1)}, \ldots, x^{(K)})$.

We specify an *ensemble base measure*, with density $\zeta(x^{(1)}, \ldots, x^{(K)})$. Define $\hat{T}$ as

$$\hat{T}(x^{(1)}, \ldots, x^{(K)} \,|\, x) \;=\; \frac{1}{K} \sum_{k=1}^{K} \zeta_{-k|k}(x^{(-k)} \,|\, x) \, \delta_x(x^{(k)})$$

where δ_x is the distribution concentrated at x, and $x^{(-k)}$ is all components of the ensemble other than $x^{(k)}$.

A simple example: $\mathcal{X} = [0, 2\pi)$, $K = 2$, and ζ is the uniform distribution on pairs of points with $x^{(2)} = x^{(1)} + \omega \pmod{2\pi}$, for some constant $\omega \in [0, 2\pi)$.

Then $\zeta_{-1|1}(\,\cdot\,|x^{(1)}) = \delta_{x^{(1)}+\omega \pmod{2\pi}}(\,\cdot\,)$ and $\zeta_{-2|2}(\,\cdot\,|x^{(2)}) = \delta_{x^{(1)}-\omega \pmod{2\pi}}(\,\cdot\,)$.

Algorithm for the map, $\hat{T}$, from x to $y = (x^{(1)}, x^{(2)})$:

– pick k uniformly from $\{1, 2\}$
– set $x^{(k)} = x$
– if $k = 1$, set $x^{(2)} = x + \omega \pmod{2\pi}$; if $k = 2$, set $x^{(1)} = x - \omega \pmod{2\pi}$.

The Ensemble Distribution

Having defined $\hat{T}$, we can find the *ensemble density*, $\rho(x^{(1)}, \ldots, x^{(K)})$, that results from applying $\hat{T}$ to a point drawn from π. Letting ζ_k be the marginal distribution for $x^{(k)}$ under the ensemble base distribution, ζ, we get:

$$\begin{aligned}
\rho(x^{(1)}, \ldots, x^{(K)}) &= \int \pi(x)\, \hat{T}(x^{(1)}, \ldots, x^{(K)} \,|\, x)\, dx \\
&= \frac{1}{K} \sum_{k=1}^{K} \zeta_{-k|k}(x^{(-k)} \,|\, x^{(k)})\, \pi(x^{(k)}) \\
&= \frac{1}{K} \sum_{k=1}^{K} \frac{\zeta(x^{(1)}, \ldots, x^{(K)})}{\zeta_k(x^{(k)})}\, \pi(x^{(k)}) \\
&= \zeta(x^{(1)}, \ldots, x^{(K)})\, \frac{1}{K} \sum_{k=1}^{K} \frac{\pi(x^{(k)})}{\zeta_k(x^{(k)})}
\end{aligned}$$

The simple example: ζ_1 and ζ_2 are both uniform over $[0, 2\pi)$, and ζ is uniform over pairs in $\mathcal{X}^2$ with $x^{(2)} = x^{(1)} + \omega$. So for pairs satisfying this constraint, the ensemble density follows this proportionality:

$$\rho(x^{(1)}, x^{(2)}) \;\propto\; \pi(x^{(1)}) + \pi(x^{(2)})$$

Mapping Back to a Single Point

To return to a single state in $\mathcal{X}$ from an ensemble in $\mathcal{X}^K$, we randomly select one of $x^{(1)}, \ldots, x^{(K)}$ with probabilities proportional to $\pi(x^{(k)})/\zeta_k(x^{(k)})$.

This $\check{T}$ mapping produces a point from π if the ensemble was from ρ:

$$\begin{aligned}
&\int \rho(y')\, \check{T}(x'|y')\, dy' \\
&\quad = \int \left[\zeta(x^{(1)}, \ldots, x^{(K)})\, \frac{1}{K} \sum_{k=1}^{K} \frac{\pi(x^{(k)})}{\zeta_k(x^{(k)})} \right] \left[\sum_{j=1}^{K} \delta_{x^{(j)}}(x')\, \frac{\pi(x^{(j)})}{\zeta_j(x^{(j)})} \Big/ \sum_{k=1}^{K} \frac{\pi(x^{(k)})}{\zeta_k(x^{(k)})} \right] \\
&\quad = \frac{1}{K} \sum_{j=1}^{K} \int \delta_{x^{(j)}}(x')\, \pi(x^{(j)})\, \frac{\zeta(x^{(1)}, \ldots, x^{(K)})}{\zeta_j(x^{(j)})}\, dx^{(1)} \cdots dx^{(K)} \\
&\quad = \frac{1}{K} \sum_{j=1}^{K} \int \pi(x')\, \zeta_{-j|j}(x^{(-j)}|x')\, dx^{(-j)} \\
&\quad = \frac{1}{K} \sum_{j=1}^{K} \pi(x') \quad = \quad \pi(x')
\end{aligned}$$

The simple example: $\check{T}$ picks $x^{(1)}$ or $x^{(2)}$ with probabilities proportional to π.

Doing Updates on Points and on the Ensemble

Since we can go back and forth between a single point in $\mathcal{X}$ and an ensemble of K points in $\mathcal{X}^K$, we can do Markov chain updates in either or both of these spaces.

The general scheme:

From some initial state in $\mathcal{X}$, repeat the following as many times as needed:

1) Do some updates (eg, Metropolis, using π) on $\mathcal{X}$.
2) Map from $\mathcal{X}$ to $\mathcal{X}^K$.
3) Do some updates (eg, Metropolis, using ρ) on $\mathcal{X}^K$.
4) Map from $\mathcal{X}^K$ to $\mathcal{X}$.

Note that we could omit updates on $\mathcal{X}$ in step (1), or updates on $\mathcal{X}^K$ in step (3), or even both (just mapping from $\mathcal{X}$ to $\mathcal{X}^K$ and back to $\mathcal{X}$ may change the state).

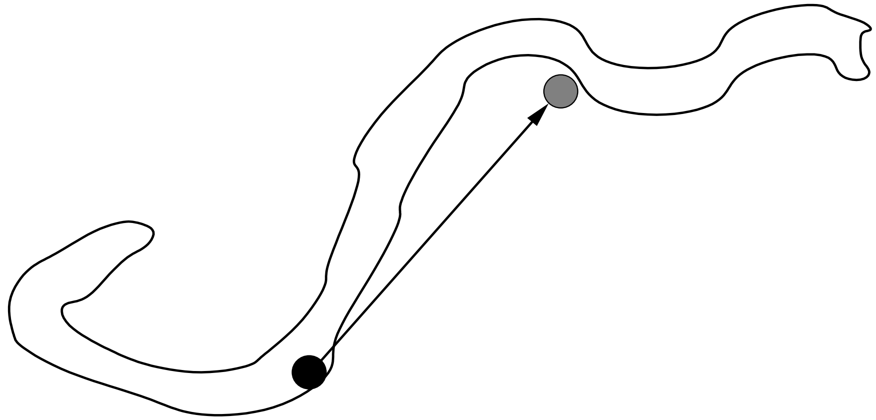
To make use of the Markov chain for estimation, we look at only the points in $\mathcal{X}$. (Though it would be possible to map states in $\mathcal{X}^K$ to $\mathcal{X}$ in a post-processing step.)

When are Updates on the Ensemble Beneficial?

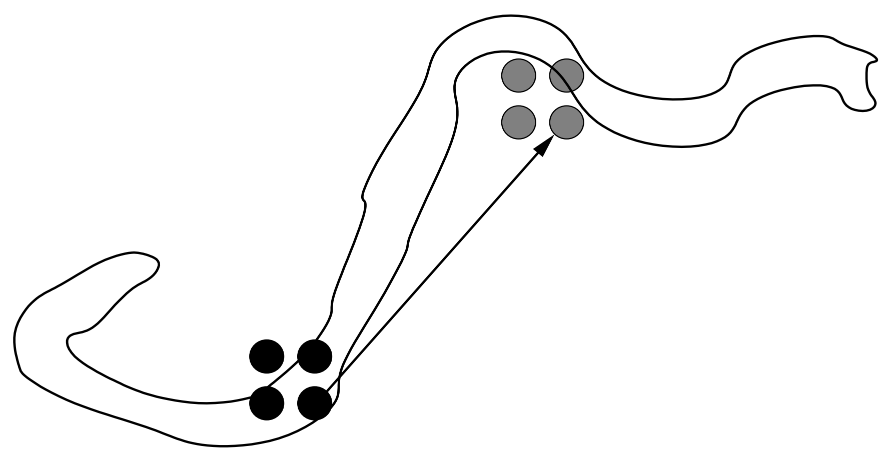
The ensemble density ρ is somewhat like an average of π over K points in $\mathcal{X}$ (sometimes exactly that, as in the simple example). Such averaging may smoothing out variation in π, making updates with bigger changes possible.

Suppose that π is uniform on some region of the plane, and an ensemble is four points arranged in a square, with ρ proportional to the sum of π at these points:

Compare:
– A Metropolis update for X proposing $X + \Delta$, with Δ a random displacement.
– A Metropolis update for $X^{(1)}, \ldots, X^{(4)}$ proposing $X^{(1)} + \Delta, \ldots, X^{(4)} + \Delta$.



proposal rejected



proposal accepted with probability 1/3

Some Possible Ensembles

Several choices of ensembles (ie, ensemble base measure, ζ) may provide useful averaging over the parameter space:

IID:

For some marginal $\zeta(x)$, let $\zeta(x^{(1)}, \ldots, x^{(K)}) \;=\; \prod_{k=1}^{K} \zeta(x^{(k)})$.

Exchangeable:

For some "prior" $\zeta(\theta)$ and conditional distribution $\zeta(x|\theta)$, let

$$\zeta(x^{(1)}, \ldots, x^{(K)}) \;=\; \int \zeta(\theta) \prod_{k=1}^{K} \zeta(x^{(k)}|\theta)\, d\theta$$

Constellation:

For some fixed $x_*^{(1)}, \ldots, x_*^{(K)}$ and arbitrary $\zeta(\theta)$, let

$$\zeta(x^{(1)}, \ldots, x^{(K)}) \;=\; \int \zeta(\theta) \prod_{k=1}^{K} \delta_{x_*^{(k)}+\theta}\,(x^{(k)})\, d\theta$$

We can obtain improper ensemble base measures by letting $\zeta(\theta)$ become improper.

The Cost of Ensemble Density Computations

Recall the formula for the ensemble density:

$$\rho(x^{(1)}, \ldots, x^{(K)}) \;=\; \zeta(x^{(1)}, \ldots, x^{(K)})\, \frac{1}{K} \sum_{k=1}^{K} \frac{\pi(x^{(k)})}{\zeta_k(x^{(k)})}$$

We need to compute this to do Metropolis updates on the ensemble (as well as for most other MCMC updates).

Computing ρ in the obvious way takes at least K times as long as computing π. Assuming that computing π is a major cost, this would usually cancel any advantage from using an ensemble.

Conclusion: Ensemble MCMC is likely to be useful only when π and the choice of ensemble (ie, ζ) allow calculation of ρ more quickly than K evaluations of π.

One class of tricks for this involves "caching" the results of computations for later reuse. Here I look at one context where this is useful...

Distributions with “Fast” and “Slow” Variables

Consider a problem with $x = (x_1, x_2)$, where $x_1 \in \mathcal{X}_1$ consists of one or more “slow” variables, and $x_2 \in \mathcal{X}_2$ consists of one or more “fast” variables.

When we compute $\pi(x_1, x_2)$, we can save some intermediate results, and later compute $\pi(x_1, x_2')$ for any x_2' in much less time than would be needed to compute $\pi(x_1', x_2')$ for a new x_1'.

Example: Bayesian models of the cosmic microwave background radiation (see Lewis and Bridle 2002) involve some parameters, x_1, of a complex simulation model of the early universe, together with other parameters, x_2, that relate the output of the simulation to observations.

Finding the posterior density of $x = (x_1, x_2)$ for a new value of x_1 requires running a new simulation. Finding the posterior density for a new value of x_2 but an old value of x_1 is much faster, provided the simulation results for x_1 were saved.

A simple method: After each Metropolis update for all variables, do many Metropolis updates changing only fast variables.

Ensembles with Slow Variables Fixed, Fast Variables Varying

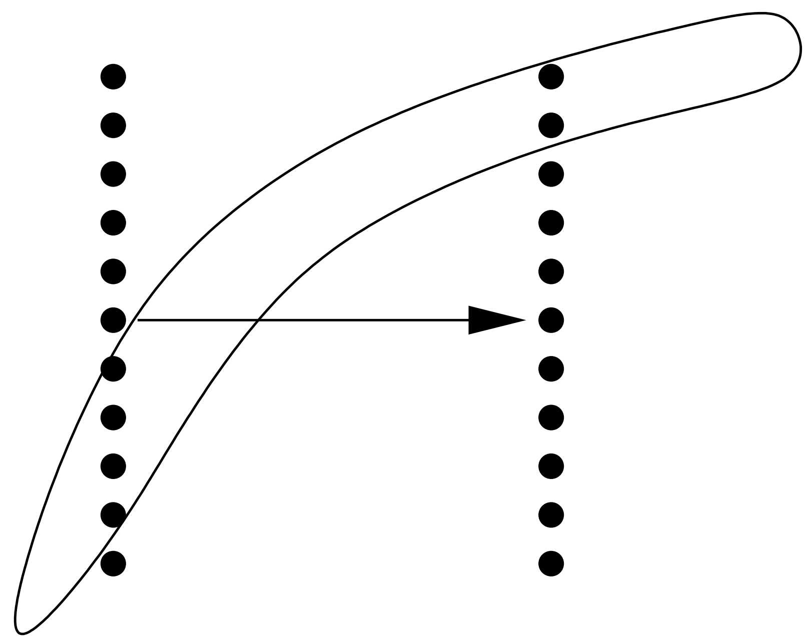
We can apply ensemble MCMC for these problems using an ensemble base measure in which slow variables are the *same* for all members of the ensemble.

Computing ρ then requires only *one* slow evaluation of π. If such slow evaluations are the dominant cost, evaluating ρ will be almost as fast as evaluating π.

For the fast variables, we can use analogues of any of the ensembles discussed earlier — IID, exchangeable, or constellation (eg, grid). As $K \to \infty$, these ensembles can approach the ideal of integrating away the fast variables.

Illustration with one slow variable (horizontal) and one fast variable (vertical). The distribution is uniform over the outlined region.

A grid ensemble is used. The proposed move of the ensemble will be accepted with probability 2/4.



Gaussian Process Regression

We observe $(z^{(i)}, y^{(i)})$ for $i = 1, \ldots, n$, where $z^{(i)}$ is a vector of p covariates, and $y^{(i)}$ a real-valued response.

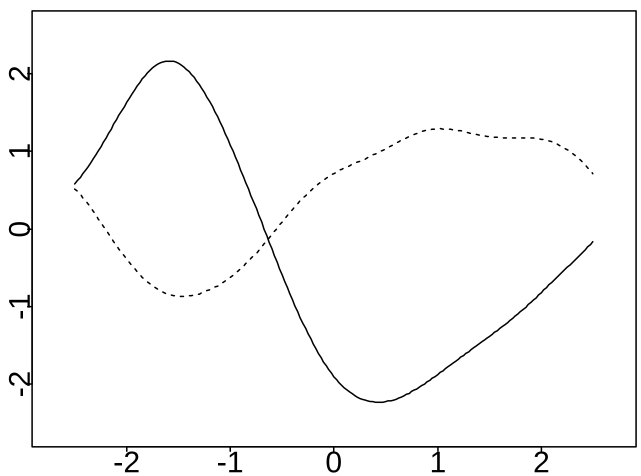
Our model is $y^{(i)} = f(z^{(i)}) + e^{(i)}$, where f is an unknown function, and $e^{(i)}$ is independent Gaussian noise with mean zero and variance σ^2.

We can give f a Gaussian process prior, with mean zero and some covariance function $C(z, z') = E[f(z)f(z')]$. The Gaussian noise can be absorbed into this Gaussian prior.

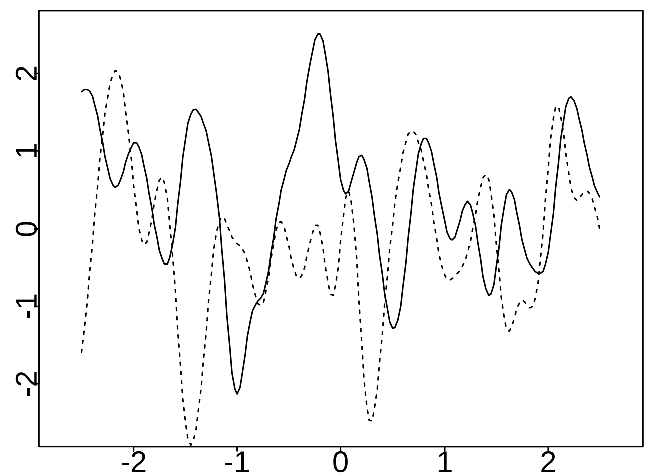
The $y^{(i)}$, together with any unobserved y^* at known z^*, have a multivariate Gaussian joint distribution. So we can predict y^* by its conditional distribution given the observed $y^{(1)}, \ldots, y^{(n)}$. This can be done with matrix operations in $O(n^3)$ time.

The choice of covariance function, $C(z, z')$, affects the properties of functions drawn from the prior...

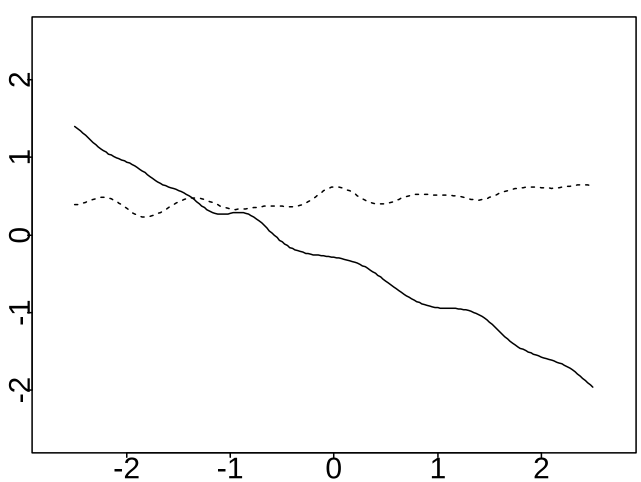
Functions From Some Gaussian Process Priors



$\exp\left(-(z-z')^2\right)$

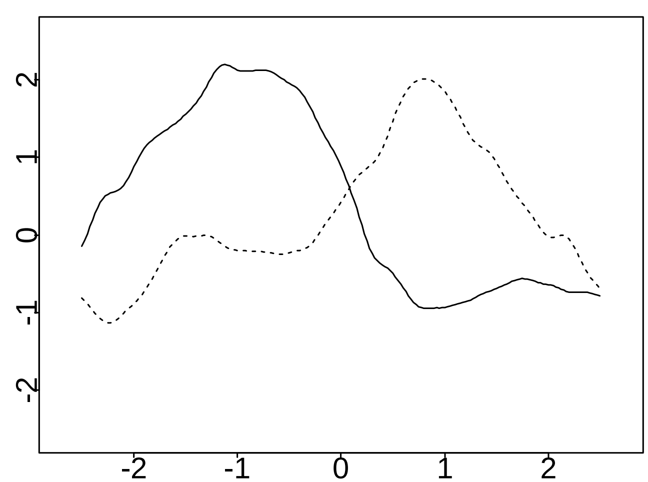


$\exp\left(-5^2(z-z')^2\right)$



$1 + zz'$

$+\ 0.1^2 \exp\left(-3^2(z-z')^2\right)$



$\exp\left(-(z-z')^2\right)$

$+\ 0.1^2 \exp\left(-5^2(z-z')^2\right)$

Inference for Parameters of a Gaussian Process

We would seldom know enough about the unknown regression function to fix a covariance function for its Gaussian process prior.

Instead, we parameterize the covariance function, give the parameters priors, and base inference/prediction on the posterior distribution of these parameters.

The log likelihood is just that of a multivariate Gaussian:

$$\log L \;=\; -(1/2)\log\det\Sigma \;-\; (1/2)y^T\Sigma^{-1}y$$

where $y = [y^{(1)}, \ldots, y^{(n)}]^T$ and Σ is the prior covariance matrix of y, with $\Sigma_{ij} = C(y^{(i)}, y^{(j)})$. Note that Σ depends on the unknown parameters of the covariance function.

Two ways to compute $\log L$:

- Find the Cholesky decomposition of Σ (lower-triangular L such that $LL^T = \Sigma$).
- Find the eigenvectors and eigenvalues of Σ.

Both take $O(n^3)$ time, but the Cholesky method is about 15 times faster.

Note that computing Σ takes $O(n^2)$ time, but with a larger constant factor. For moderate n, this may dominate the $O(n^3)$ operations, especially if p is big.

Fast Variables for Gaussian Process Inference

Here is the log likelihood again:

$$\log L \;=\; -(1/2) \log \det \Sigma \;-\; (1/2) y^T \Sigma^{-1} y$$

If $\log L$ has been computed for Σ, and $\det \Sigma$ and $y^T \Sigma^{-1} y$ were saved, then the likelihood for $s\Sigma$ for any scalar s is easily found:

$$\det(s\Sigma) \;=\; s^n \det(\Sigma), \qquad y^T (s\Sigma)^{-1} y \;=\; (1/s) y^T \Sigma^{-1} y$$

So an overall scale factor for covariances can be a fast variable, requiring $O(1)$ time for fast recomputation.

If we compute the $\log L$ using the eigenvectors $(e_1, \ldots, e_n)$ and eigenvalues $(\lambda_1, \ldots, \lambda_n)$ of Σ, and save these as well as the projections of y on the eigenvectors $(u_i = y^T e_i)$, we can quickly recompute $\log L$ for $s\Sigma + tI$:

$$\det(s\Sigma + tI) \;=\; \prod_{i=1}^{n} (s\lambda_i + t), \qquad y^T \Sigma^{-1} y \;=\; \sum_{i=1}^{n} u_i^2 / (s\lambda + t)$$

This allows both an overall scale parameter and the noise variance to be fast variables, with $O(n)$ time for fast recomputations.

A Test Problem

I generated $n = 100$ synthetic observations with $p = 12$ covariates. The response was a non-linear function of just the first three covariates:

$$f(z) \;=\; 0.7z_1^2 \;+\; 0.8\sin(0.3 + (4.5 + 0.5z_1)\,z_2) \;+\; 0.85\cos(0.1 + 5z_3 + 0.1z_2^2)$$

However, the other nine covariates were correlated with the first three, so they might appear relevant. The noise standard deviation was 0.4.

The details of this test problem were adjusted so that, for the Gaussian process model used, the posterior distribution given this data has multiple modes. These modes correspond to different sets of covariates being regarded as relevant, with corresponding variation in the inferred scale of the noise-free response and in the noise variance.

The Gaussian Process Model Used

For the test problem, I used the following covariance function:

$$\mathrm{Cov}(y^{(i)}, y^{(j)}) \;=\; \eta^2 \left(a^2 \;+\; \exp\left(-\sum_{h=1}^{p} \left(\nu_h \, (z_h^{(i)} - z_h^{(j)}) \right)^2 \right) + r^2 \delta_{ij} \right) + \sigma^2 \delta_{ij}$$

where δ_{ij} is zero if $i \neq j$ and one if $i = j$. I fixed $a = 1$ and $r = 0.01$.

In the prior used, η, σ, and $\nu = (\nu_1, \ldots, \nu_9)$ are independent, with

$$\begin{aligned} \log(\eta) &\sim N(\log(1),\, 1.5^2) \\ \log(\sigma) &\sim N(\log(0.5),\, 1.5^2) \\ \log(\nu) &\sim N(m,\, S) \end{aligned}$$

Here, the all components of the mean vector m are $\log(0.5)$, and in the covariance matrix S, all variances are 1.8^2, and all correlations are 0.69.

Metropolis MCMC for the Gaussian Process Model

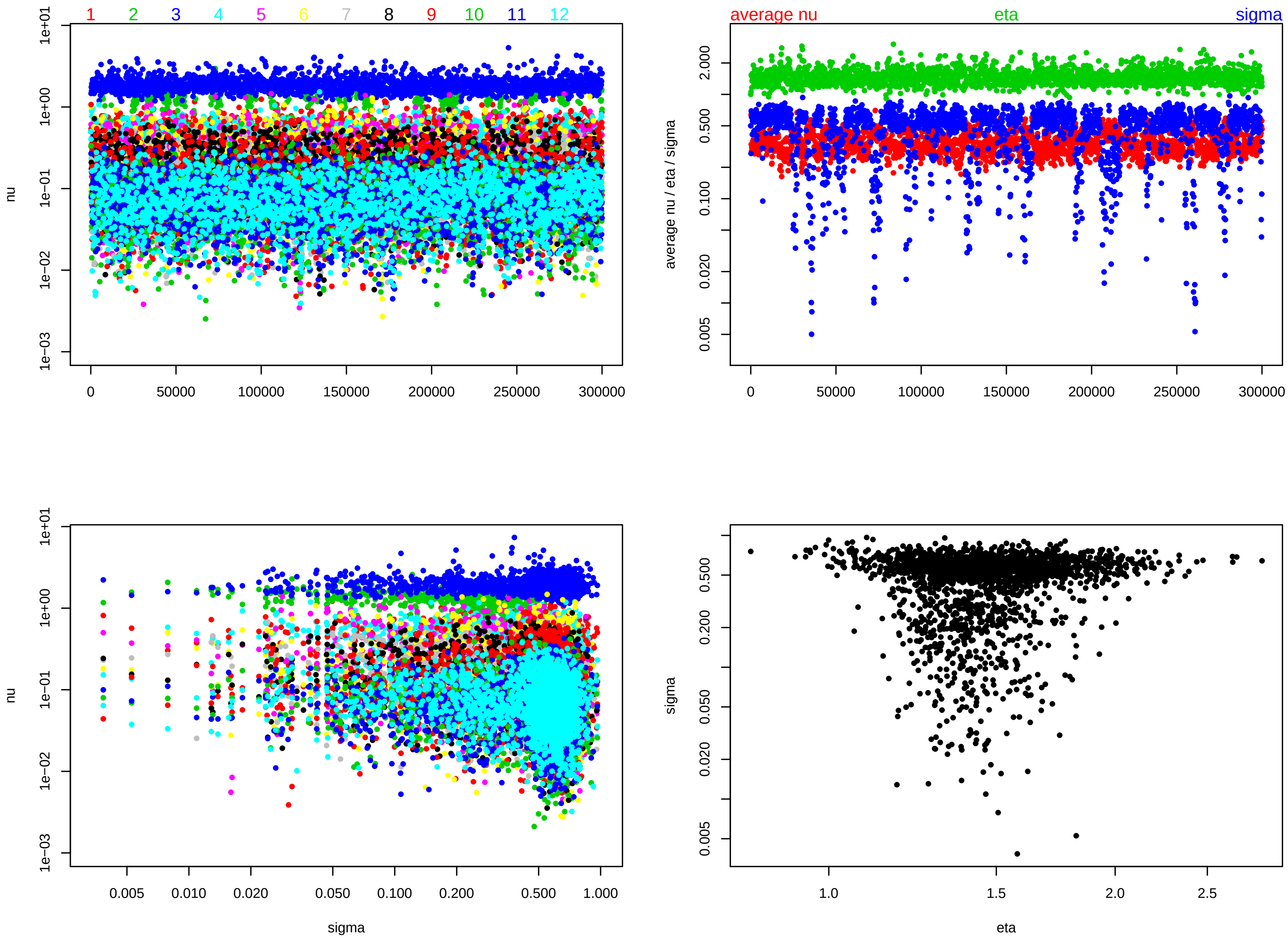
Two options:

- Update all parameters at once, proposing change from $N(0, sI)$.
- Update one parameter at a time, proposing change from $N(0, s_i^2)$.

We can also try adding extra updates of just the fast variables.

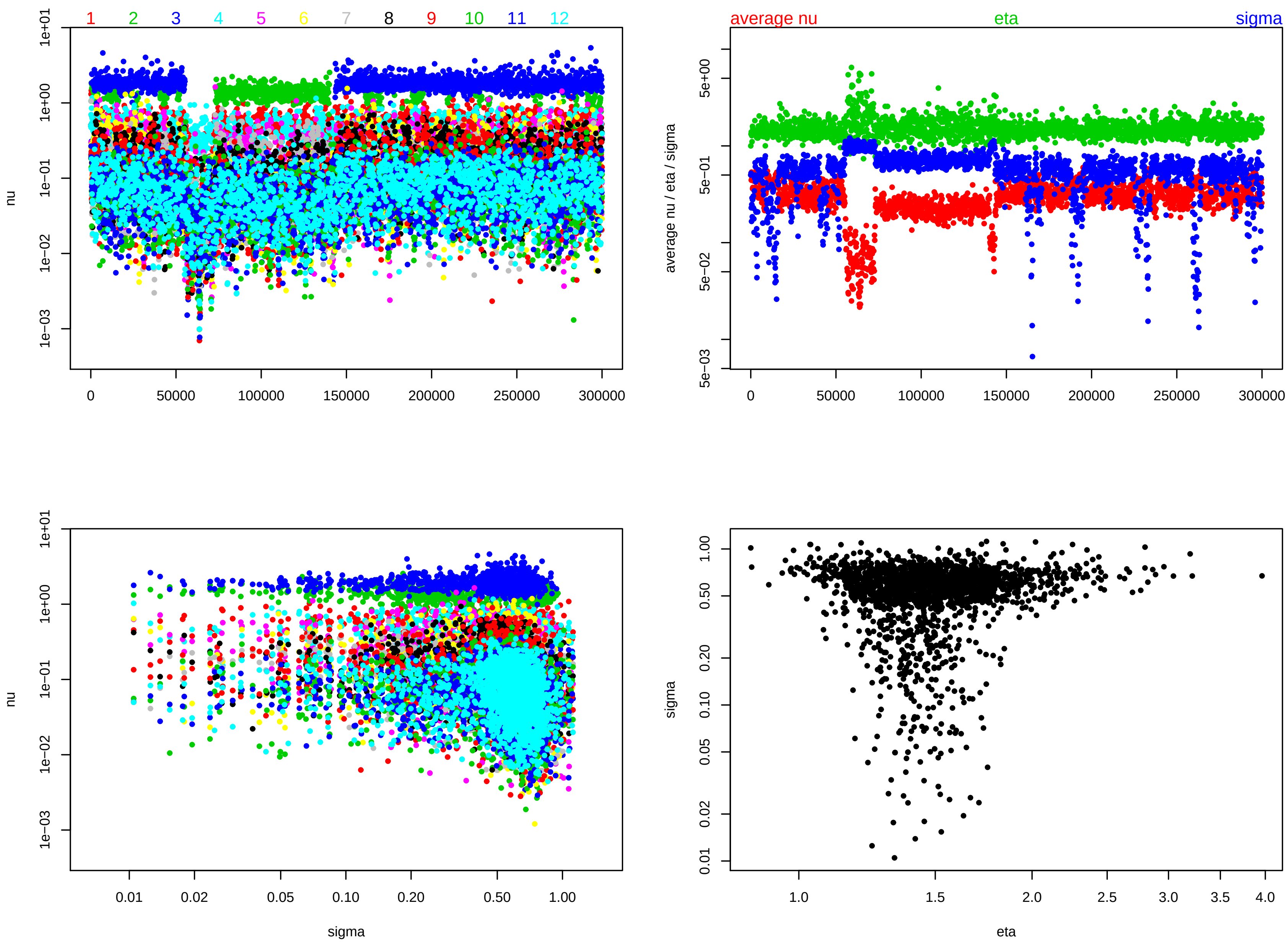
All runs did 300,000 slow evaluations, so they are comparable if this is the dominant cost.

Standard Metropolis — Multivariate Updates (0.25)



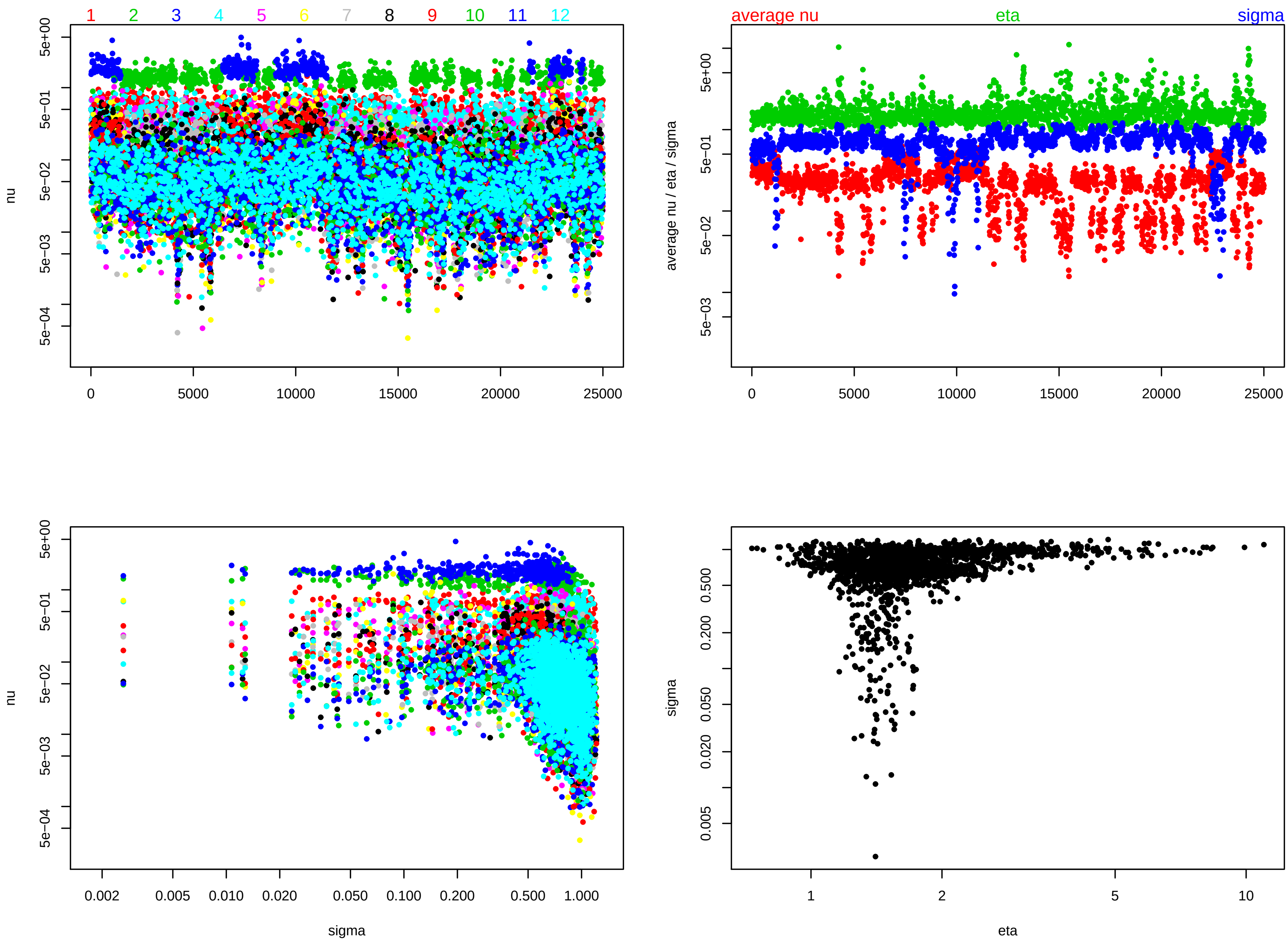
Proposal standard deviation of 0.25

Standard Metropolis — Multivariate Updates (0.35)



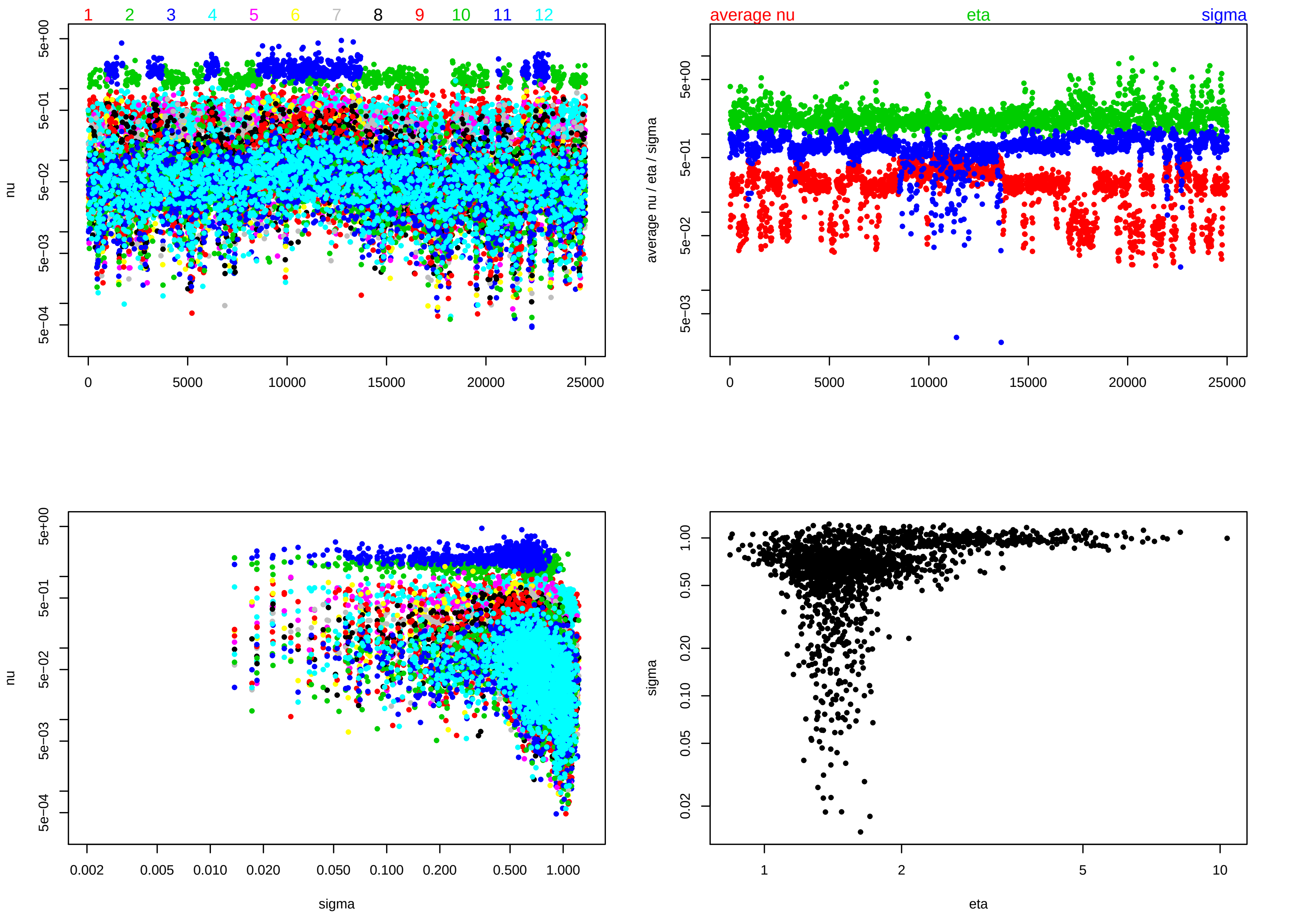
Proposal standard deviation of 0.35

Standard Metropolis — Single Variable Updates



Proposal standard deviation of 2.0 for $\log(\nu)$, 0.6 for $\log(\eta)$ and $\log(\sigma)$

Standard Metropolis — Single Variable Updates + Extra Fast



Proposal standard deviation of 2.0 for $\log(\nu)$, 0.6 for $\log(\eta)$ and $\log(\sigma)$, 49 extra for $\log(\eta)$ and $\log(\sigma)$

Ensemble MCMC for the Gaussian Process Model

There are two computational strategies:

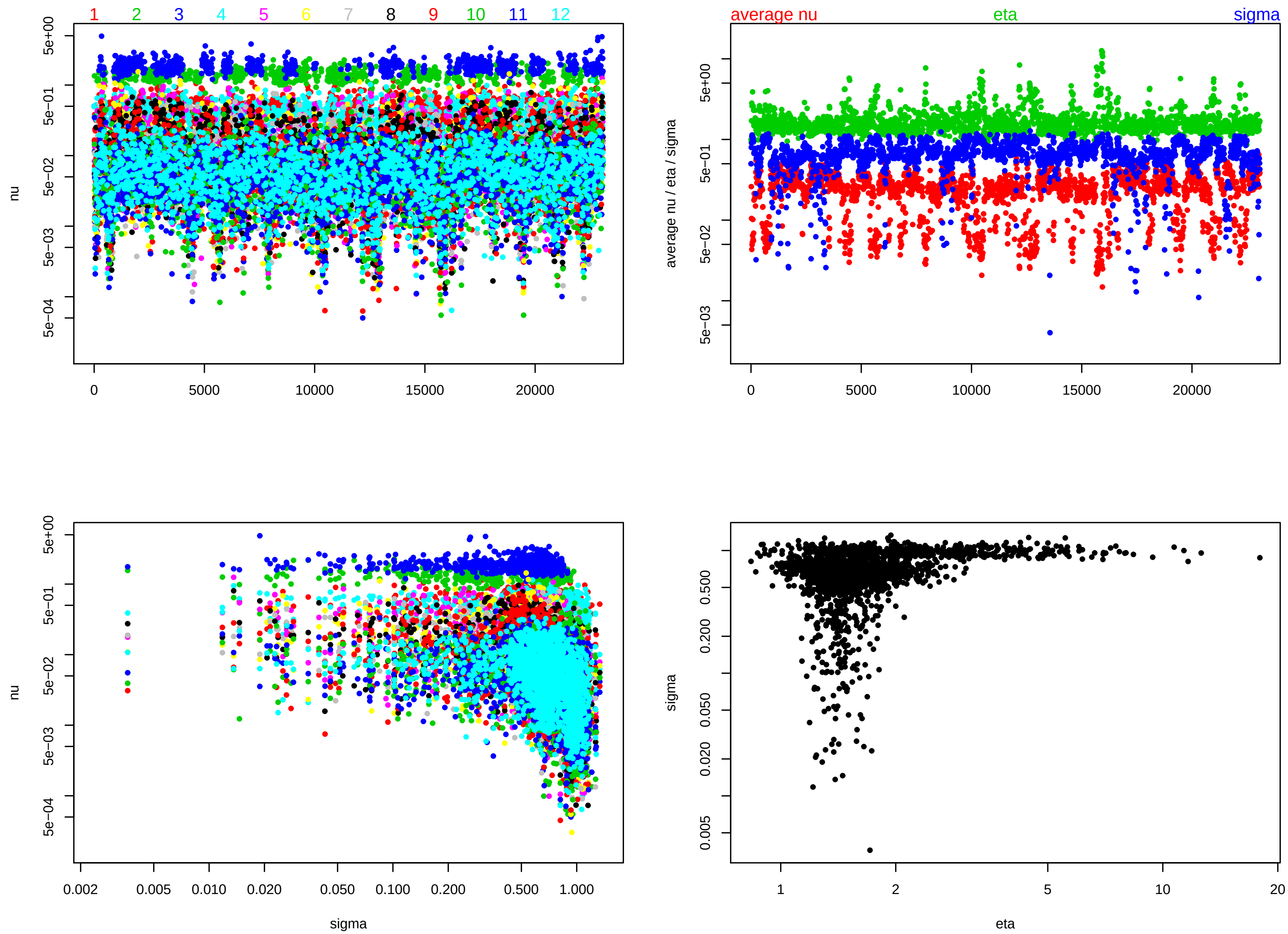
- Cholesky (only η fast); replace σ by $\psi = \sigma/\eta$.
- Eigenvectors (both η and σ fast).

For each, I tried three ensembles: IID, exchangeable, grid.

I tried ensemble sizes of 49 and 400.

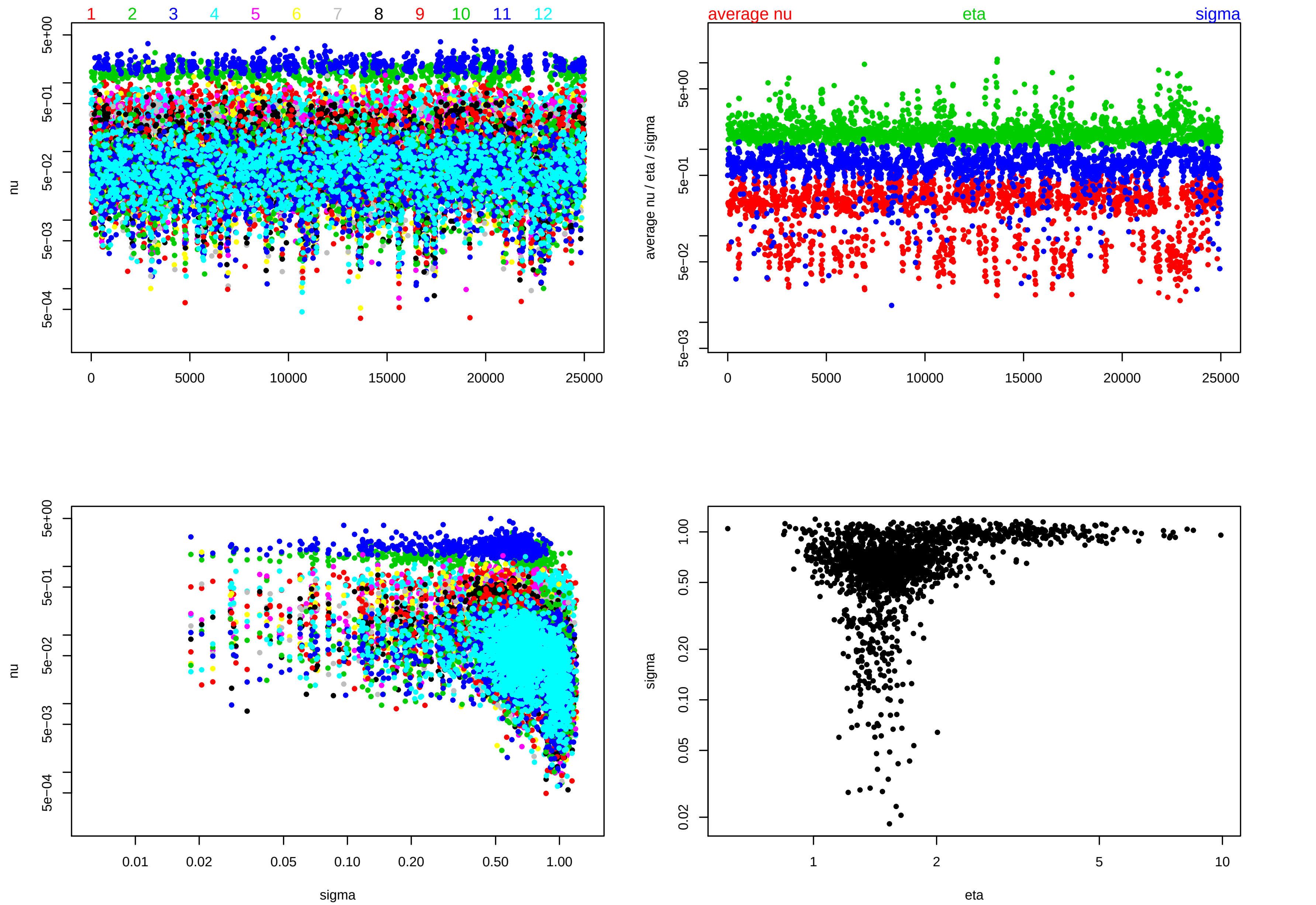
With suitable tuning, the three ensembles produced similar results. I'll show results for the exchangeable ensemble.

Exchangeable Ensemble (Chol, 49) — Single Variable Updates



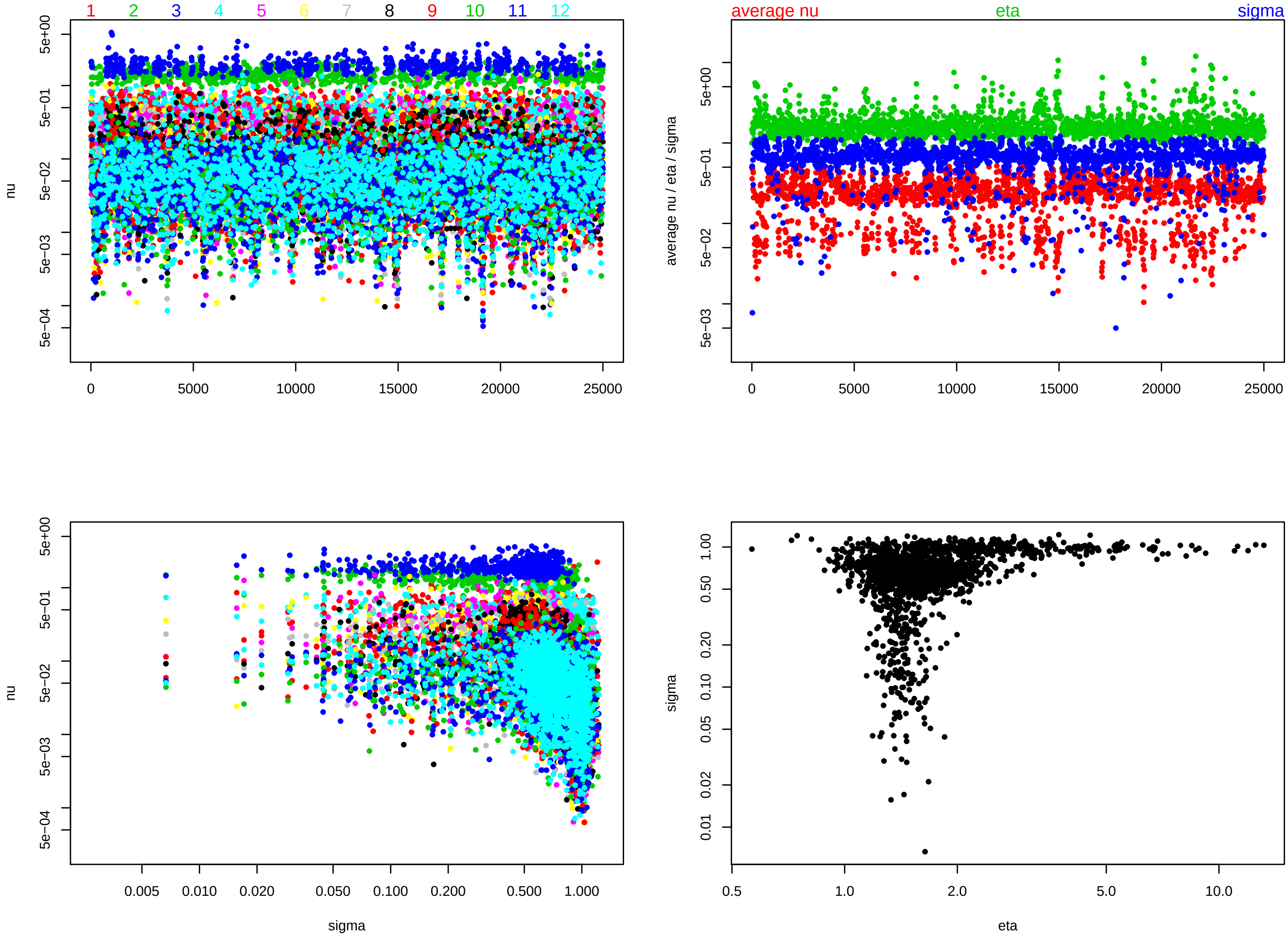
Proposal sd 2.0/0.6, ensemble sd 0.4 × prior sd

Exchangeable Ensemble (Eigen, 49) — Single Variable Updates



Proposal sd 2.0/0.6, ensemble sd 0.4 × prior sd

Exchangeable Ensemble (Eigen, 400) — Single Variable Updates



Proposal sd 2.0/0.6, ensemble sd 0.4 × prior sd

Comments

- We've seen that ensemble MCMC with fast/slow variables can be useful for Gaussian process regression. The usefulness will vary with n, p, the complexity of the covariance function, and the properties of the posterior distribution for the given data.

- Ensemble MCMC may also be useful for Gaussian process models with latent variables — equivalent computationally to "generalized linear mixed models" — with the latent variables being the fast variables.

- Ensemble MCMC using a "constellation" may also be useful when a change to *any* single variable allows fast recomputation.

- My earlier use of "embedded hidden Markov models" to sample posterior distributions for non-linear state space models can also be seen as an application of ensemble MCMC.

- Are there other applications?