INVESTIGATING MULTIMODAL DENSITY SAMPLING METHOD REPELLING-ATTRACTING METROPOLIS

STAT 654 - Term Project - Group 8

Professor - David E Jones

Abhishek Sinha
Abhishek Soma
Shubham Zope
Srujan Jabbireddy
Suryananaryana Nadumpalli

PROJECT OUTLINE

- Multimodal Posterior Density
- Metropolis – Hastings Algorithm and its Limitations
- Repelling-Attracting Metropolis (RAM)
- RAM on Sensor Network Localization With 6 Nodes
- Simulated Sensor Network With 9 Nodes
- Sensor Network- Performance Statistic
- Comparison of RAM with BFGS Optimization Method
- Exoplanet: Posterior Sampling of a Simulated Radial Velocity Dataset

Multimodal Posterior Density

What is multimodal posterior density/distribution.

- A continuous probability distribution with more than or equal to two peak/mode.
- The posterior distribution contains all the information about the possible parameter values. A multimodal density/distribution represents more than 1 mode i.e. more than 1 possible parameter value.

Various methods to find multimodal density

- Popular MCMC strategy for dealing with multimodality are
  - Tempering such as parallel tempering (Geyer, 1991),
  - Simulated tempering (Geyer and Thompson, 1995),
  - Tempered transitions (Neal, 1996), and
  - Equi-energy sampler (Kou et al., 2006).

Though these methods are powerful, they typically require extensive tuning.

Metropolis-Hastings

- Markov Chain – a stochastic process in which future states are independent of past states, given the current state.
- Monte Carlo method - an method that helps to obtain a desired value by performing simulations involving probabilistic choices.
- Metropolis Hastings
  - one of the most popular MCMC method for sampling from posterior distribution.
  - However, in case of multimodal posterior, MH algorithm generates Markov Chains that do not readily jump between modes.
  - Used when direct sampling is difficult

Metropolis-Hastings algorithm

Let $f(x)$ be the (possibly unnormalized) target density, $x^{(j)}$ be a current value, and $q(x|x^{(j)})$ be a proposal distribution, then

- Sample $x^* \sim q(x|x^{(j)})$.
- Calculate the acceptance probability

$$\rho(x^{(j)}, x^*) = \min\left\{1, \frac{f(x^*)}{f(x^{(j)})}\frac{q(x^{(j)}|x^*)}{q(x^*|x^{(j)})}\right\}.$$

- Set $x^{(j+1)} = x^*$ with probability $\rho(x^{(j)}, x^*)$, otherwise set $x^{(j+1)} = x^{(j)}$.

Metropolis Hasting Gibbs Sampling

Gibbs Sampling:

- A special case of Metropolis Hastings algorithms
- Applicable when the joint distribution is not known explicitly or is difficult to sample from directly, but the conditional distribution of each variable is known
- Here we break the problem of sampling from the high-dimensional joint distribution into a series of samples from low-dimensional conditional distributions.
  - we generate posterior samples by sweeping through each variable to sample from the conditional distribution with the remaining variables set to their current values.
  - Sampling depends on whether we can derive the conditional posterior distributions.

Initialization: Initialize $\mathbf{x}^{(0)} \in \mathcal{R}^D$ and number of samples N

- **for** $i = 0$ to $N-1$ **do**
- $x_1^{(i+1)} \sim p(x_1 | x_2^{(i)}, x_3^{(i)}, ..., x_D^{(i)})$
- $x_2^{(i+1)} \sim p(x_2 | x_1^{(i+1)}, x_3^{(i)}, ..., x_D^{(i)})$
- $\vdots$
- $x_j^{(i+1)} \sim p(x_j | x_1^{(i+1)}, x_2^{(i+1)}, ..., x_{j-1}^{(i+1)}, x_{j+1}^{(i)}, ..., x_D^{(i)})$
- $\vdots$
- $x_D^{(i+1)} \sim p(x_D | x_1^{(i+1)}, x_2^{(i+1)}, ..., x_{D-1}^{(i+1)})$

return $(\{\mathbf{x}^{(i)}\}_{i=0}^{N-1})$

Repelling-Attracting Metropolis

TEXAS A&M UNIVERSITY

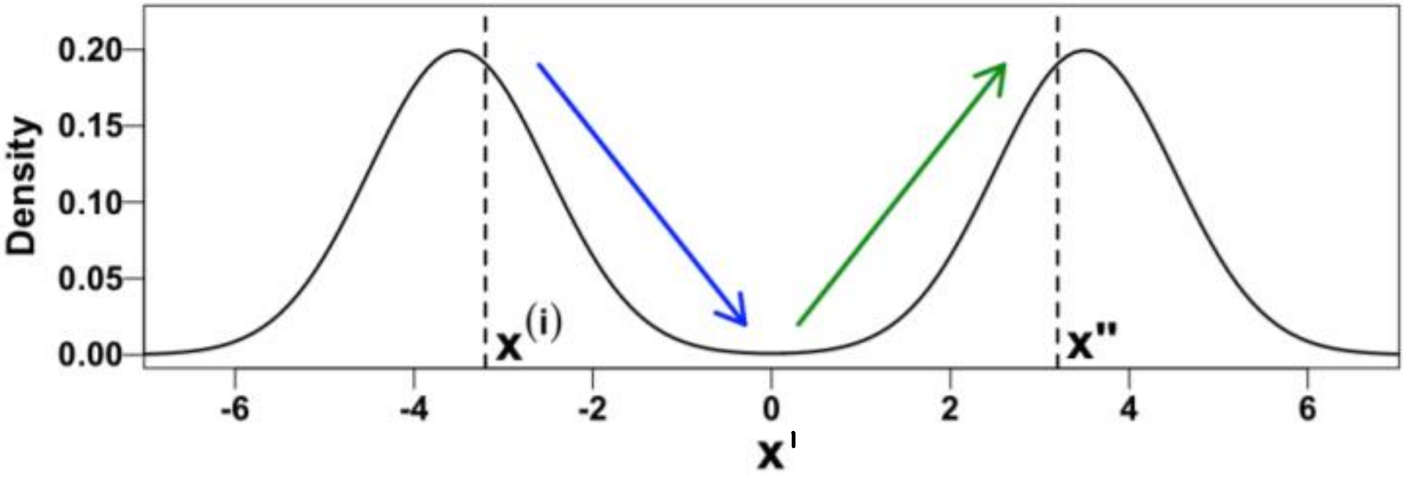
Repelling-Attracting Metropolis (RAM) is an MH algorithm with a unique joint jumping density and an easy-to-compute acceptance probability that preserves the target marginal distribution.

- This method was developed by **Hyungsuk Tak, Xiao-Li Meng, David A. van Dyk**.
- This algorithm improve metropolis ability to jump the modes more often than Metropolis, and with less tuning requirements than tempering methods.

RAM generates a proposal via **forced downhill and forced uphill Metropolis transitions.**

- The forced downhill Metropolis transition uses a **reciprocal ratio of the target densities in its acceptance probability** which encourages it to prefer downward moves.
- The subsequent forced uphill Metropolis transition generates a final proposal with a standard Metropolis ratio that **makes local modes attracting**.
- Together, the downhill and uphill transitions form a proposal for a Metropolis-Hastings sampler.

RAM PROPOSAL



$x^{(i)}$: Current state

***x′* : Intermediate proposal**

***x″* : Final proposal**

Two step procedure:

- ***Downhill Metropolis***: Generate $x' \sim N(x^{(i)}, \sigma^2)$ and
  - accept x′ with probability $\alpha_\epsilon^D(x'|x(i)) = min\ \{1, \frac{\pi(x^{(i)})+\epsilon}{\pi(x')+\epsilon}\}$.
- ***Uphill Metropolis***: Generate $x'' \sim N(x', \sigma^2)$ and
  - accept x″ with probability $\alpha_\epsilon^U(x''|x(i)) = min\ \{1, \frac{\pi(x'')+\epsilon}{\pi(x')+\epsilon}\}$.

Above steps are repeated until the proposal is accepted (forced metropolis)

Acceptance/Rejection Probability

Accept x″ with a Metropolis-Hastings acceptance probability

$$\alpha^{DU}(x''|x^{(i)}) = \min\{ 1, \frac{\pi(x'')q^{DU}\left(x^{(i)} \mid x''\right)}{\pi(x^{(i)})q^{DU}(x''|x^{(i)})} \}$$

$$\alpha^{DU}(x''|x^{(i)}) = \min\{ 1, \frac{\pi(x'') \int N(x|x^{(i)},\sigma^2)\alpha_\epsilon^D(x\,|x^{(i)})dx}{\pi(x^{(i)}) \int N(x|x'',\sigma^2)\alpha_\epsilon^D(x\,|x'')dx} \}$$

- But calculating the integral part of acceptance probability is intractable.
- So we introduce an auxiliary variable and produce a joint distribution.
- We introduce a joint target density π(x, z) = π(x)q(z | x)

The joint acceptance probability draws down to:

$$\alpha^J(z'',x''|z^{(i)},x^{(i)}) = \min\{ 1, \frac{\pi^J(z'',x'')q^J\left(z^{(i)},x^{(i)}\,|z'',x''\right)}{\pi^J(z^{(i)},x^{(i)})q^J(z'',x''\,|z^{(i)},x^{ii})} \}$$

$$= \min\{ 1, \frac{\pi(x'')\min\{1,\frac{\pi\left(x^{(i)}\right)+\epsilon}{\pi\left(z^{(i)}\right)+\epsilon}\}}{\pi\left(x^{(i)}\right)\min\{1,\frac{\pi(x'')+\epsilon}{\pi(z'')+\epsilon}\}} \}$$

RAM Algorithm

RAM is composed of 4 steps in each iteration.

a) Steps 1-3 generates a joint proposal (z'', x'')
b) Step 4 accepts or rejects the joint proposal (z'', x'')
c) ε is introduced to avoid 0/0 values and its value is chosen to be 10e-308 (minimum positive number in R)

Table 1: A repelling-attracting Metropolis algorithm.

Set initial values $x^{(0)}$ and $z^{(0)}$ $(= x^{(0)})$. For $i = 0, 1, \ldots$

Step 1: ($\searrow$) Repeatedly sample $x' \sim q(x' \mid x^{(i)})$ and $u_1 \sim \text{Uniform}(0,1)$ until $u_1 < \min\left\{1, \frac{\pi(x^{(i)})+\epsilon}{\pi(x')+\epsilon}\right\}$.

Step 2: ($\nearrow$) Repeatedly sample $x^* \sim q(x^* \mid x')$ and $u_2 \sim \text{Uniform}(0,1)$ until $u_2 < \min\left\{1, \frac{\pi(x^*)+\epsilon}{\pi(x')+\epsilon}\right\}$.

Step 3: ($\searrow$) Repeatedly sample $z^* \sim q(z^* \mid x^*)$ and $u_3 \sim \text{Uniform}(0,1)$ until $u_3 < \min\left\{1, \frac{\pi(x^*)+\epsilon}{\pi(z^*)+\epsilon}\right\}$.

Step 4: Set $(x^{(i+1)}, z^{(i+1)}) = (x^*, z^*)$ if $u_4 < \min\left\{1, \frac{\pi(x^*)\min\{1, (\pi(x^{(i)})+\epsilon)/(\pi(z^{(i)})+\epsilon)\}}{\pi(x^{(i)})\min\{1, (\pi(x^*)+\epsilon)/(\pi(z^*)+\epsilon)\}}\right\}$, where $u_4 \sim \text{Uniform}(0,1)$, and set $(x^{(i+1)}, z^{(i+1)}) = (x^{(i)}, z^{(i)})$ otherwise.

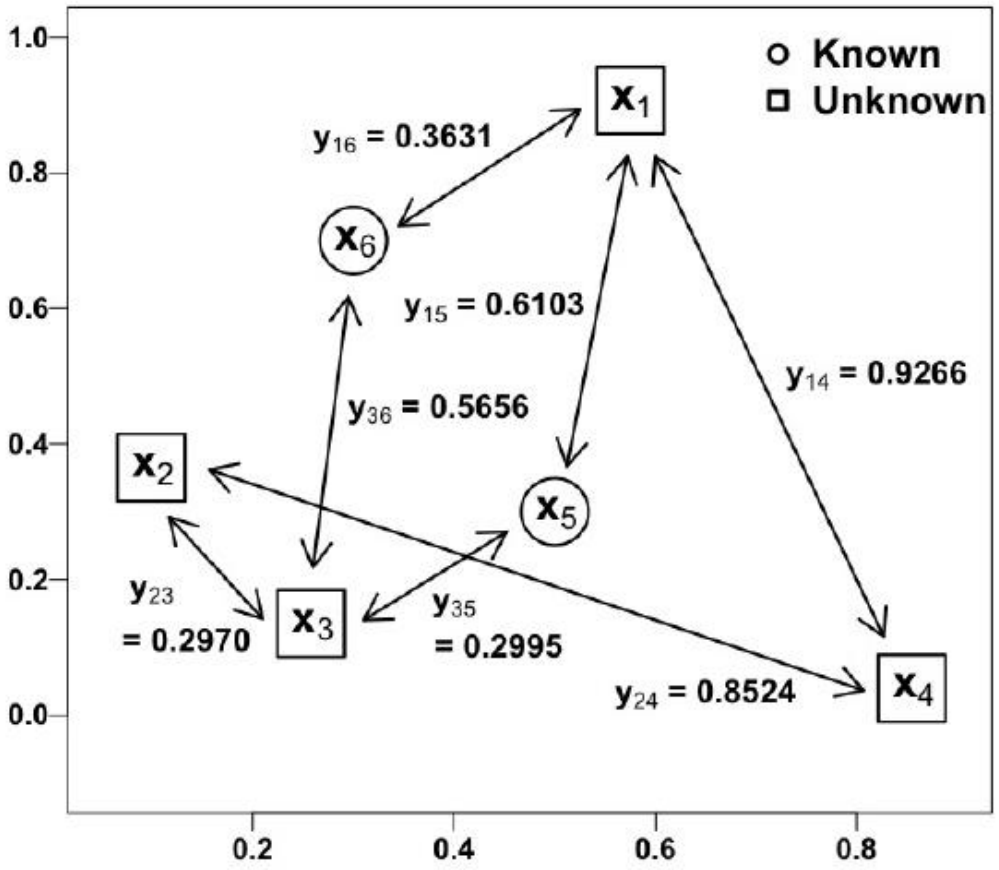
Sensor Network Localization

- Sensor localization is a critical step for effective application of large sensor networks and manual calibration of each sensor may be impractical.

- Sensor localization is obtaining estimates of each sensor's position as well as accurately representing the uncertainty of each estimate.

- Here, we consider a realistic example from Ihler et al. (2005): Searching for unknown sensor locations within a network using the noisy distance data. This problem is known to produce a high-dimensional, banana-shaped, and multimodal joint posterior distribution.

- We assume that the locations of the last two sensors, x5 and x6 are known and the locations of the other sensors, x1; x2; x3, and x4, are unknown parameters of interest.



Sensor Network Localization

- The Likelihood function is defined as below

$$L(x_1, x_2, x_3, x_4) \propto \prod_{j>i} \left[\exp\left(-\frac{(y_{ij} - \|x_i - x_j\|)^2}{2 \times 0.02^2} \right)^{w_{ij}} \times \exp\left(-\frac{w_{ij} \times \|x_i - x_j\|^2}{2 \times 0.3^2} \right) \times \left(1 - \exp\left(-\frac{\|x_i - x_j\|^2}{2 \times 0.3^2} \right) \right)^{1-w_{ij}} \right]$$

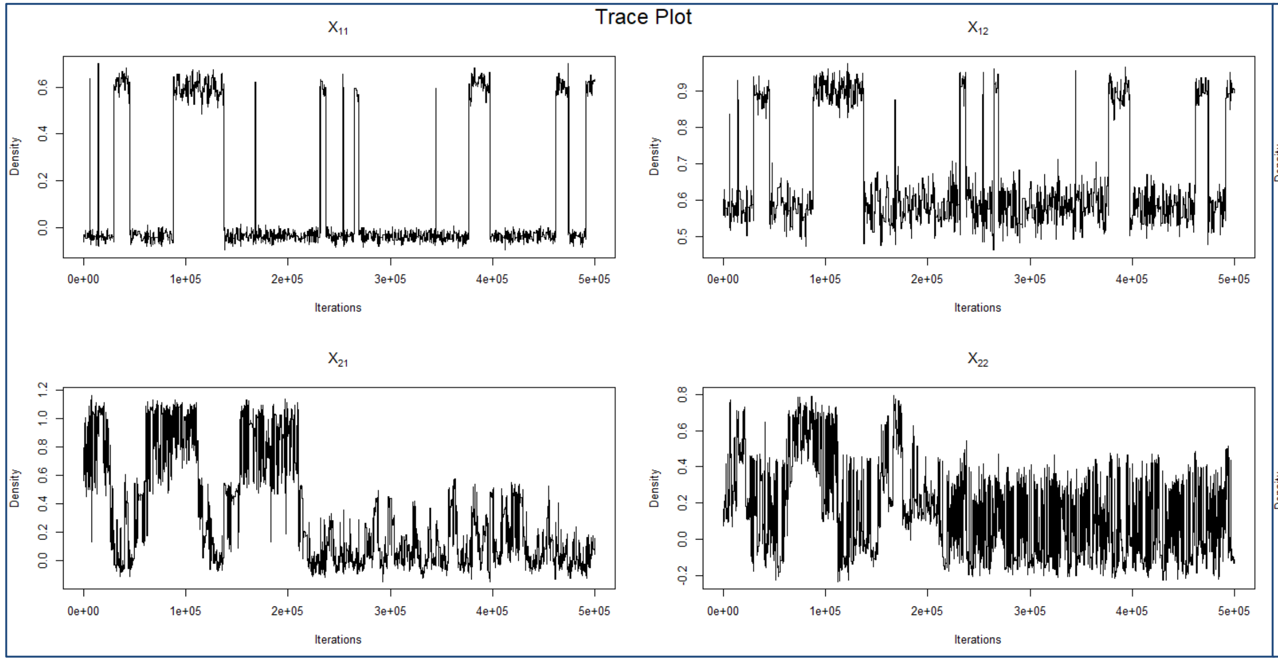
- The full posterior distribution is

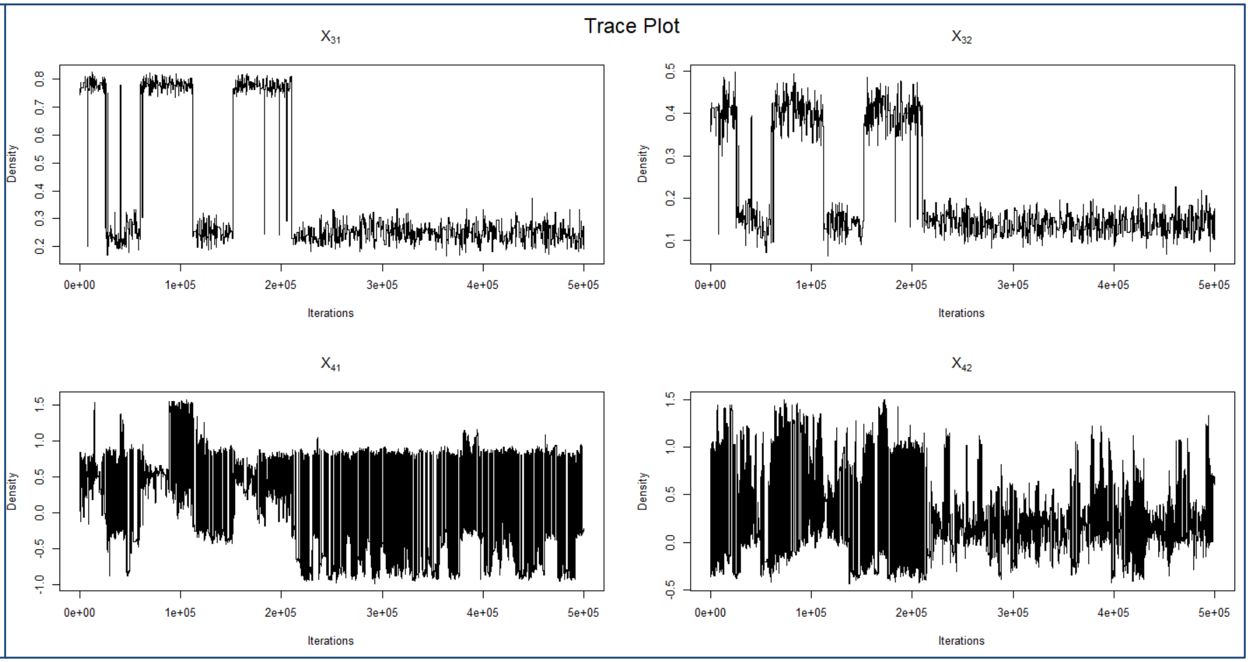
$$\pi(x_1, x_2, x_3, x_4 \mid y, w) \propto L(x_1, x_2, x_3, x_4) \times \exp\left(-\frac{\sum_{k=1}^{4} x_k^\top x_k}{2 \times 10^2} \right)$$

Sensor Network Localization - RESULTS

TRACE PLOT

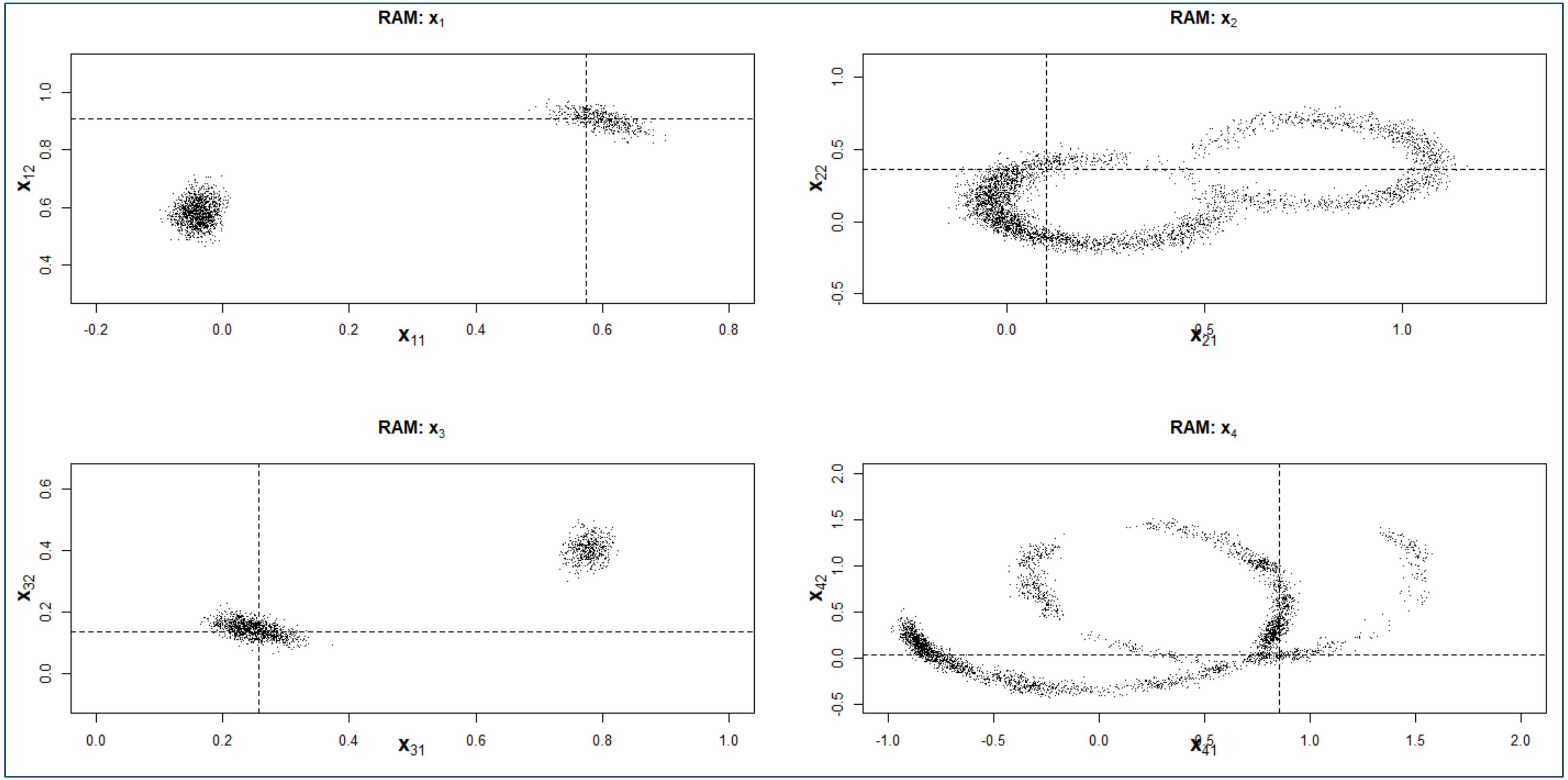
- The trace plot, shows the sampled values of a parameter over time. This plot helps to study the convergence of MCMC procedure.



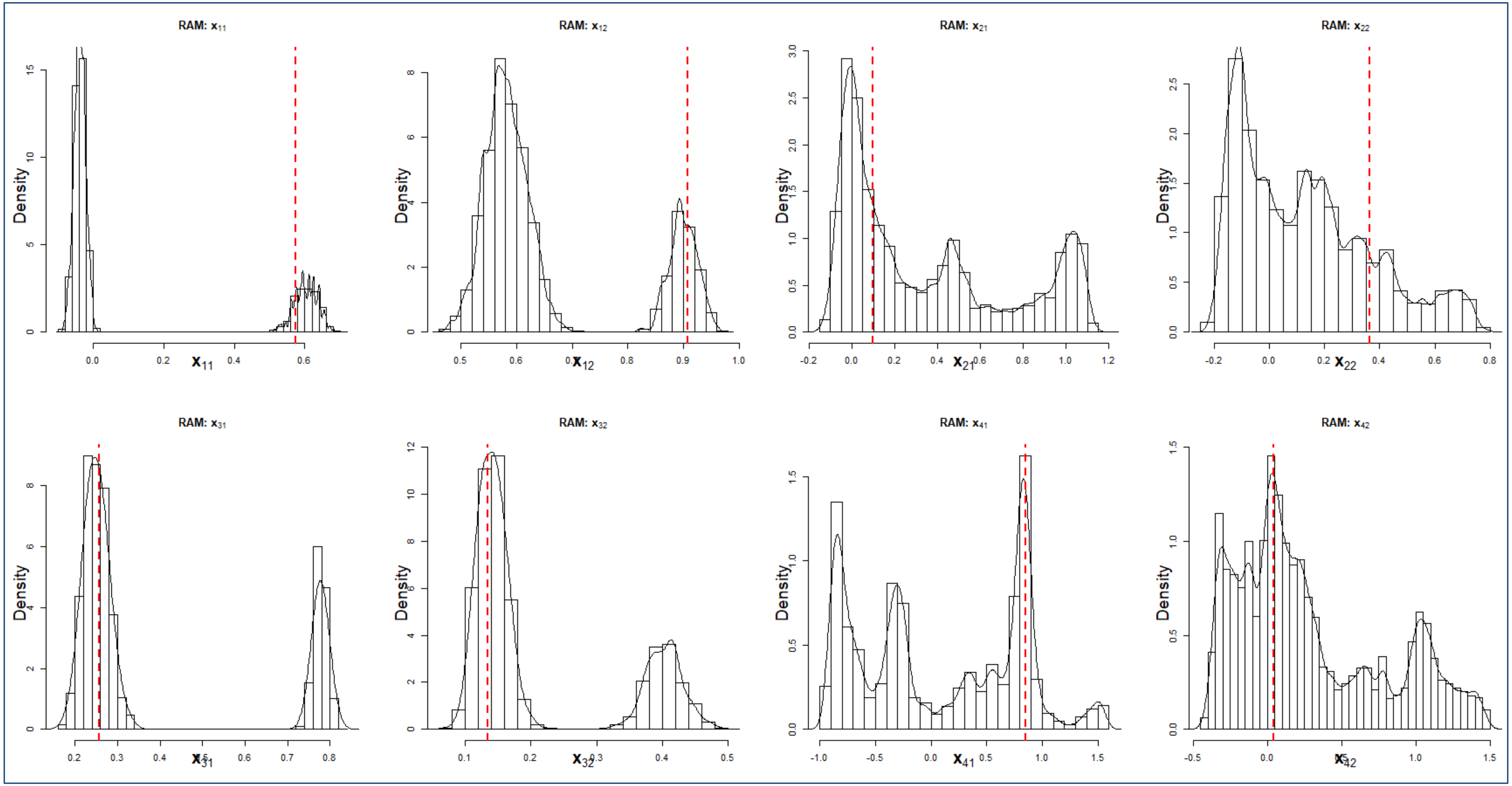


Sensor Network Localization - RESULTS

Scatterplots of the posterior sample of each location

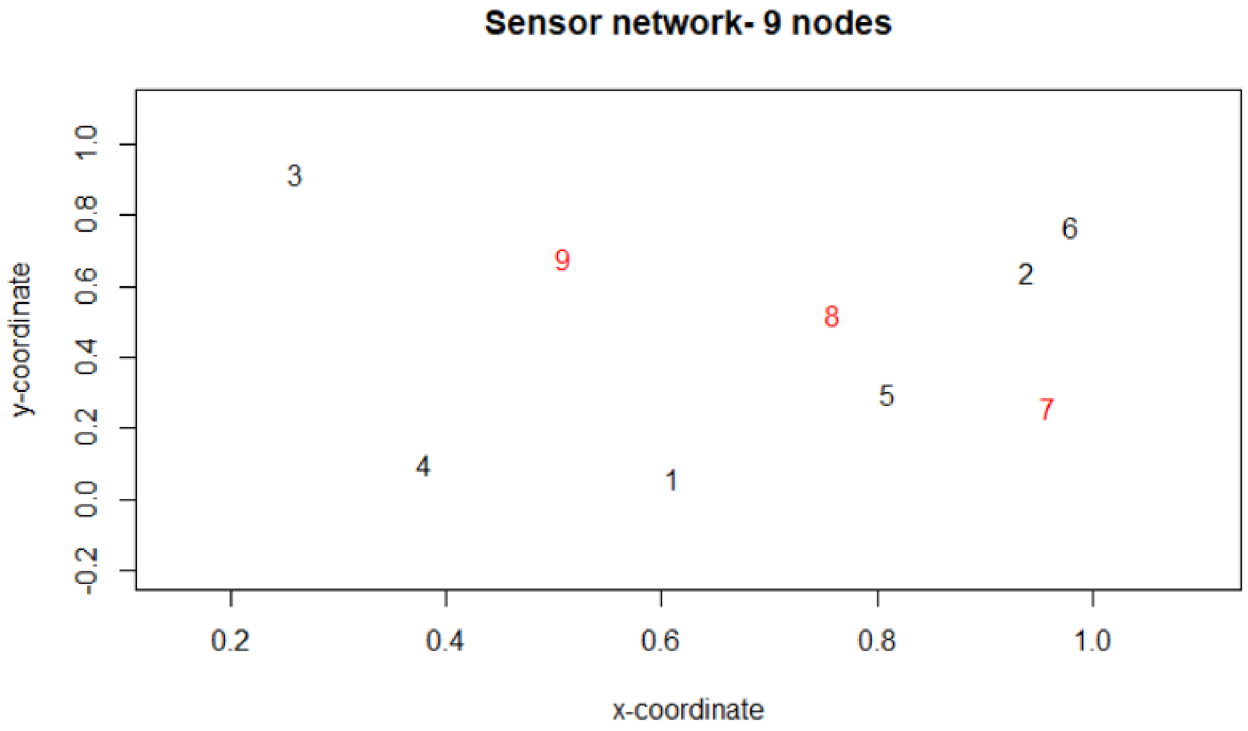


Sensor Network Localization - Histograms



Simulated sensor network

- We simulate a sensor network similar to the previous network but with 9 nodes.
- Nodes 7-9 are assumed to be known whereas we need to estimate parameters for nodes 1-6.
- 9 points are randomly simulated with coordinates between 0 and 1 and then distance is measured, random error is added and some of the distances are randomly removed.
- The simulated sensor network is illustrated below.

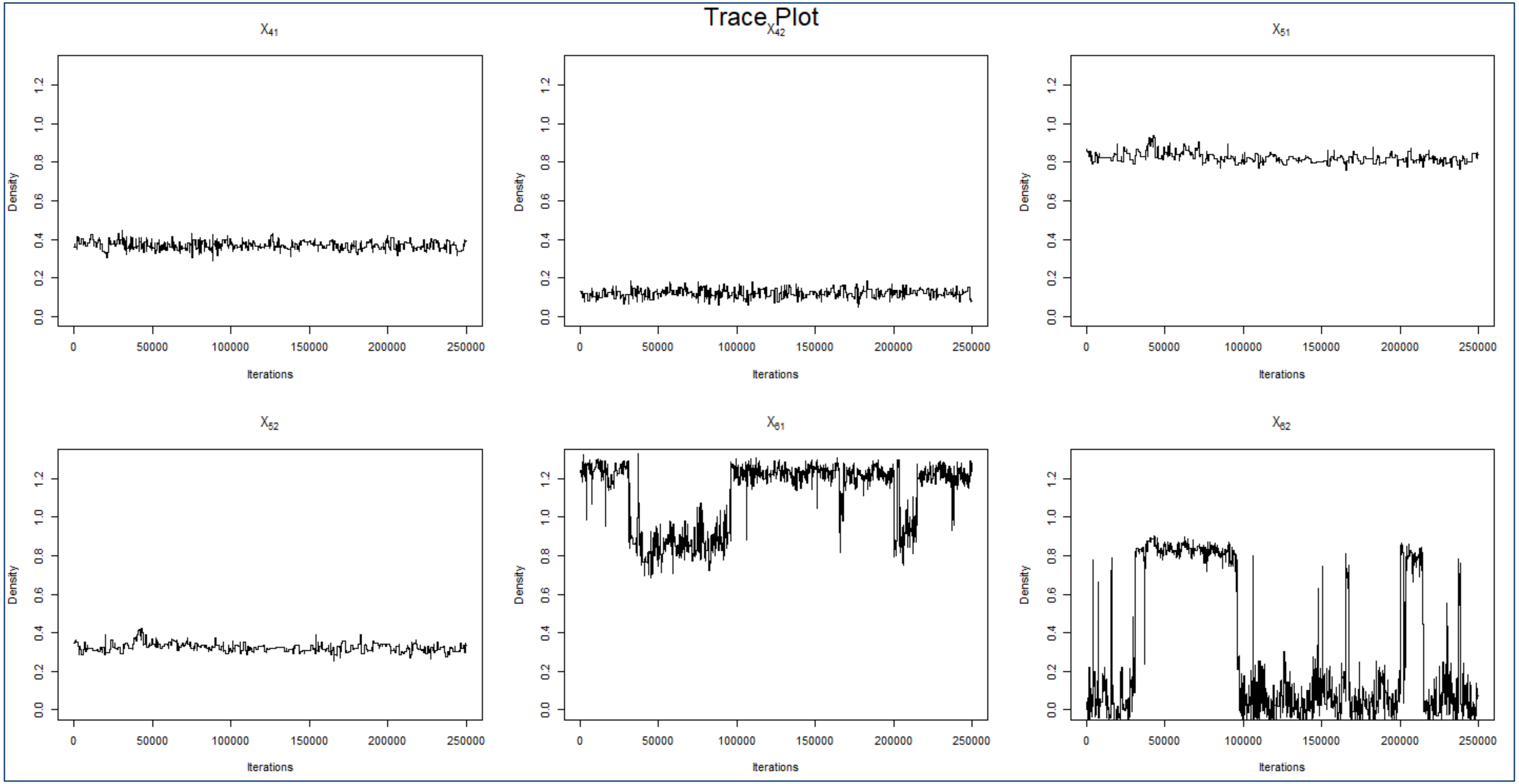


- Known points – RED
- Unknown points - BLACK

Simulated sensor network- Trace plots

TEXAS A&M UNIVERSITY

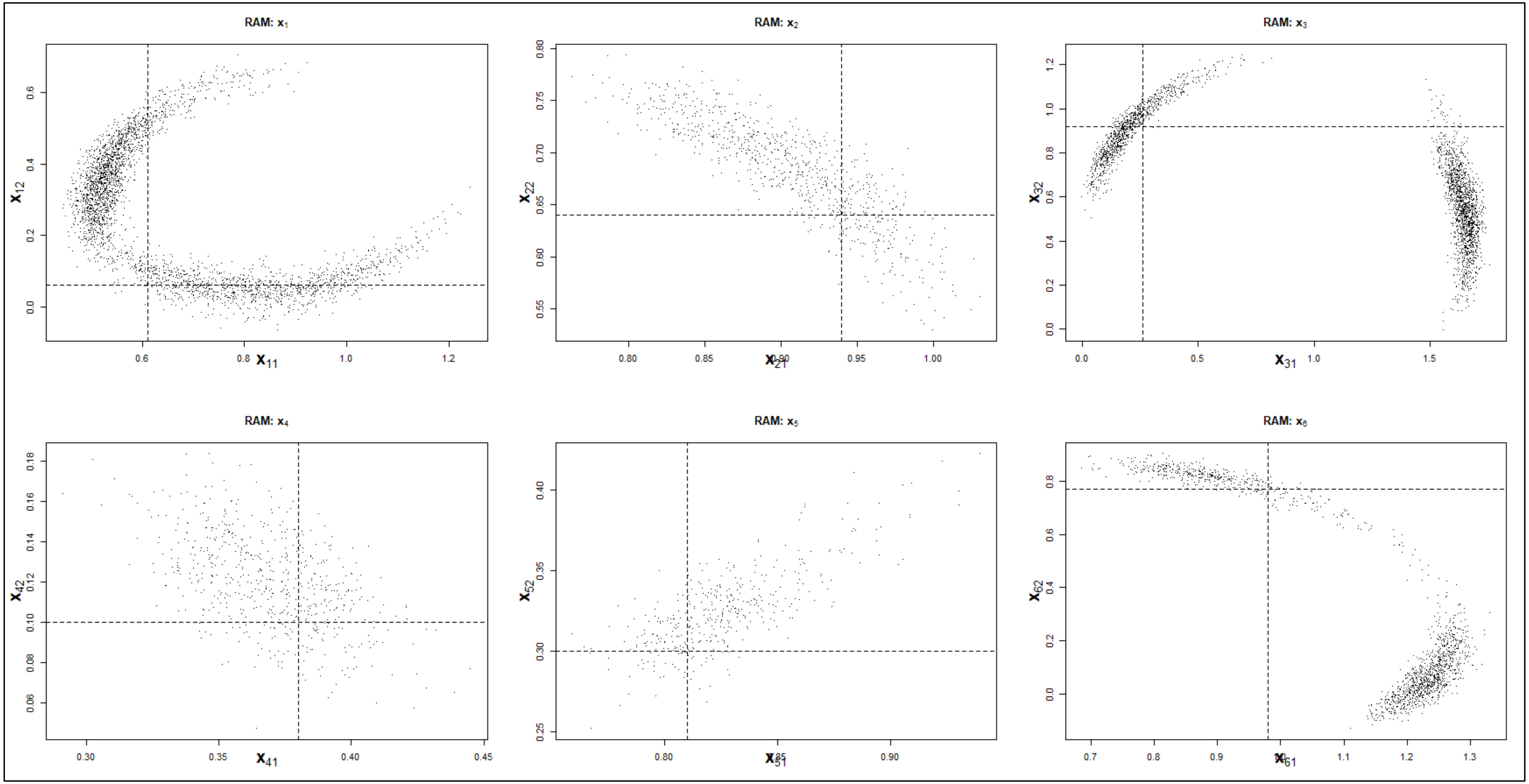
Trace Plots of 6 out of 12 parameters



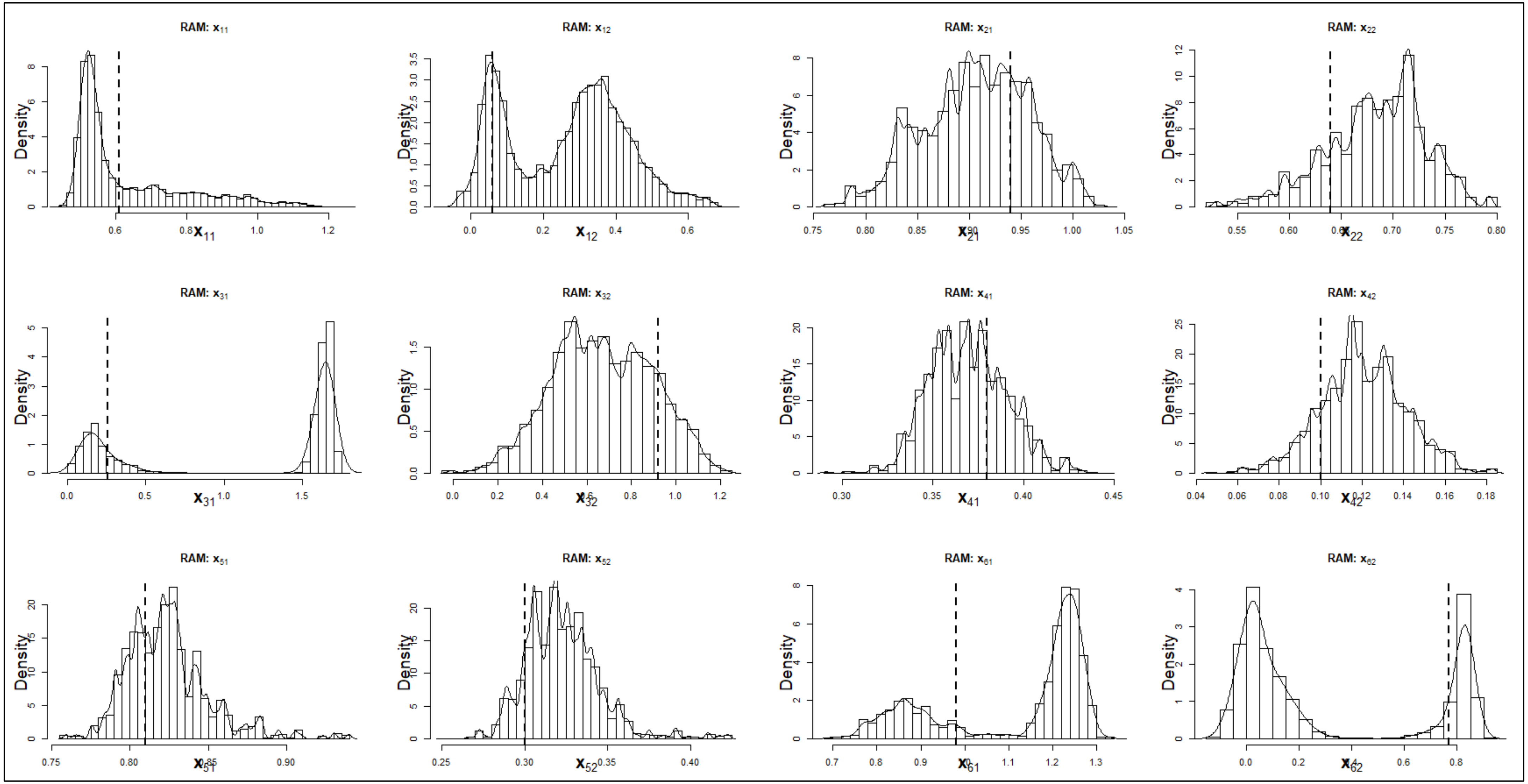
Number of Iteration: 250,000

Normal Proposal density with SD = 1.08

Simulated sensor network- Scatterplots



Simulated sensor network- Histograms



Sensor Network Localization- Performance Statistic

TEXAS A&M UNIVERSITY

Performance statistic of RAM on 7-sensor network problem.

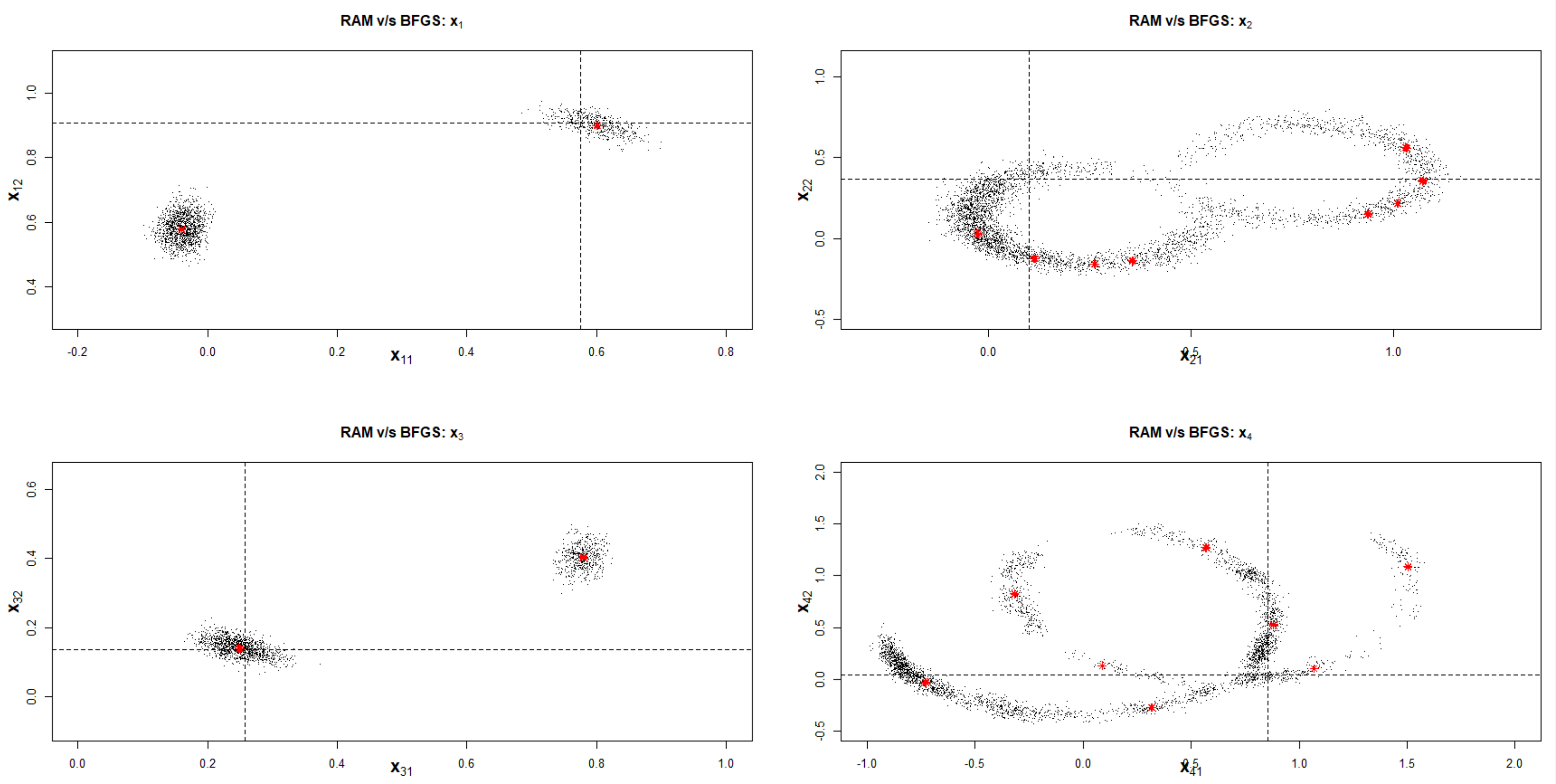
Parameters	No. of iterations	Burn in	Avg. No. of downhill proposals at each iteration	Avg. no. of uphill proposals at each iteration	Avg. No. of downhill proposals for auxiliary variable	Total Proposals	Average acceptance rate	Total accepted proposals
X1	500,000	200,000	1.0001	7.19	1.07	9.26	0.003625	1632.5
X2	500,000	200,000	1.0003	6.54	1.07	8.61	0.0084	4200
X3	500,000	200,000	1.0001	7.29	1.05	9.34	0.00348	1740
X4	500,000	200,000	1.0003	6.98	1.13	9.11	0.007475	3737.5

RAM v/s BFGS

Comparison of accuracy of RAM (250k samples) & BFGS (1000 runs) results on the Sensor network

- BFGS - an iterative method for solving non-linear optimization problems
  - gives a point solution i.e. a posterior mode.

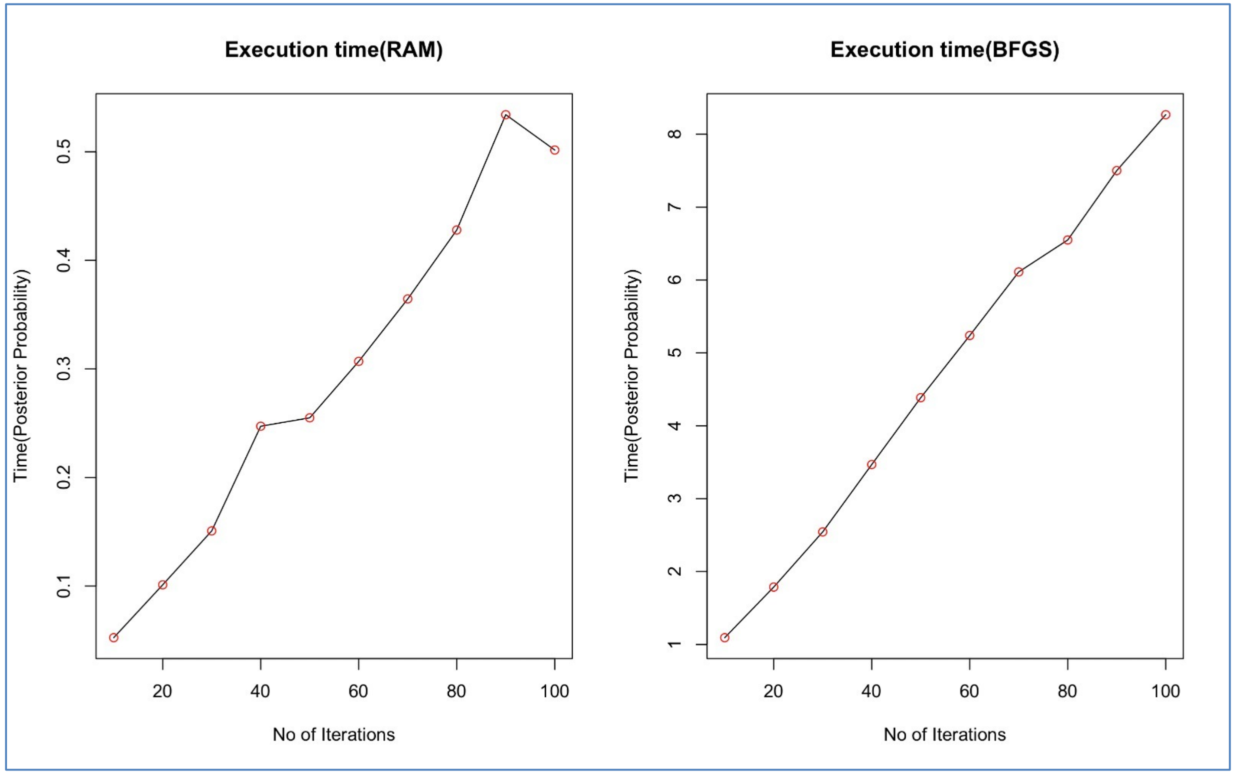
RAM on the other hand gives sample from the posterior distribution.



RAM v/s BFGS

- We compare the computational speed of RAM v/s BFGS
- The Computational speed for the RAM method is very high compared to BFGS method for same no. of iterations.
- The time comparison is done using Microbenchmark
- Ideally RAM is used to calculate the posterior distribution for large sample of order 10^4
- The time complexity of both methods - O(n)
- For comparing the methods,

RAM	BFGS
220,000 iterations – 2700 s	1000 iterations - 78 s



Exoplanet dataset

- We use RAM sampler to sample posterior from an exoplanet dataset.
- We had 6 simulated datasets with anywhere from zero to three planets in each dataset.
- For the purpose of our model generation, we used the first dataset and a 1 planet model.
- Dataset properties:
  - The dataset is a simple timeseries, including the time of observation (t_i), measured" radial velocity (v_i), and a measurement uncertainty (σ_i).
  - Number of observations: nobs = 200
  - Observing baseline: 600 days
  - The dataset includes between zero and three planets (inclusive).
  - The dataset includes a single velocity offset and correlated, Gaussian noise to represent stellar activity.

Exoplanet dataset- Statistical Model

- Each simulated data point is generated according to

$$v_i = v_{\text{pred}}(t_i|\theta) + \epsilon_i,$$

where the first term is the velocity predicted at time ti by a model parameterised by ϴ and εi.

- The appropriate likelihood is a multi-variable normal distribution, centered on the predictions of the model.

$$\log \mathcal{L}(\theta) = -\frac{1}{2}(\mathbf{v} - \mathbf{v}_{\text{pred}}(\theta))^T \Sigma^{-1} (\mathbf{v} - \mathbf{v}_{\text{pred}}(\theta)) - \frac{1}{2}\log|\det \Sigma| - \frac{n_{\text{obs}}}{2}\log(2\pi)$$

- The Gaussian noise is correlated from one observation to the next. The covariance matrix is given by

$$\Sigma_{i,j} = K_{i,j} + \delta_{i,j}\left(\sigma_i^2 + \sigma_J^2\right)$$

- For the quasi-periodic kernel $K_{i,j}$, we assume

$$K_{i,j} = \alpha^2 \exp\left[-\frac{1}{2}\left\{\frac{\sin^2[\pi(t_i - t_j)/\tau]}{\lambda_p^2} + \frac{(t_i - t_j)^2}{\lambda_e^2}\right\}\right],$$

Exoplanet dataset- Priors

- There are 7 parameters considered in a 1 planet model. The prior of these parameters are

Para.	Prior	Mathematical Form	Min	Max
$T(days)$	Jeffreys	$\frac{1}{T \ln\left(\frac{T_{max}}{T_{min}}\right)}$	39.81	44.66
$K(ms^{-1})$	Mod. Jeffreys	$\frac{(K+K_0)^{-1}}{\ln\left(\frac{K_0+K_{max}}{K_0}\right)}$	1.0	999.0
$V(ms^{-1})$	Uniform	$\frac{1}{V_{max}-V_{min}}$	-1000	1000
e	Uniform	1	0	1
ϖ	Uniform	$\frac{1}{2\pi}$	0	2π
χ	Uniform	1	0	1
$s(ms^{-1})$	Mod. Jeffreys	$\frac{(s+s_0)^{-1}}{\ln\left(\frac{s_0+s_{max}}{s_0}\right)}$	1	99

where T – Planet's Orbital Period

K – Planet's RV Semi Amplitude

e – Planet's eccentricity

w – Planet's argument of pericenter

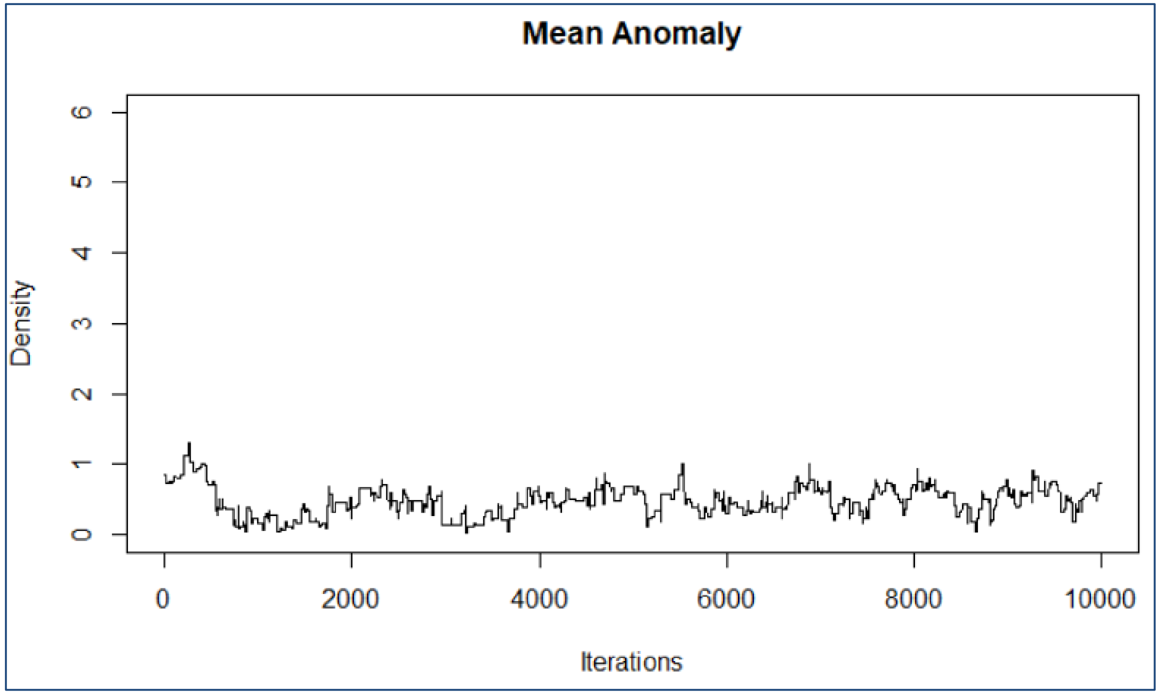
X – Planet's mean anomaly

s – Additional white noise term

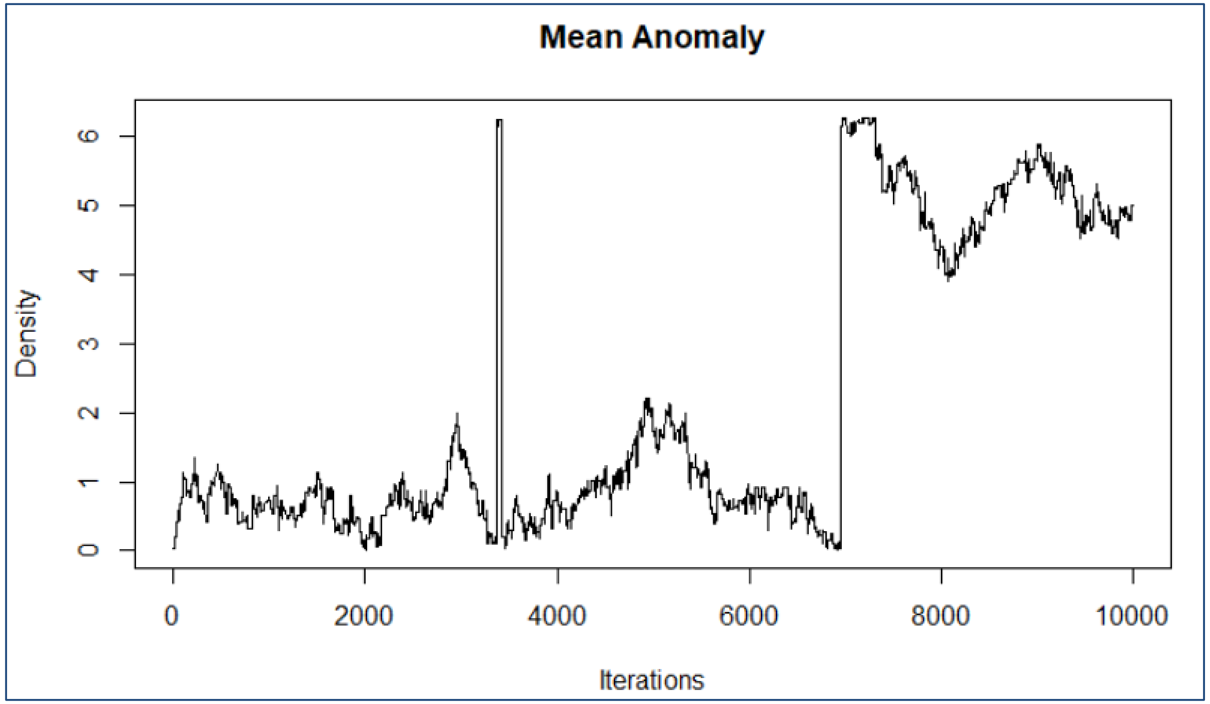
V – RV Velocity offset

Exoplanet dataset - RAM v/s Metropolis

- We compare the trace plots of one of the parameter using Metropolis and RAM with same number of runs.
- We can see that RAM is more likely to jump between modes than Metropolis.

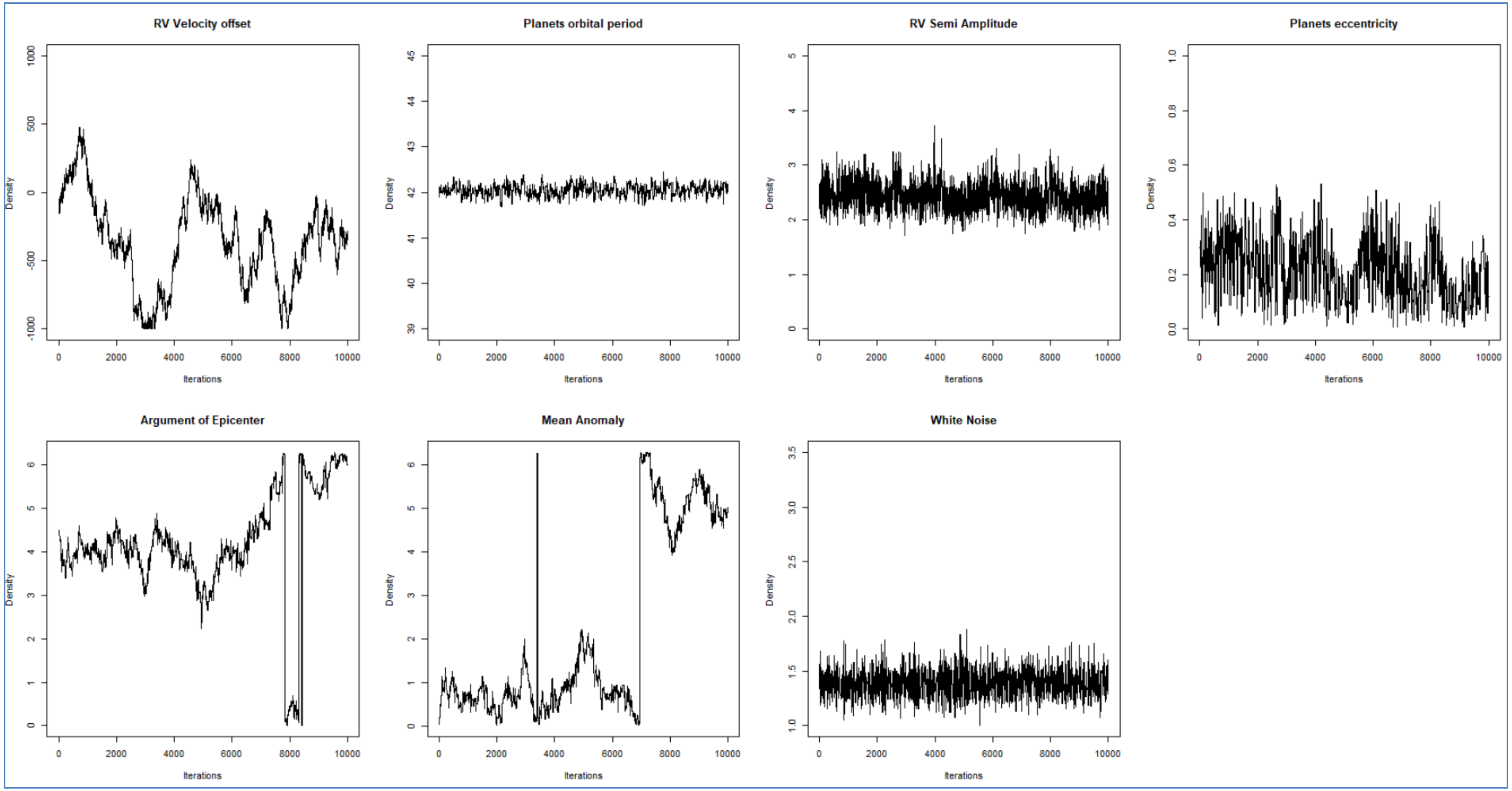


Metropolis

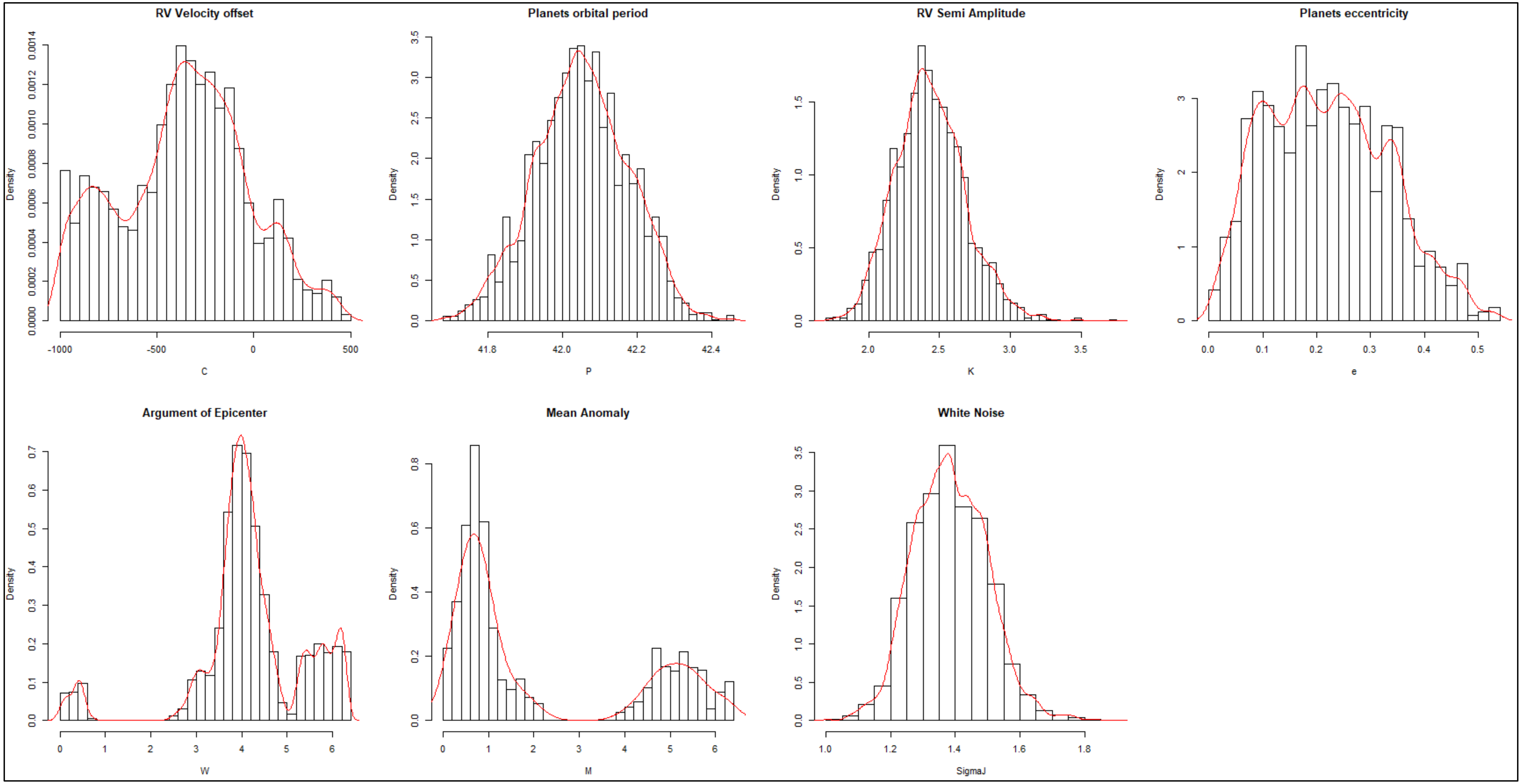


RAM

Exoplanet dataset- TRACE PLOTS



Exoplanet dataset- Histogram



QUESTIONS?

THANK YOU