MDI SMC: Progress Report

Nathan Cunningham

August 16, 2016

Work completed

Algorithm 1 Gibbs sampler

```
1: Initialise $\Gamma$ matrix of prior allocation weights and $\Phi$ matrix of dataset concordance values
2: for i = 1, ..., number of iterations do
3:     Conditional on $\Gamma_{i-1}$ and $\Phi_{i-1}$ update the cluster labels, $c_i$, using alg. 2
4:     Conditional on $c_i$ update $\Gamma_i$ and $\Phi_i$
5: end for
```

Algorithm 2 Particle filter to update cluster allocations

```
1: for i = 1, ..., n do                                              ▷ Loop over observations
2:     for m = 1, ..., M do                                          ▷ Loop over particles
3:         for j = 1, ..., d do                                      ▷ Loop over datasets
4:             Sample $c_{i,j}^{(m)}$                                ▷ Propose a cluster for each datum
5:             $q(c_{i,j}^{(m)} = k) \propto k^*(y_{i,j}|c_{i,j}^{(m)} = k)\gamma_{i,k,j}(1+\phi_i)$
6:             $\xi^(m) = \xi^(m) + \gamma_{i,k,j}(1+\phi_i)k^*(y_{i,k}|c_{i,j}^{(m)} = k)$
7:         end for
8:     end for
9: end for
```

Where

$$k^*(y_{i,k}|c_{i,k}^{(m)} = k) = (\mathbf{y_{i,k}} - \mu_{\mathbf{k}})\mathbf{\Sigma^{-1}}(\mathbf{y_{i,k}} - \mu_{\mathbf{k}})^\top \tag{1}$$

$$\Phi \text{ is a measure of cluster label correspondence across datasets} \tag{2}$$

$$\gamma_{i,k,j} \text{ is a prior weight for assigning observation i, in dataset k to cluster j} \tag{3}$$

Previously, it was suggested that eqn 1 be calculated as the `logMarginalLikelihood` value output from the `Gaussian` function. However, this represents the likelihood of the cluster given the data point being assigned to it, as opposed to the likelihood of the data arising from a specific cluster. Using this as such resulted in all observations being assigned to a single cluster, or being allocated incorrectly to clusters. Using eqn 1

- Empty clusters are assumed to arise from a multivariate standard normal distribution, that with mean-vector 0 and covariance matrix equal to the identity matrix. The data are normalised prior initially, so this is reasonable.

- Non-empty clusters are assumed to have covariance matrix:

$$\frac{\mathbb{I}}{((n_{c,k,m}/a^(m)) + (1/(1-a^(m)))} \tag{4}$$

  where $n_{c,k,m}$ represents the number of observations in dataset k have been assigned to cluster c in particle m and $a^{(m)}$ is a measure of central tendency sampled for each particle from a beta distribution. This is based on the code in [1] and performs better than calculating a covariance from the observations already allocated to the cluster.

These problems become less prevalent as you progress through observations as clusters are more ‘settled’. This means that the clustering of data early in the dataset can be more uncertain. This can be seen in Fig. 1 where the data include two clusters, sorted within the dataset such that cluster 2 occurs subsequent to cluster 1. Cluster 2 is well-discovered by the algorithm, as evidenced by the large dark grey square. Cluster 1 is much more poorly-defined. The tick marks are the suggested partition of the data into clusters (the algorithm is run here attempting to partition the data into 5 clusters).

This is in contrast to Fig. 2 where there is clear evidence of two distinct clusters. As the model is trying to partition the data into 5 subsets, the clusterings are more uncertain. Re-running the algorithm to search for two components gives greater certainty.

The dataset used for this is trivially simple to cluster, as shown in the data are very distinctly separable. I also tested a more complicated dataset, shown in Fig. . The data were generated with five clusters. I ran the algorithm to search for a maximum of 10 clusters for relatively few iterations (100) and with relatively few particles (10), yet the data appears to successfully recover the true five clusters.

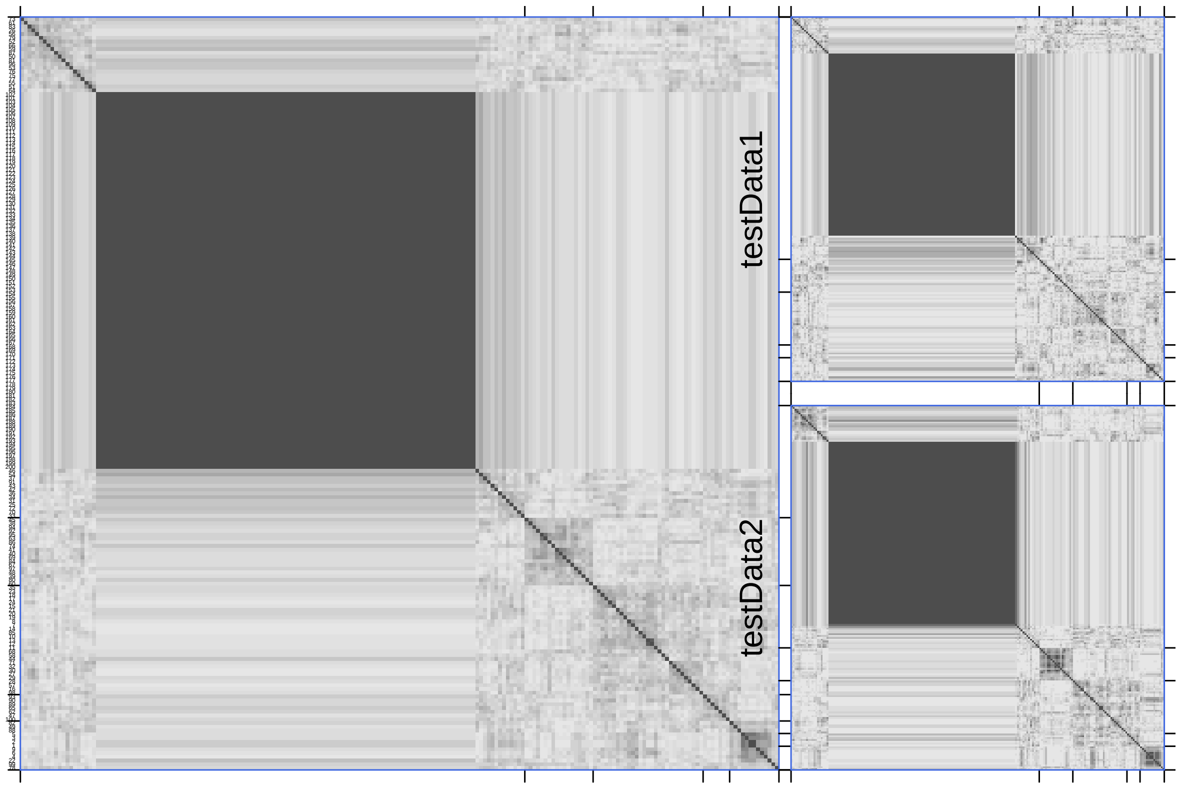

Figure 1: Cluster allocation agreement for mixture of Gaussian data with two clusters. Darker colours indicate greater agreement of allocations across runs of the algorithm. The smaller panels represent the posterior similarity matrix for each dataset, while the larger panel represents the consensus. The data are not shuffled prior to running the particle filter.

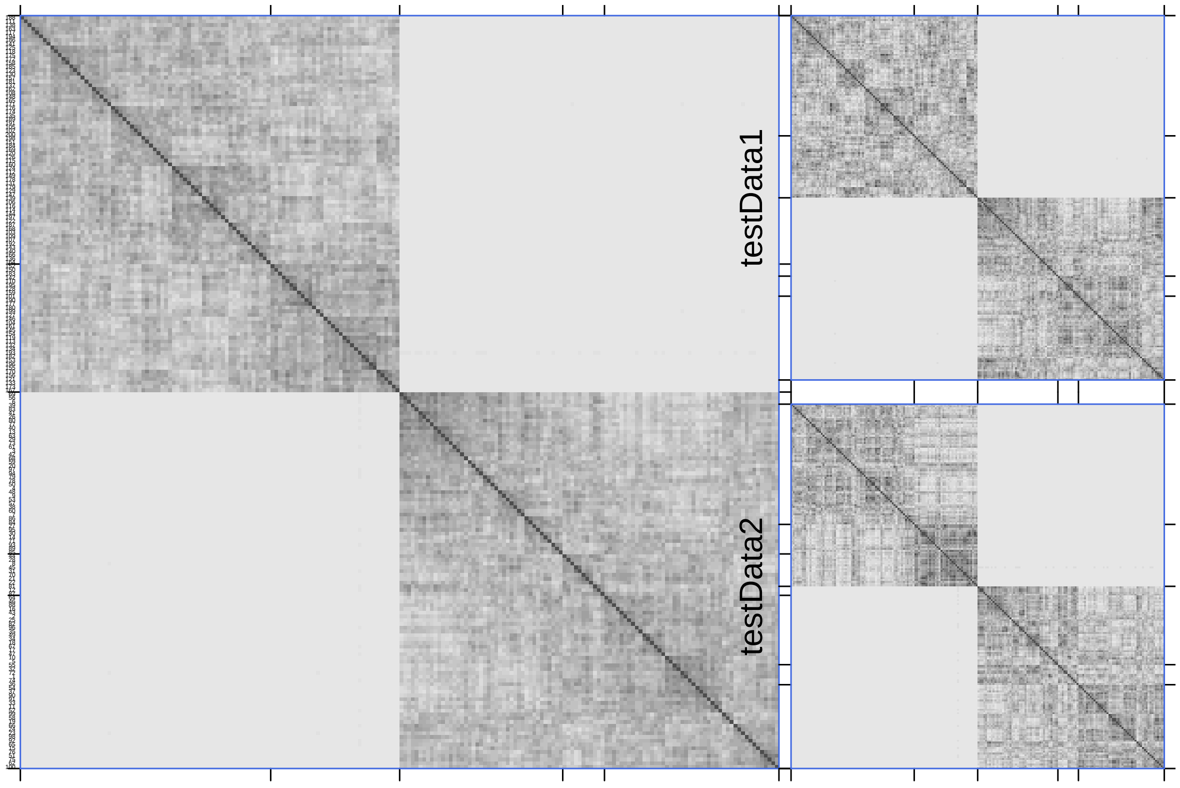

Figure 2: Cluster allocation agreement for mixture of Gaussian data with two clusters. Darker colours indicate greater agreement of allocations across runs of the algorithm. The smaller panels represent the posterior similarity matrix for each dataset, while the larger panel represents the consensus. The data are shuffled prior to each run of the particle filter.

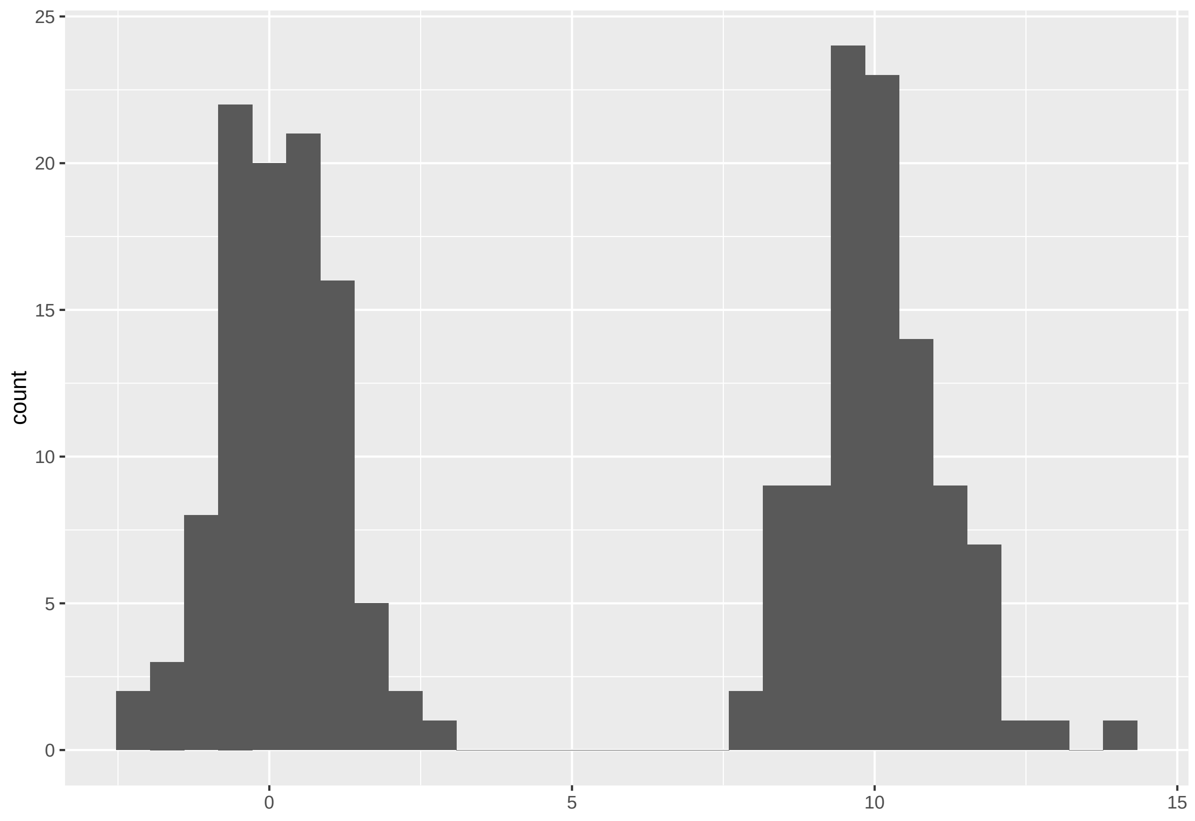

Figure 3: Synthetic data used for testing with two very distinct clusters defined by a gaussian distribution.

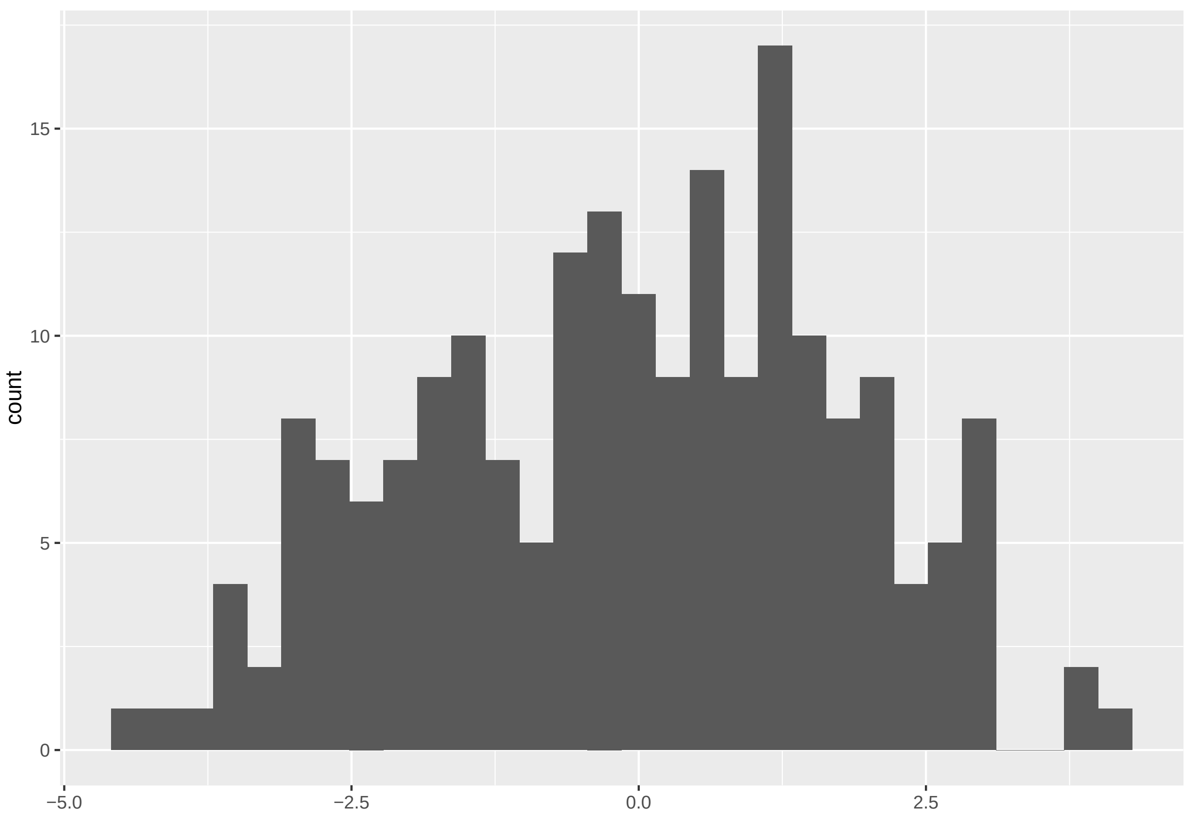

Figure 4: Synthetic data used for testing with five non-distinct clusters each originating from a gaussian distribution.

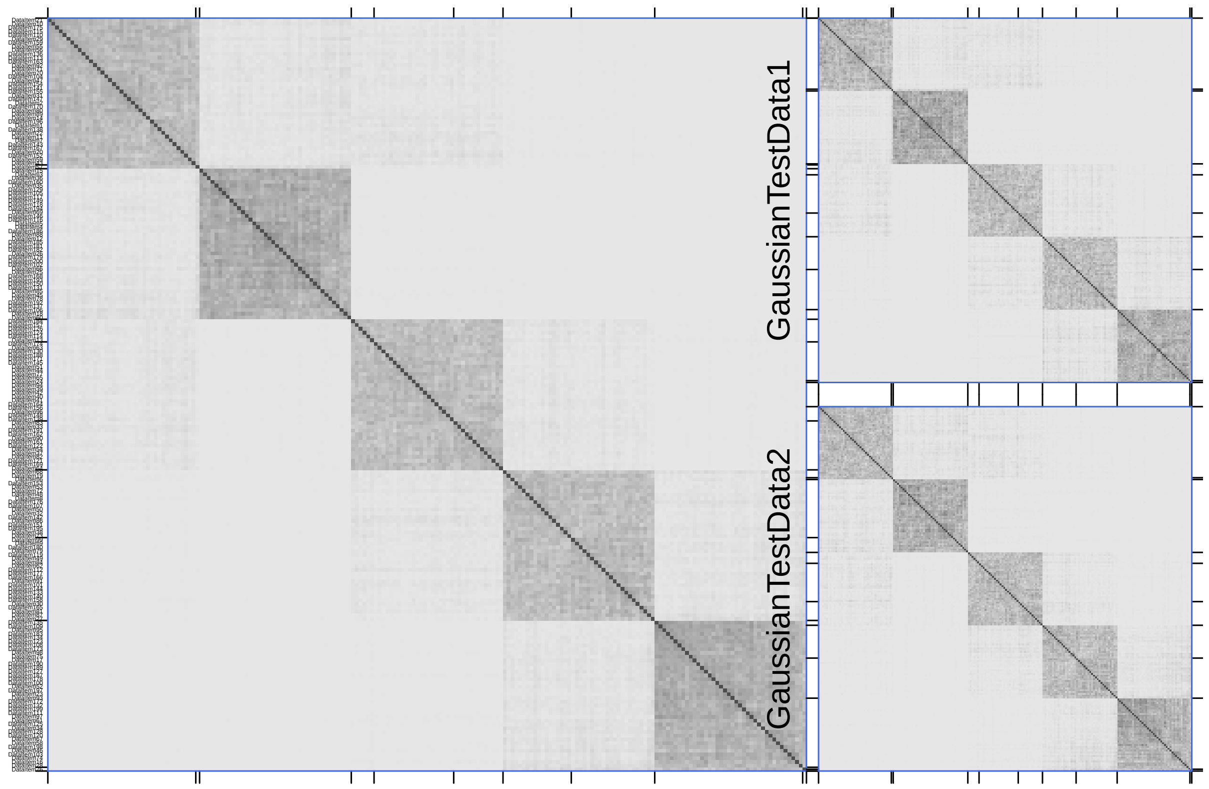

Figure 5: Posterior similarity matrix for mixture of Gaussian data with five clusters. Darker colours indicate greater agreement of allocations across runs of the algorithm. The smaller panels represent the posterior similarity matrix for each dataset, while the larger panel represents the consensus. The algorithm is aiming to looking to partition the data into 10 clusters, however it appears to find the 5 true clusters.

Problems

- Is the form of the covariance matrix an issue? It appears to run well within the dataset.
- Is the shuffling of the data between runs an issue?
- The algorithm currently is quite slow. Using 100 particles in the more complicated example above can result in a single iteration of the Gibbs sampler taking upwards of 30 seconds to run. Parallelisation may help here.
- As the MDI algorithm originally was not set up for multiple particles, a single particle is selected at the end of the particle filter (based on $xi^(m)$) and that is fed into the next step of the Gibbs sampler. A possible solution would be to treat the particles as like separate datasets, so all the particles are returned and the Γ and Φ values are updated based on all of the particles. Not sure if that's a good idea.

References

[1] JE Griffin. Sequential monte carlo methods for mixtures with normalized random measures with independent increments priors. *Statistics and Computing*, pages 1–15, 2014.