2 Cluster estimation

Suppose we wish to estimate the real clustering of the data, assuming the DPM model holds true. A first rough estimate is the *final clustering*, that is, the state values corresponding to the last iteration of the algorithm. This estimate does not require an appropriate function to be implemented, since the state values are already available in `allocations` and `unique_values` after the algorithm is `run()`. However, due to oscillating behavior, the last clustering may not be the optimal one. Instead, we chose to implement a *least square* estimate in the following function:

```
unsigned int cluster_estimate();
```

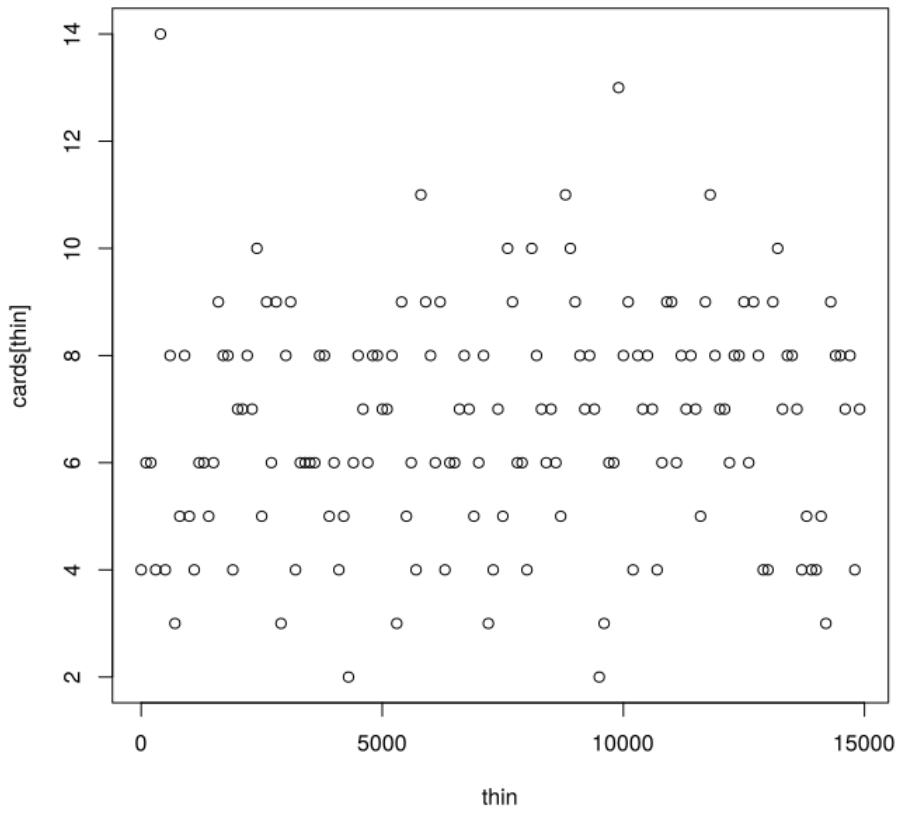
This function exploits the `chain` pseudo-vector, in which states of all iterations of the algorithm were saved via `save_iteration()` (of course, only after the burn-in phase) and the `protobuf` library. This function loops over all `IterationOutput` objects in `chain`, finds the iteration at which the best clustering occurred, saves the whole object into the `best_clust` class member, and returns the iteration number of this best clustering. As briefly touched upon earlier, the best clustering is found via the minimization of the squared posterior *Binder's loss function*. An equivalent approach is computing the so-called *dissimilarity matrix* for each iteration, computing its sample mean over all iterations, and finding the iteration that is the closest to the mean with respect to the *Frobenius norm*. More specifically, for each iteration k, the dissimilarity matrix $D^{(k)}$ is a symmetric, binary n-by-n matrix (where n is the number of available data points) whose entries $D_{ij}^{(k)}$ are 1 if datum i and j are placed in the same cluster at iteration k and 0 otherwise. After each $D^{(k)}$ and the sample mean $\bar{D} = \frac{1}{K}\sum_k D^{(k)}$ are computed, where K is the number of iterations (not counting the ones in the burn-in phase), the best clustering $\hat{k}$ is found by minimizing the Frobenius norm of the difference with $\bar{D}$:

$$\hat{k} = \arg\min_k \left\| D^{(k)} - \bar{D} \right\|_F^2 = \arg\min_k \sum_{i,j} \left(D_{ij}^{(k)} - \bar{D}_{ij} \right)^2 .$$

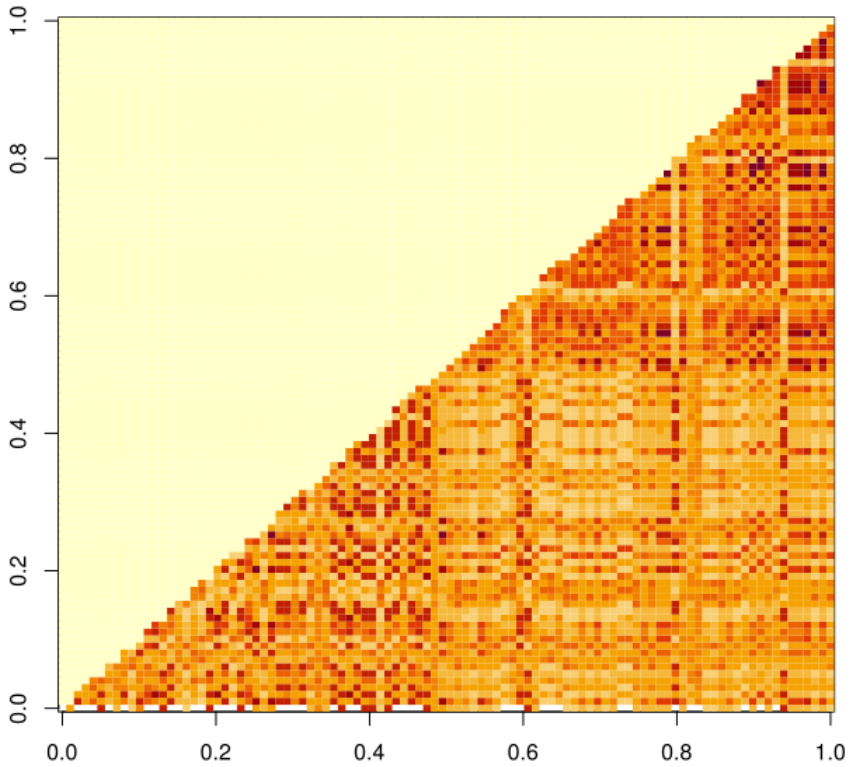
By virtue of the involved matrices being symmetric, the latter summation can be computed over all $i < j$ instead of all i, j for efficiency.

3 Clustering analysis

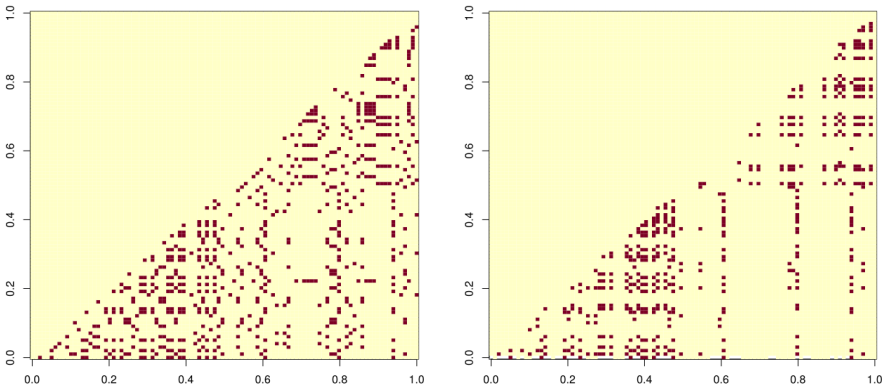
Our clustering analysis was conducted on 100 data points, the former 50 of which were iid sampled from a $\mathcal{N}(4, 1)$ and the latter from a $\mathcal{N}(6, 1)$. The `Neal8` algorithm with $m = 3$ auxiliary blocks was run for 20000 iterations, and the first 5000 were discarded as burn-in. Then, `cluster_estimate()` was called. The resulting best clustering is found at iteration 2975 and has 6 clusters in it.
The obtained clusterings are highly fluctuating in the creation and destruction of clusters, as shown by the below plot:

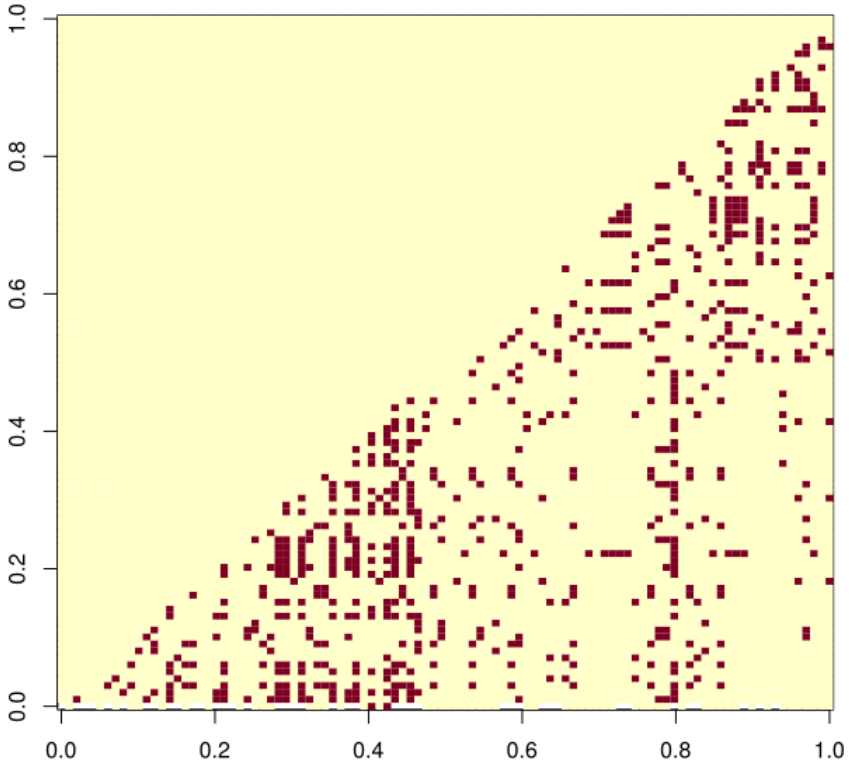


A thinning of one iteration every 100 was performed for better readability of the plot. Here, one can clearly see that the number of clusters at all iterations varies significantly bewteen 2 and 14, even in the last thousands of iterations, with most of the iterations hovering between 6 and 9 clusters.
This shows that the *"best"* clustering in a *least sqaure* sense (via Binder's loss function, as previously discussed) does not necessarily have a small number of clusters. Indeed, let us take a closer look at the structured produced by the `cluster_estimate()` function by looking at the heat map for the mean dissimilarity matrix $\bar{D}$:



Darker, reddish colors indicate values closer to 1 (both data are in the same cluster), while light, yellowish colors indicate values closer to 0 (both data are in different clusters). We only show the lower triangular part, as the matrix is symmetric and the diagonal entries are obviously all equal to 1. In this mean dissimilarity matrix, entries are numbers from 0 to 1, and they can be interpreted as probabilities of data points belonging to the same cluster. More specifically, the more iterations keep datum i and j in the same cluster, the closer to 1 the corresponding entry $\bar{D}_{ij}$. Since actual dissimilarity matrices are binary, the least square approach means finding the matrix which is the closest to the *rounded* mean matrix. On the left side below, we have the heat map for dissimilarity matrix number 2975 (the "best" clustering), which we can compare to the rounded mean dissimilarity matrix on the right side. The bottom matrix instead represents the difference (in absolute value) between the two above matrices. As we can see, the two matrices look pretty similar, but not overwhelmingly so.


0.0
0.2
0.4
0.6
0.8
1.0


0.0
0.2
0.4
0.6
0.8
1.0