Consensus clustering for Bayesian mixture models: Supplementary materials

Stephen Coleman, Paul DW Kirk and Chris Wallace

September 9, 2020

Abstract

Input: Distance between means Δ_μ
A common standard deviation σ^2
A number of clusters K
The number of items to generate in total N
The number of features to generate in total P
An indicator vector of feature relevance $\phi = (\phi_1, \ldots, \phi_P)$
The expected proportion of items in each cluster $\pi = (\pi_1, \ldots, \pi_K)$
A method for sampling x times from the array y, with weights π:
$Sample(y, x, \pi)$
A method for permuting a vector x: $Permute(x)$
A method for generating a value from a univariate Gaussian
distribution with mean μ and standard deviation σ^2: $Gaussian(\mu, \sigma^2)$
Output: A dataset, X
The generating cluster labels $c = (c_1, \ldots, c_N)$
begin

    `/* initialise the empty data matrix                    */`
    $X \leftarrow 0_{N \times P}$;
    `/* create a matrix of K means                          */`
    $\mu \leftarrow (\Delta_\mu, \ldots, K\Delta_\mu)$;
    `/* generate the allocation vector                      */`
    $c \leftarrow Sample(1 : K, N, \pi)$;
    $\mathbf{M} \leftarrow \mathbf{0}_{N \times N}$;
    for $p = 1$ **to** P **do**

        `/* Test if the feature is relevant, if relevant`
        `   generate data from a mixture of univariate`
        `   Gaussians, otherwise draw all items from the same`
        `   distribution                                     */`
        if $\phi_p = 1$ **then**
            $\nu \leftarrow Permute(\mu)$;
            for $n = 1$ **to** N **do**
                $X(n, p) \leftarrow Gaussian(\nu_{c_n}, \sigma^2)$
            end
        end
        if $\phi_p = 0$ **then**
            for $n = 1$ **to** N **do**
                $X(n, p) \leftarrow Gaussian(0, \sigma^2)$
            end
        end
    end
    `/* Mean centre and scale the data                      */`
    $X \leftarrow Normalise(X)$

end

Algorithm 1: Data generation for a mixture of Gaussian with independent features. This algorithm is implemented in the `generateSimulationDataset` function from the `mdiHelpR` package available at `www.github.com/stcolema/mdiHelpR`.
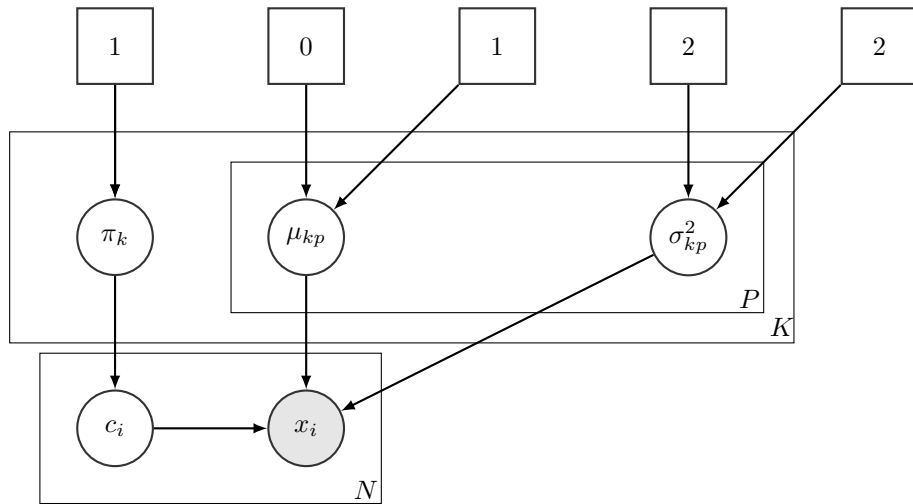
Figure 1: Directed acyclic graph for the Bayesian mixture of Gaussians used.
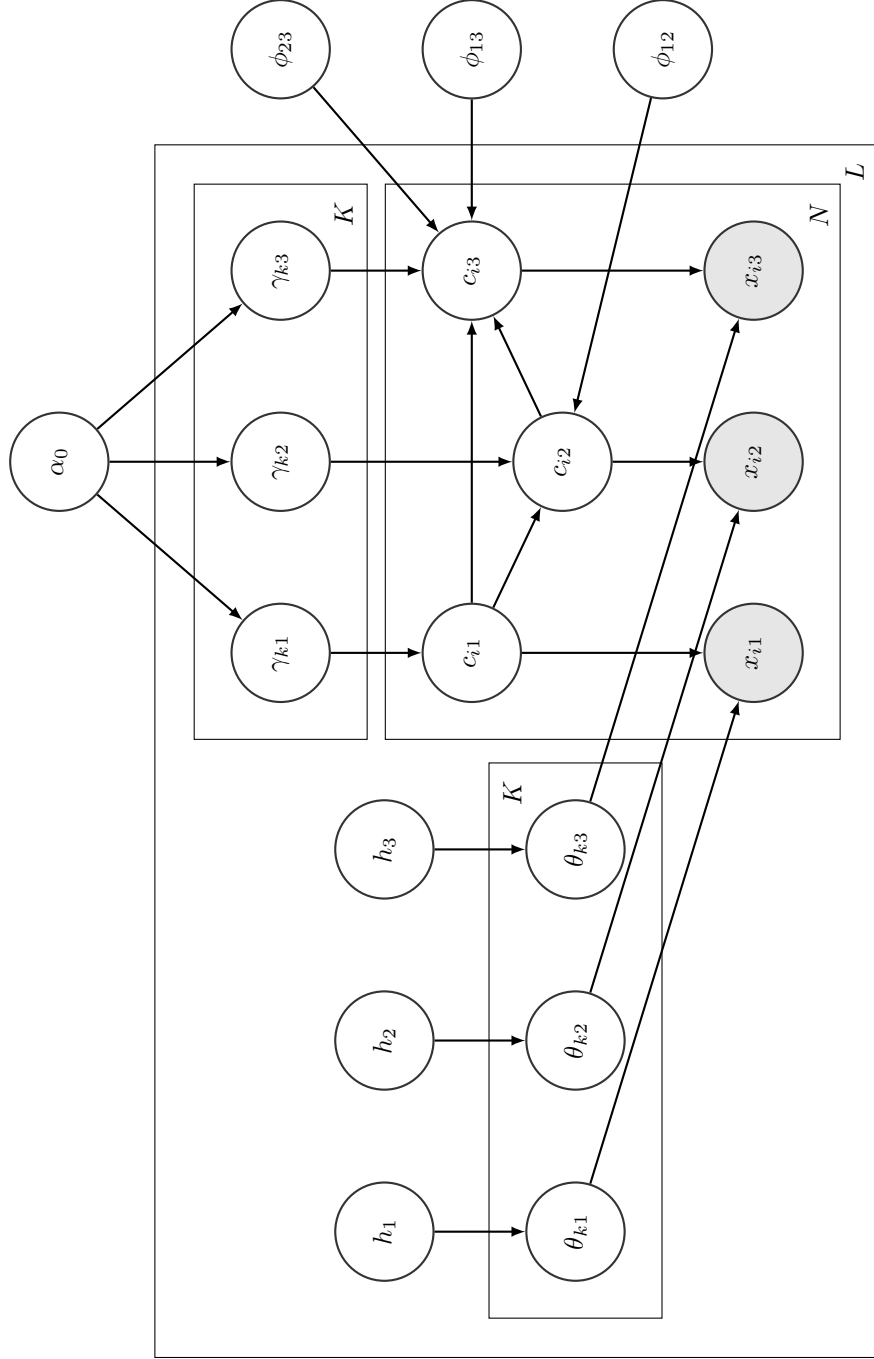
Figure 2: Directed acyclic graph for the Multiple Dataset Integration model for 3 datasets.
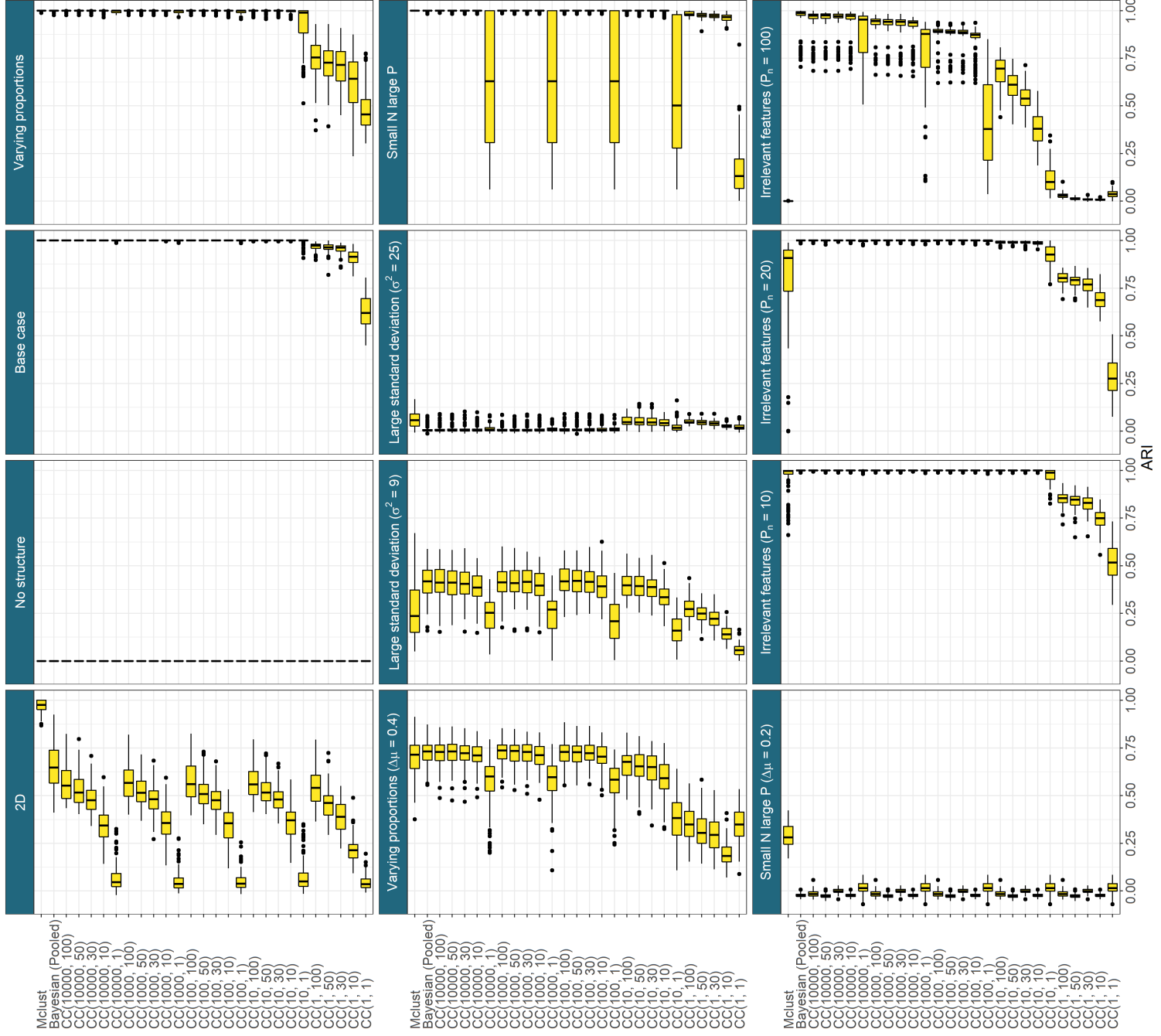
Figure 3: Predictive performance across all simulations. $CC(R, S)$ denotes consensus clustering using the Rth sample from S different chains.
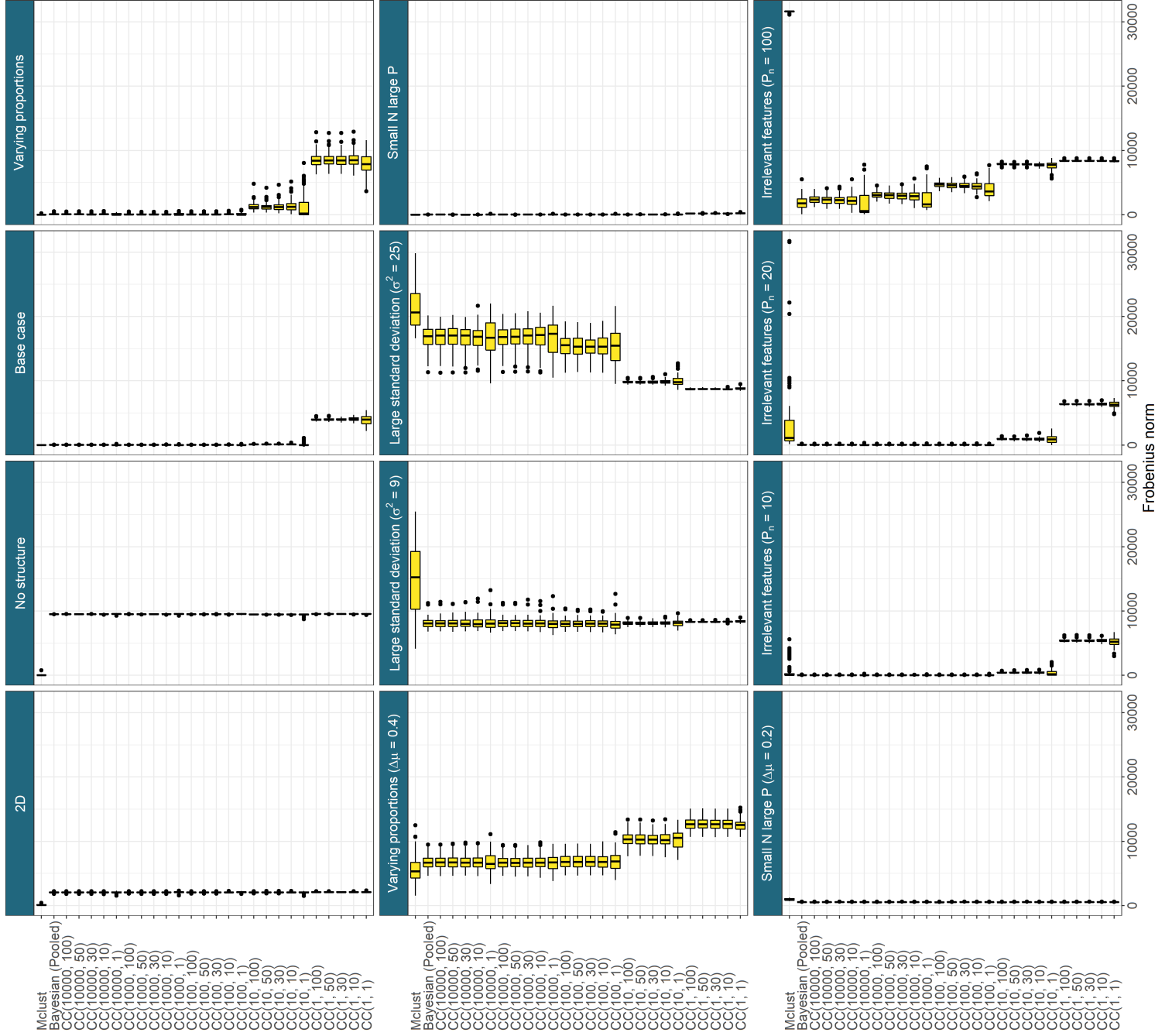
Figure 4: Predictive performance across all simulations. $CC(R, S)$ denotes consensus clustering using the Rth sample from S different chains.