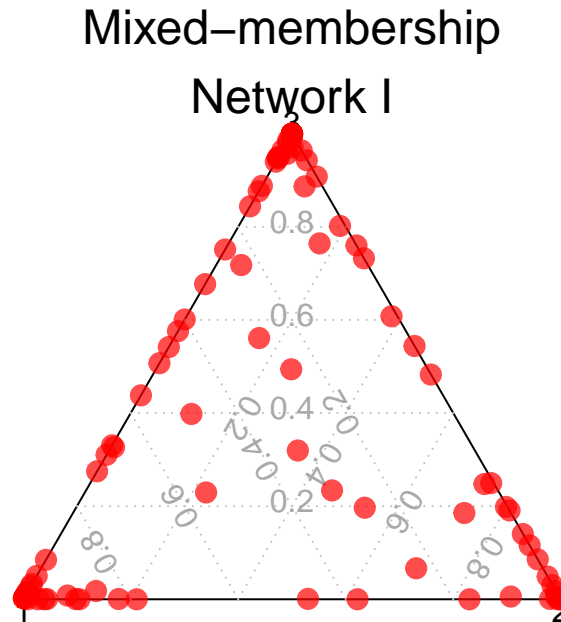


Simulation Experiments

Experiment 1: Static network without covariates.

Our first simulation exercise estimates the original mixed-membership stochastic block model of Airoldi et al. (2008): a model without covariates and without dynamics. We simulate a single network of 100 nodes and 3 latent blocs using the model's DGP. We set the blocmodel parameters to reflect high in-group interaction probabilities and low out-group interaction probabilities, and generate mixed-memberships that are roughly uniform. The following figure displays the mixed-membership vectors for all 100 nodes in our simulated network. While some nodes are clearly in one of the three blocks (and are therefore depicted in one of the vertices of the simplex), many spread their memberships much more evenly among two or three blocs.

```
##  
## Attaching package: 'Matrix'  
  
## The following object is masked from 'package:tidyr':  
##  
##      expand
```



We estimate our model using a k-means initialization strategy, and set special priors for the blockmodel means that reflect the known DGP. All other hyper-parameters are left at their default values.

```
set.seed(831213)  
strt <- proc.time()  
net1.model <- mmsbm(formula.dyad = Y ~ 1,  
  senderID = "sender",  
  receiverID = "receiver",  
  nodeID = "node",  
  data.dyad = net1$dyad.data[train_index,],  
  n.blocks = net1$BLK,  
  directed = net1$DIRECTED,  
  mmsbm.control = list(mu_b = c(5,-5),  
    var_xi = 10,
```

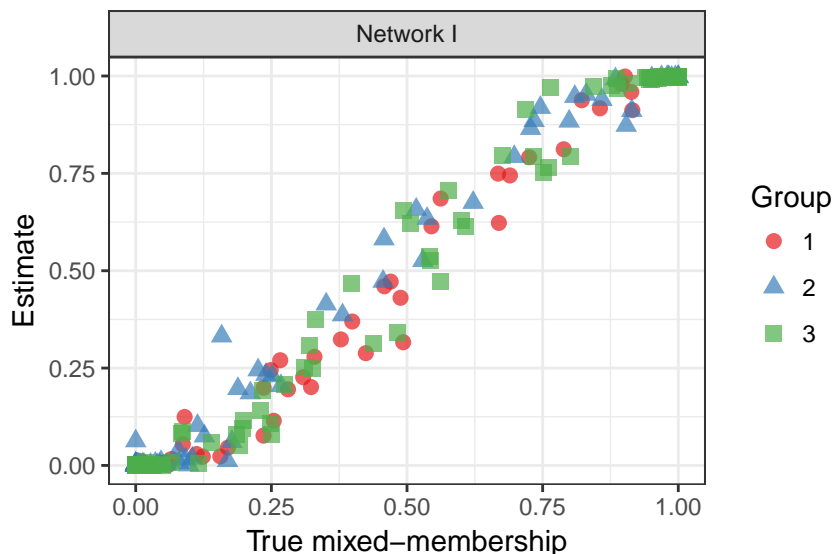
```

init = "kmeans"
))

our_time <- proc.time() - strt

```

After using a common implementation of the Hungarian algorithm to resolve the allocation problem raised by label switching, we compare known and estimated mixed-membership vectors for all 100 nodes in Figure 2 below. Alignment with the 1:1 diagonal indicates almost perfect retrieval of truth.



We can also compare our results to those of a fully Bayesian implementation of the MMSBM (viz. the one in package `lda`, which uses a fast collapsed Gibbs sampler to obtain posterior samples of both mixed-membership vectors and the blockmodel). Since our implementation allows for parallelization, we expect its performance to be better than that of a `lda`'s single-chain Gibbs sampler, which cannot be parallelized.

Whereas our implementation takes around 3.96 seconds to estimate the model's parameters using 4 parallel threads, taking 9000 samples from the model's collapsed posterior takes about 29.39 seconds on the same computer – over 7 times as long. Despite the speedup, both implementations produce virtually indistinguishable AUROC's on a held-out random sample of 1000 edges: 0.869 and 0.877, respectively.

Experiment 2: Static network with covariates

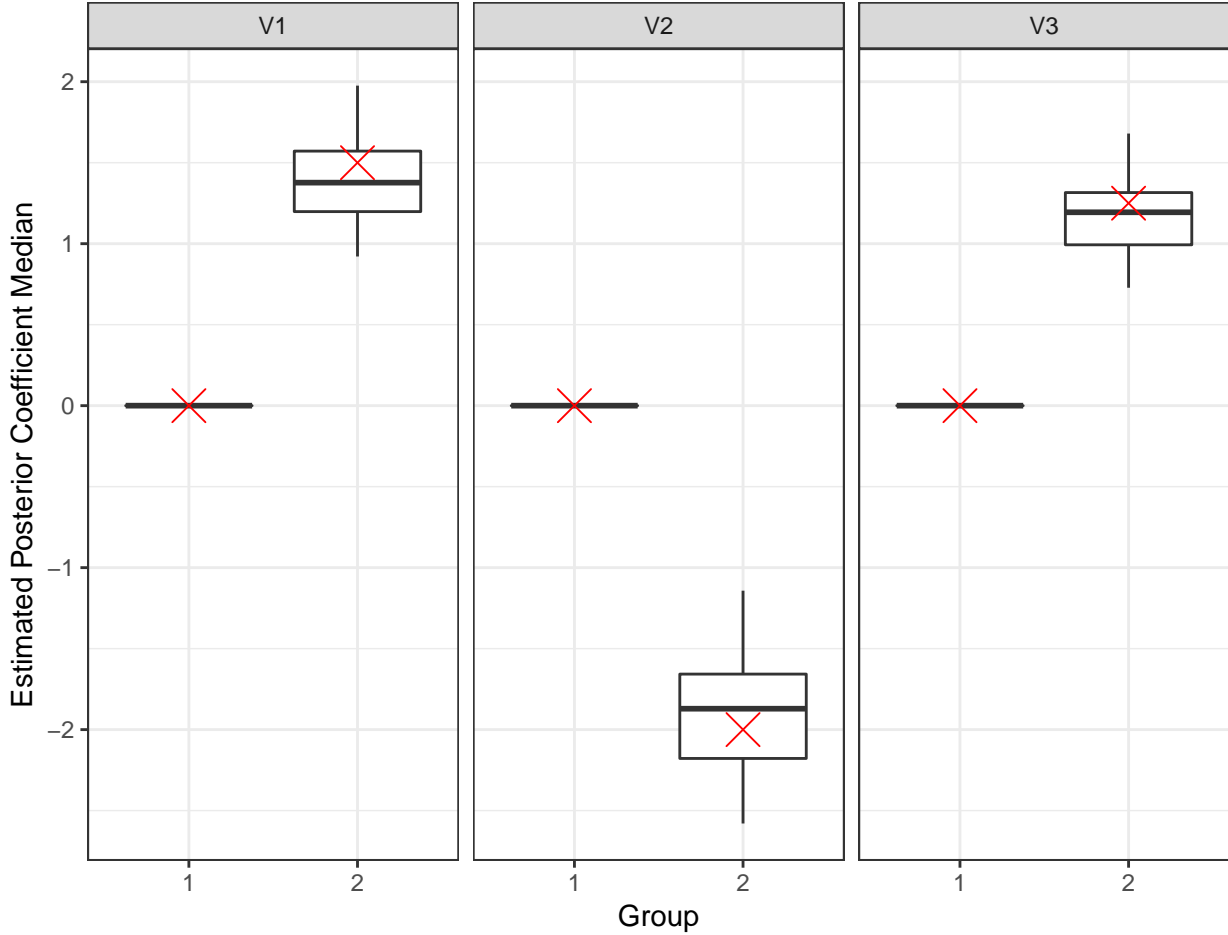
Our second set of simulations illustrates our model's ability to define a regression model for the mixed-membership vector. We generate 50 networks of 300 nodes and 3 latent blocs, generating mixed-membership vectors as a function of a hypothetical binary treatment assignment. Specifically, the treatment makes nodes much more likely to be part of bloc 2 and less likely to be part of either of the other two blocs. Figure 3 below shows a typical mixed-membership profile for these 50 networks, showing treated nodes as black circles and control nodes as red triangles.

Retrieve effect estimates

We first obtain effect estimates for each of the 50 networks, and plot estimates for the treatment effect for each group of using boxplots in the figure below. We also show the known value of the treatment effect. In all cases, the treatment effect is correctly identified, although our ridge penalty, introduced to avoid early divergence of coefficient estimates, results in attenuation bias.

Table 1: Comparison of median area under ROC curve for models estimated with and without monadic covariates, across 50 replications of the the same network.

| Model | AUROC |
|--------------------|-------|
| With covariates | 0.629 |
| Without covariates | 0.629 |



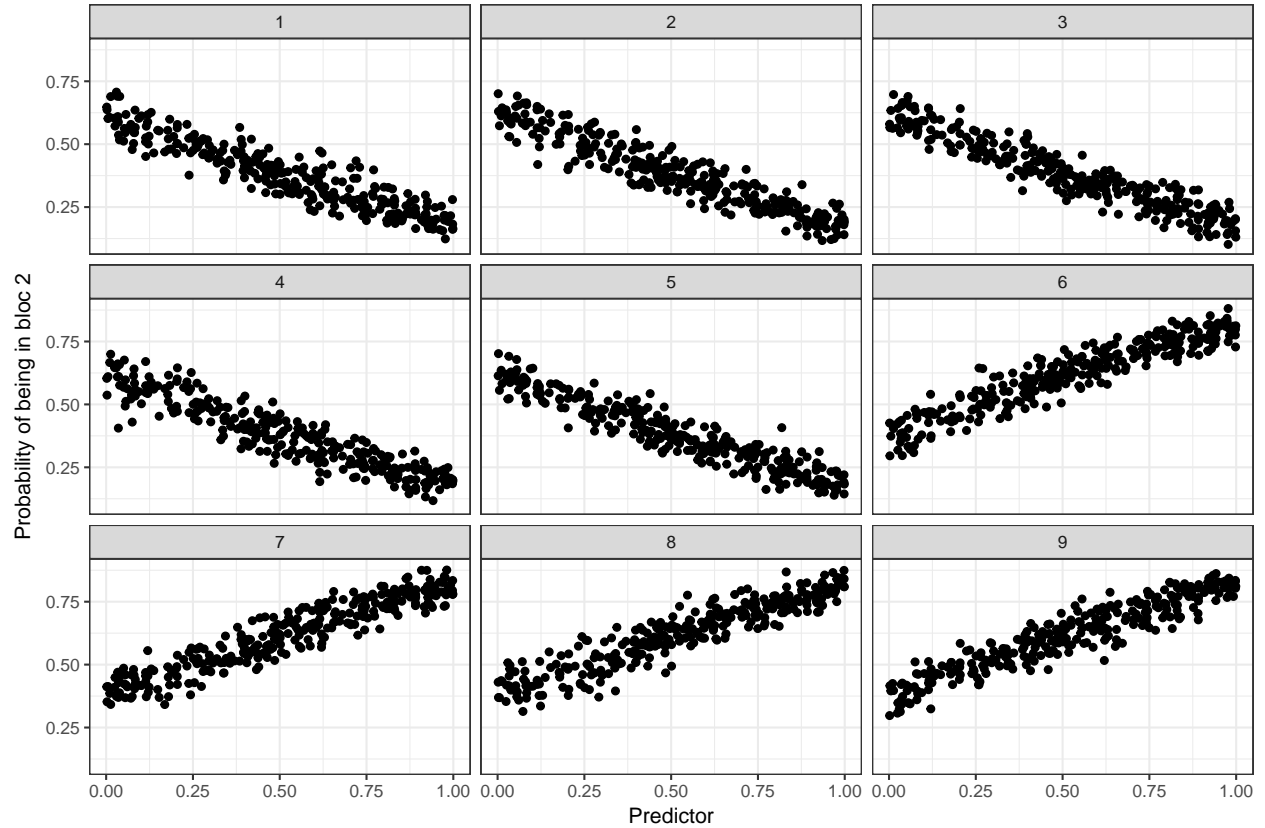
Correct classification

Next, we consider the predictive accuracy of the model with covariates. We compute the area under the ROC curve for a held-out set of 1000 edges. The accuracy measure is then compared to that of a model estimated on the same data but excluding information on the binary treatment variable. We find that, on average, predictive accuracy is higher for the model that incorporates the treatment covariate information.

Experiment 3: Dynamic network with covariates

For our third set of simulations, we illustrate the ability of our model to model network dynamics using a hidden Markov model for the mixed-membership vectors. We generate a network with 300 nodes, and let it evolve over 9 years according to a forward-moving, 2-state hidden Markov model. During the first stage

of this 9-year history, a continuous covariate is associated with a lower probability of being part of bloc 2 relative to bloc 1. In turn, the relationship is reversed during the second stage of history. Figure 4 shows the distribution of node-level probabilities of belonging to bloc 2 for each time period (a transition occurs from year 5 to year 6).



We compare our model's performance to that of a static model with an interaction between the focal covariate and year on two fronts: effect retrieval and accurate forecasting.

To show how such forecasting would work, we use the estimated HMM transition matrix and conditional distribution over states for the last observed year to generate predictions one year into the future, and compare those predictions to the held-out membership probabilities.