Testing sbmlogit package

Carter Allen

1/20/2020

Contents

To-Do

- Create plotting functions
- Assess the effect of degree correction (see P & C section 7.2.1)
- Fit models to more test data sets of differing community structure
- Look into model comparison criteria
- Look into model extensions
- Better understand interpretation of model parameters

Intro

Load packages. `igraph` is required, as well as the `C` library version.

```
library(igraph)
library(igraphdata)
library(sbmlogit)
library(sbmlhelpers)
library(ggraph)
library(tidygraph)
library(tidyverse)
library(coda)
```

Karate Data

Load karate data (an `igraph` object). Note that `sbmlogit` operates on data in the form of `igraph` objects.

```
data("karate")
```

Obtain MCMC samples for $K = 2$ specified clusters. We can specify a two cluster model by setting `alpha = 2`, in which case K is set automatically to 2, and α is set to $\alpha_{K\times 1} = (1/K, ..., 1/K)^T$. The number of MCMC iterations is controlled with `nsamples`.

```
fitK2 <- sbmlogit.mcmc(graph = karate,alpha = 2,nsamples = 2000)
```

Define the `mp` ("most probable"?) function from P & C (2016), where `apply(Sigma, 2, mp, K)` returns a $K \times N$ matrix, the transpose of which is the $N \times K$ matrix $\mathbf{P}$, where P_{ij} is the proportion of MCMC iterations where node i $(i = 1, ..., N)$ belonged to cluster j $(j = 1, ..., K)$.

```
# Function for estimator
mp = function(vec, K){
  v = rep(1:K)
  l = length(vec)

  for (i in 1:K){
    v[i] = sum(vec==i)/l
  }
  return(v)
}
```

Now, we apply the `which.max()` function to the matrix $\mathbf{P}$ described above to find the most probable cluster membership for each node. The `sbmlogit.remap()` function remaps the posterior estimate of $\boldsymbol{\sigma}$ to the canonical version described in P & C (2016).

Compute estimators.

```
SigmaK2 <- fitK2$sample # posterior samples
sigmaK2 <- apply(t(apply(SigmaK2, 2, mp, 2)), 1, which.max) # posterior estimator
scentroidK2 <- sbmlogit.remap(sigmaK2) # remapped posterior estimator
print(scentroidK2)
```

```
##  [1] 1 1 1 1 1 1 1 1 2 2 1 1 1 1 2 2 1 1 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2
```

Evaluate model fit with WAIC.

```
# function to compute WAIC
# verify this is correct
waic <- function(fit,burn = 0)
{
    S <- length(fit$lhood)
    ls <- exp(fit$lhood[(burn+1):S])
    w <- 2*(log(mean(ls)) - mean(log(ls)))
    return(w)
}
```

Compare models with varying K.

```
fitK3 <- sbmlogit.mcmc(graph = karate,alpha = 3,nsamples = 1000)
fitK4 <- sbmlogit.mcmc(graph = karate,alpha = 4,nsamples = 1000)
fitK5 <- sbmlogit.mcmc(graph = karate,alpha = 5,nsamples = 1000)
```

```
waic(fitK2, burn = 100)
```

```
## [1] 36.11339
```

```
waic(fitK3, burn = 100)
```

```
## [1] 53.68402
```

```
waic(fitK4, burn = 100)
```

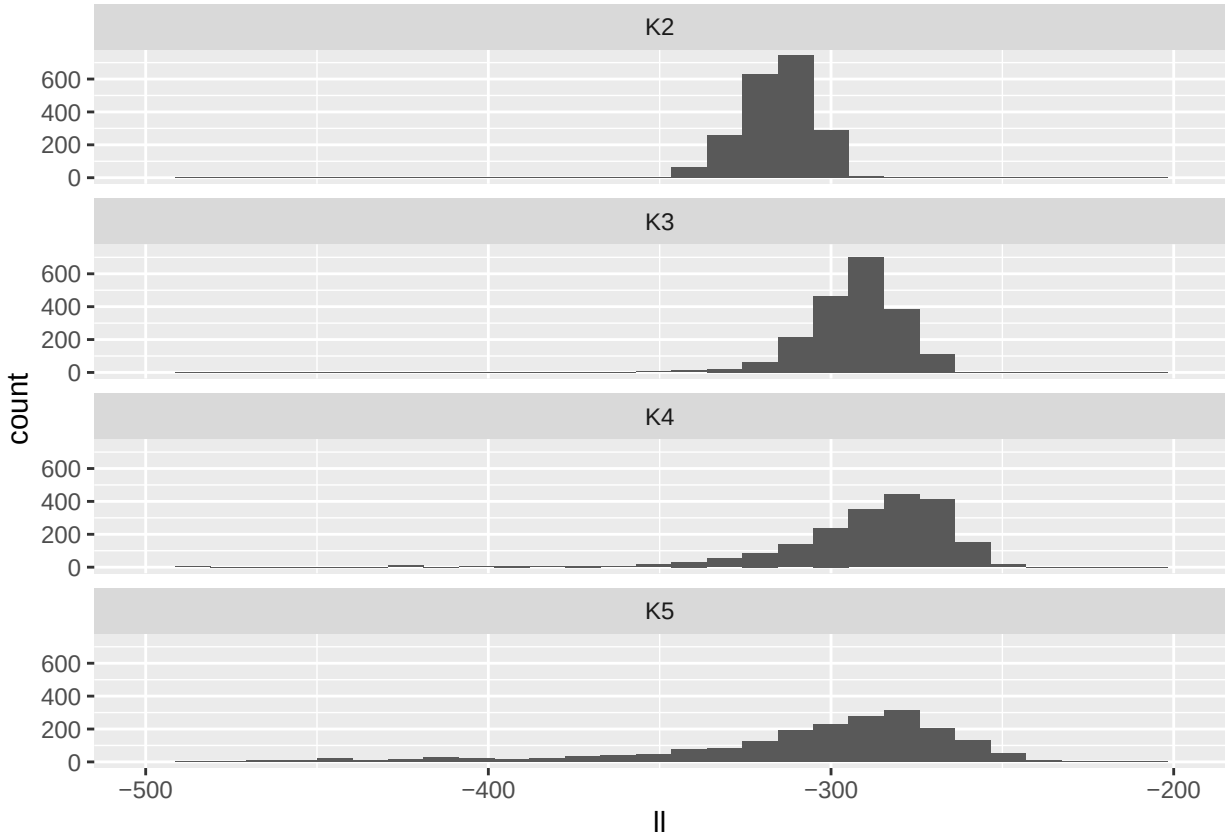
```
## [1] Inf
```

```
waic(fitK5, burn = 100)
```

```
## [1] Inf
```

Alternatively, we could inventigate posterior distributions of model log-likelihood. We note however that this approach does not properly penalize for model complexity.

```
lls_df <- as.data.frame(cbind(fitK2$lhood,fitK3$lhood,fitK4$lhood,fitK5$lhood))
colnames(lls_df) <- c("K2","K3","K4","K5")
lls_df <- lls_df %>%
  gather(key = "Model", value = "ll")
ggplot(data = lls_df,
       aes(x = ll)) +
  geom_histogram(bins = 30) +
  facet_wrap(~ Model, nrow = 4) +
  scale_x_continuous(limits = c(-500,-200))
```



Next, let's conduct posterior inference for 2-class model. In this case, `fitK2$gamma` contains the posterior samples of γ_{12}. In general, there are $\binom{K}{2}$ of the γ parameters: $\gamma_{12}, ..., \gamma_{K-1,K}$, since the authors set $\gamma_{11} = \gamma_{22} = ... = \gamma_{KK} = 0$ for identifiability purposes. Below is a summary of the posterior distribution of γ_{12} for the two cluster model.

```
mean_CRI(fitK2$gamma)
```

```
## [1] "-3.96 (-5.5, -2.86)"
```

We can see that the posterior mean and 95% credible interval for γ_{12} are less than 0 – the intra-community connectivity of the graph enforced by assuming $\gamma_{11} = \gamma_{22} = 0$. Thus, there is significantly less propensity for edges to exist *between* the two communities than *within* the two communities. This is indicative of strong assortative community structure.

The remaining parameters to infer are $\eta_1, ..., \eta_N$, where $N = 34$ is the number of nodes in the karate graph. Each η_i accounts for the expected degree of node i on the logit scale. The authors refer to this as *node correction* as well as the more common *degree correction*. Thus, $\text{logit}^{-1}(\eta_i)$ is the expected degree of node i.

```
expit <- function(l)
{
  return(exp(l)/(1+exp(l)))
}
```

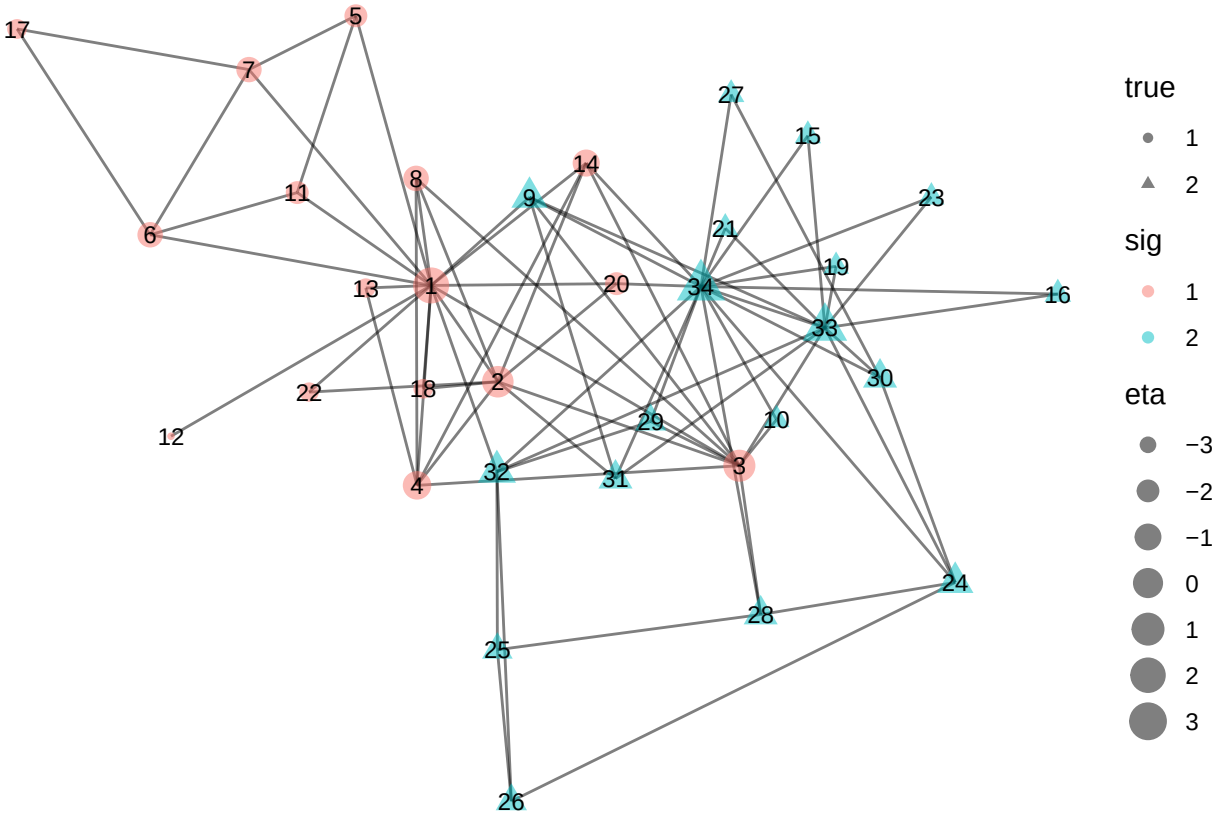
```
expit(colMeans(fitK2$eta))
```

```
##  [1] 0.95515258 0.76416645 0.80089788 0.52329408 0.17773659 0.29809708
##  [7] 0.29316696 0.29922520 0.40022617 0.09063045 0.17792511 0.03200290
## [13] 0.08929766 0.40905509 0.09689979 0.09966598 0.09793005 0.09469525
## [19] 0.08782846 0.17894335 0.09569812 0.09139508 0.08978369 0.41489586
## [25] 0.17370586 0.18968473 0.09638840 0.28989038 0.18866684 0.29587626
## [31] 0.29344243 0.51625086 0.85896004 0.95431478
```

We can see that the nodes with the highest expected degree are the actual "hubs" (i.e., the karate teachers), as the `karate` data are arranged such that the first and the last nodes are the two karate teachers.

To plot the estimated communities of the graph, we can use the custom function `plot_sbmlogit()`.

```
plot_sbmlogit(fitK2, ground = "color")
```



UK Faculty Network

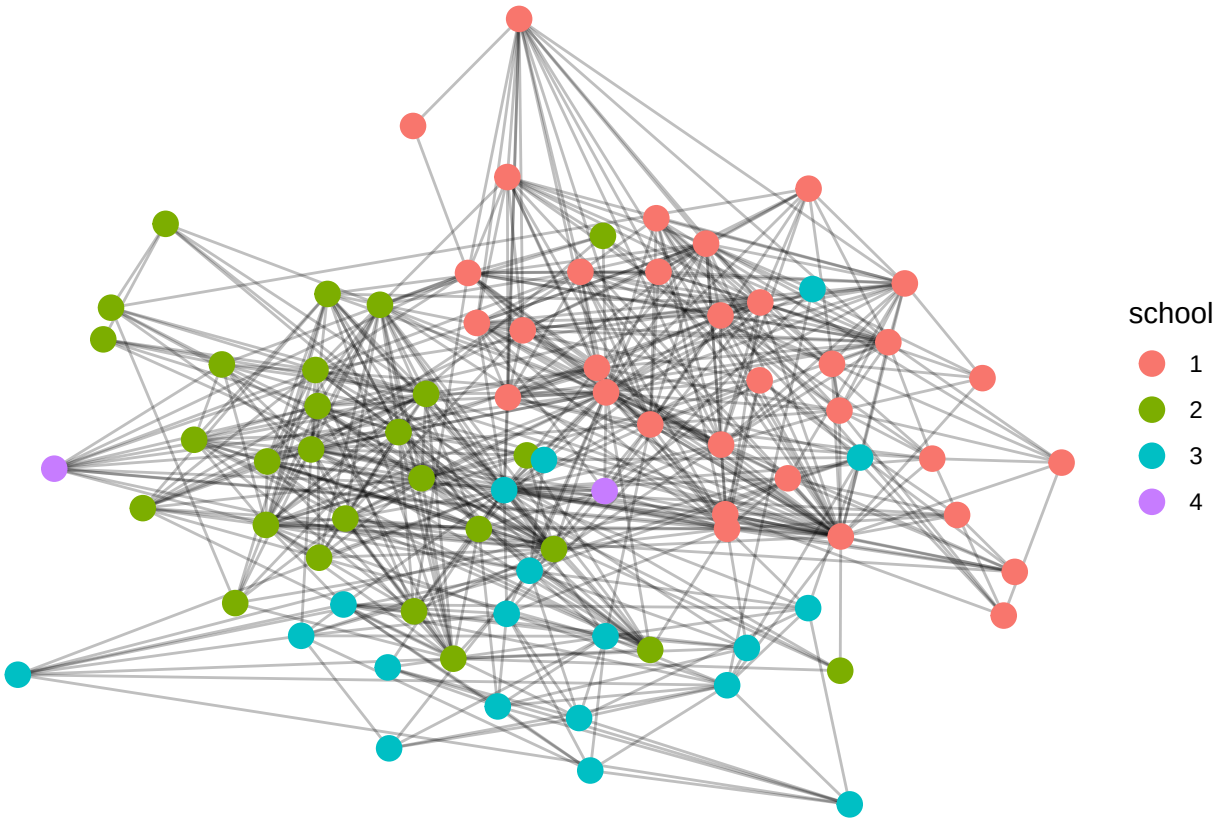
The UK faculty network data set is an `igraph` data set included in the `igraphdata` package. The network consists of friendships between 81 faculty at a university in the UK, with 817 directed and weighted edges. First, let's read in the `UKfaculty` data and save it as an undirected version, `UKfac`.

```
data("UKfaculty")
UKfac <- as.undirected(UKfaculty)
```

In the `UKfac` data, each individual's school is saved to the `Group` node attribute. We can use this information to obtain a view of the possible ground truth community memberships.

```
UKfac <- UKfac %>%
  as_tbl_graph() %>%
  activate(nodes) %>%
  mutate(school = as.factor(Group)) %>%
  as.igraph()
```

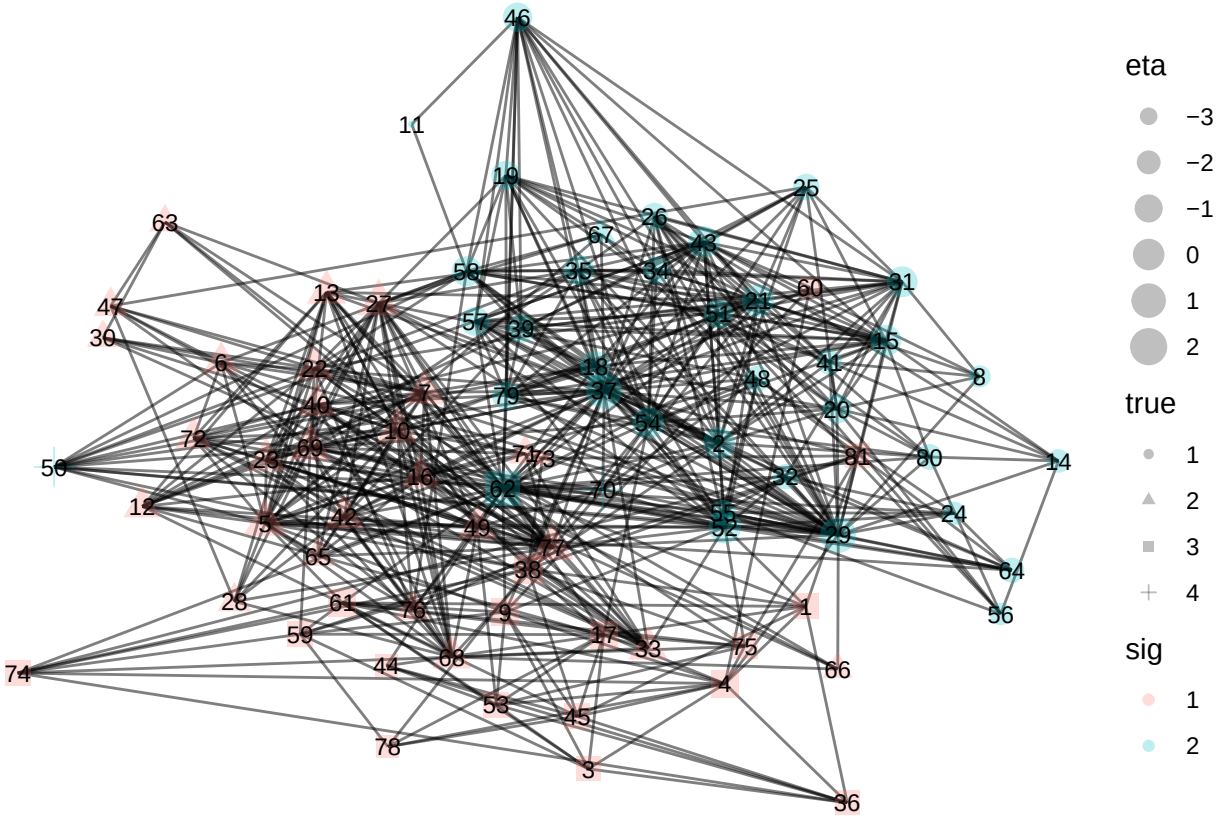
```
ggraph(UKfac,layout = "kk") +
  geom_edge_link(alpha = 0.25) +
  geom_node_point(size = 4, aes(color = school)) +
  theme_void()
```



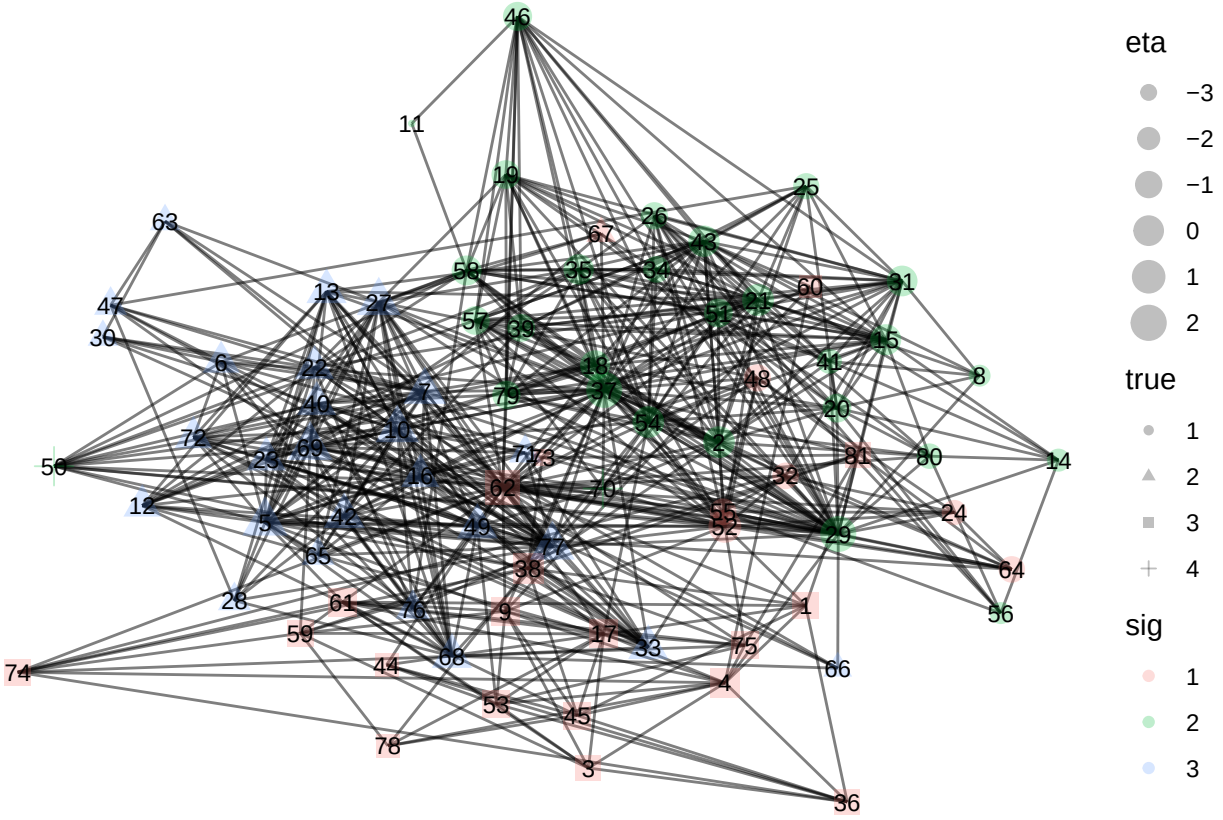
As before, we can fit SBM logit models with varying values of K and assess model fit using WAIC.

```
fitK2 <- sbmlogit.mcmc(UKfac,alpha = 2,nsamples = 5000)
fitK3 <- sbmlogit.mcmc(UKfac,alpha = 3,nsamples = 5000)
fitK4 <- sbmlogit.mcmc(UKfac,alpha = 4,nsamples = 5000)
fitK5 <- sbmlogit.mcmc(UKfac,alpha = 5,nsamples = 5000)
```

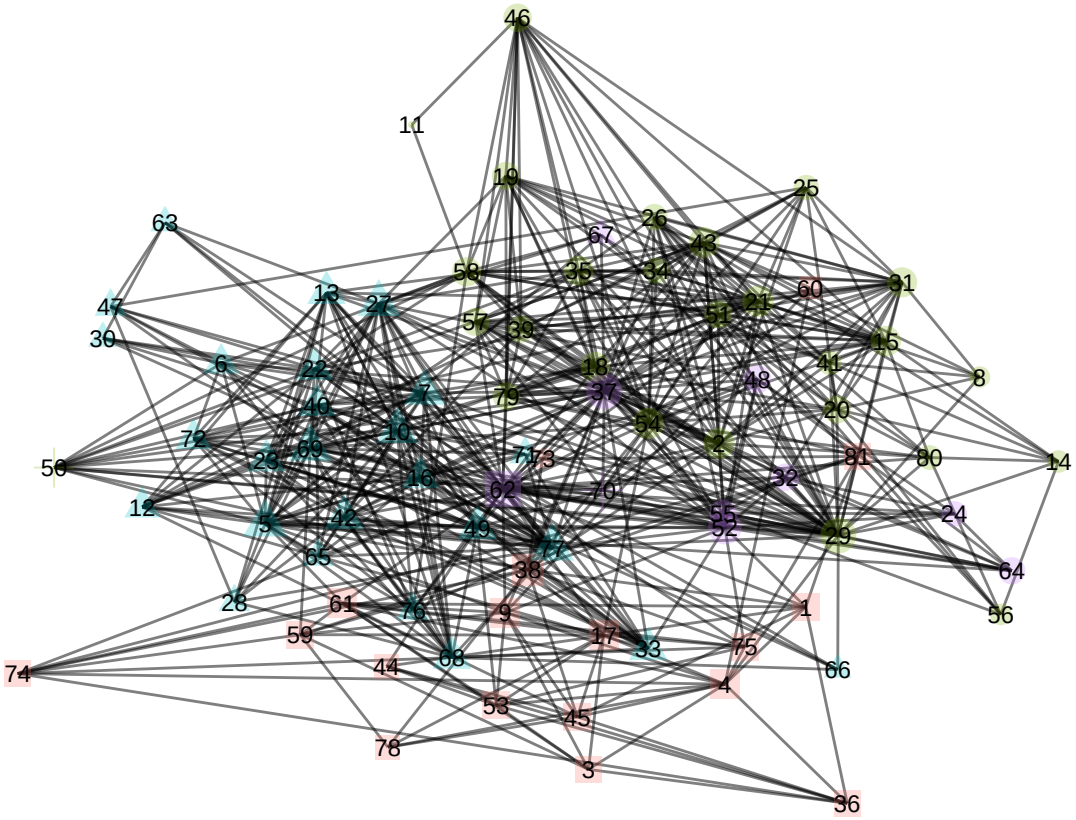
```
plot_sbmlogit(fitK2,ground = "school",alpha = 0.25)
```



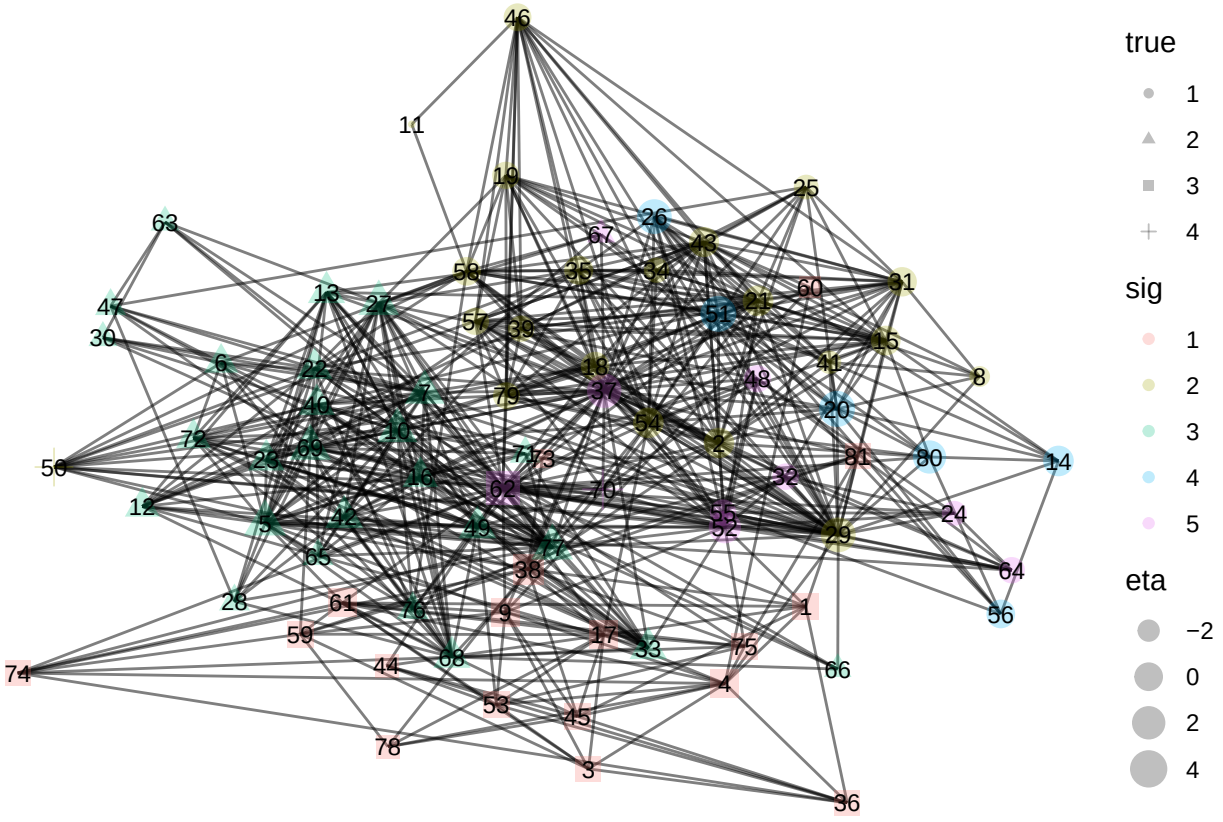

```
plot_sbmlogit(fitK3,ground = "school", alpha = 0.25)
```



```
plot_sbmlogit(fitK4,ground = "school", alpha = 0.25)
```




```
plot_sbmlogit(fitK5,ground = "school", alpha = 0.25)
```



For the 4 cluster model, we can assess the degree of community structure by investigating the γ terms.

```
apply(fitK4$gamma, 2, mean_CRI)
```

```
## [1] "-5.68 (-7.21, -4.58)" "-5.15 (-6.4, -4.27)"  "-4.27 (-5.85, -3.26)"
## [4] "-4.54 (-5.25, -3.96)" "-3.08 (-4.24, -2.34)" "-3.68 (-5.01, -2.89)"
```

It appears there is strong evidence of assortative community structure in the `UKfaculty` data since all inter-community γ parameters are negative and 95% credible intervals do not contain 0. We can make sense of the posterior community allotment by relating the inferred labels to the school membership of each faculty. To do so, we will obtain the inferred labeling from the `fitK4` model using the `get_labels()` function, and compare that to the school membership vector remapped to the canonical space.

```
sigmaK4 <- get_labels(fitK4)
schools <- UKfaculty %>%
  as_tbl_graph() %>%
  activate(nodes) %>%
  pull(Group) %>%
  sbmlogit.remap()
```

```
table(sigmaK4,schools)
```

```
##         schools
## sigmaK4  1  2  3  4
##       1 18  0  0  0
##       2  0 26  0  1
##       3  0  0 26  0
##       4  1  7  1  1
```

It appears that the inferred communities largely agree with the true school membership with the exception of

the faculty in school 4, of which there were only 2. The 4 cluster model placed only one of the faculty from school 4 into cluster 4, but it placed 9 faculty from the other schools into cluster 4. This suggests that maybe the 3 cluster model would be a better choice.

```
sigmaK3 <- get_labels(fitK3)
table(sigmaK3,schools)
```

```
##        schools
## sigmaK3  1  2  3  4
##       1 19  6  1  0
##       2  0 27  0  2
##       3  0  0 26  0
```

From the 3 cluster model, we can see that all faculty from school 1 are placed in cluster 1, with no other members in cluster 1. The model's cluster 2 is made up of the 33 members of school 2 as well as the two members of school 4. The model's cluster 3 is made up of entirely the faculty of school 3. We can further investigate the parameters of the three cluster model below.

```
apply(fitK3$gamma, 2, mean_CRI)
```

```
## [1] "-3.18 (-4.12, -2.68)" "-3.73 (-5.48, -3.1)"  "-3.82 (-4.42, -3.29)"
```

From the γ parameters above, we can see that there is strong assortative community structure in the three cluster model. We can assess the presence of hub nodes by investigating the η parameters.

```
colMeans(fitK3$eta) %>% sort()
```

```
##  [1] -3.422145410 -2.768940997 -1.944328558 -1.850299478 -1.839301326
##  [6] -1.801204060 -1.800914937 -1.688263384 -1.657871911 -1.600569070
## [11] -1.458721109 -1.448636108 -1.221163061 -1.214055872 -1.210175146
## [16] -1.198824474 -1.173434843 -1.051441217 -0.978120244 -0.971834240
## [21] -0.961474131 -0.958884270 -0.948836882 -0.793881229 -0.753810463
## [26] -0.668594399 -0.668510458 -0.643702365 -0.624885202 -0.581931176
## [31] -0.559110886 -0.548181377 -0.533814453 -0.522332856 -0.481830707
## [36] -0.479447664 -0.460727644 -0.456665645 -0.420084814 -0.305991140
## [41] -0.249595939 -0.234259813 -0.155885499 -0.082087696 -0.068478153
## [46] -0.054556791 -0.046120015 -0.025971388 -0.018145317 -0.015207590
## [51] -0.003936638  0.077557049  0.079674630  0.082266014  0.119104636
## [56]  0.216569047  0.367350408  0.388726290  0.388971527  0.400000369
## [61]  0.408638188  0.486256261  0.528342283  0.540110387  0.658911632
## [66]  0.662962871  0.771339578  0.896240426  1.006617277  1.011054726
## [71]  1.081499594  1.125007584  1.201622293  1.436542535  1.450627144
## [76]  1.668261088  1.675715102  1.790963122  2.210299590  2.729701653
## [81]  2.731222360
```

```
geweke.diag(fitK3$gamma[4001:5000,])
```

```
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##   var1   var2   var3
##  4.328  6.220 -6.188
```