Updates March 30

Owen G. Ward

3/30/2022

Initialization Procedure

We first confirm that the initialization procedure for the dense Poisson process works well.

Here we compare the final clustering performance when we use the initialisation scheme vs random initialization.

Here we fix the time ($T = 200$) and the initial period gives $n_0 = 30$ for $K = 2$ equally sized groups. Here this is a dense graph with each node having an edge to 75% of other nodes.

```
exp_12_files <- list.files(path = here("Experiments/exp_results/"),
                           pattern = "exp_12_")

exp_12_files[1:3] %>%
  ## to not care about n0 for now
  map_dfr(~readRDS(here("Experiments/exp_results/", .x))) %>%
  group_by(init, nodes) %>%
  select(-K,-model) |>
  summarise(mean_ARI = mean(ARI), sd_ARI = sd(ARI), med_ARI = median(ARI),
            num_sims = n())
```

```
## ‘summarise()‘ has grouped output by ’init’. You can override using the ‘.groups‘ argument.
```

```
## # A tibble: 6 x 6
## # Groups:   init [2]
##   init    nodes mean_ARI sd_ARI med_ARI num_sims
##   <chr>   <dbl>    <dbl>  <dbl>   <dbl>    <int>
## 1 Init      100    0.922 0.266        1       50
## 2 Init      200    0.975 0.143        1       50
## 3 Init      400    0.993 0.0321       1       50
## 4 No Init   100    0.9   0.303        1       50
## 5 No Init   200    0.86  0.351        1       50
## 6 No Init   400    0.840 0.370        1       50
```

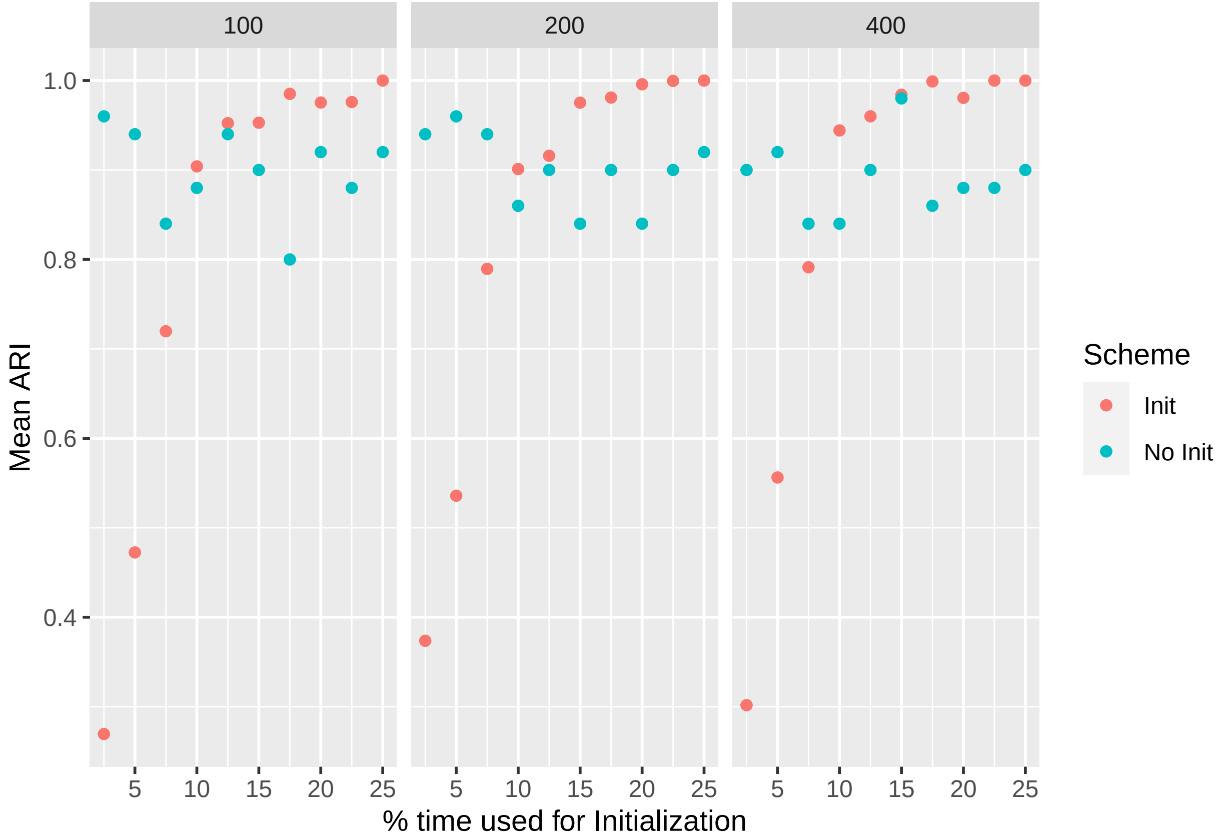
Similarly we can look at the impact of varying n_0, the period used for the initialization scheme, for the same simulation scenario.

Once a sufficient amount of time is used for the initialization scheme, it outperforms random initialization.

```
### init is definitely making the performance better, recover
### very well as the number of nodes increases also
```

```r
## varying n0
exp_12_files[4:6] %>%
  ## to not care about n0 for now
  map_dfr(~readRDS(here("Experiments/exp_results/", .x))) |>
  group_by(init, nodes, n0) |>
  summarise(mean_ari = mean(ARI)) |>
  mutate(n0 = n0/2) |>
  ggplot(aes(n0, mean_ari, colour = as.factor(init))) +
  geom_point() + facet_wrap(~nodes) +
  labs(x = "% time used for Initialization",
       y = "Mean ARI",
       colour = "Scheme")
```

```
## ‘summarise()‘ has grouped output by ’init’, ’nodes’. You can override using the ‘.groups‘ argument.
```



Experiment for Figure 1

I’ve also been trying to come up with a good experiment which shows the need to use the event times compared to binning the data. Here I’ve simulated data from a dense Poisson model with intensity functions of the form

$$\Lambda = \begin{pmatrix} \lambda_{11}(t) & \lambda_{12}(t) \\ \lambda_{21}(t) & \lambda_{22}(t) \end{pmatrix}$$

where we have

$$\lambda_{ij}(t) = a_1 \cdot 1_{0 \leq t < T/3} + a_2 \cdot 1_{T/3 \leq t < 2T/3} + a_3 \cdot 1_{2T/3 \leq T \leq T}$$

for sets of coefficients given by

$$\begin{pmatrix} \lambda_{11} \\ \lambda_{12} \\ \lambda_{21} \\ \lambda_{22} \end{pmatrix} = \begin{pmatrix} 0.25 & 0.5 & 2 \\ 0.75 & 1 & 0.33 \\ 2 & 0.25 & 0.5 \\ 0.33 & 0.75 & 1 \end{pmatrix}.$$

We repeatedly simulate data from this model, each time fitting our 4 models along with two competing methods. - We simply perform spectral clustering on the count matrix of the number of interactions. - We discretise this event data to construct a series of adjacency matrices and then apply the method of Pensky and Zhang (2019). To do this we need to choose how to discretise our event data, choosing a window size. For each window, we construct a corresponding adjacency matrix, using this series of matrices and apply the corresponding method of spectral clustering.

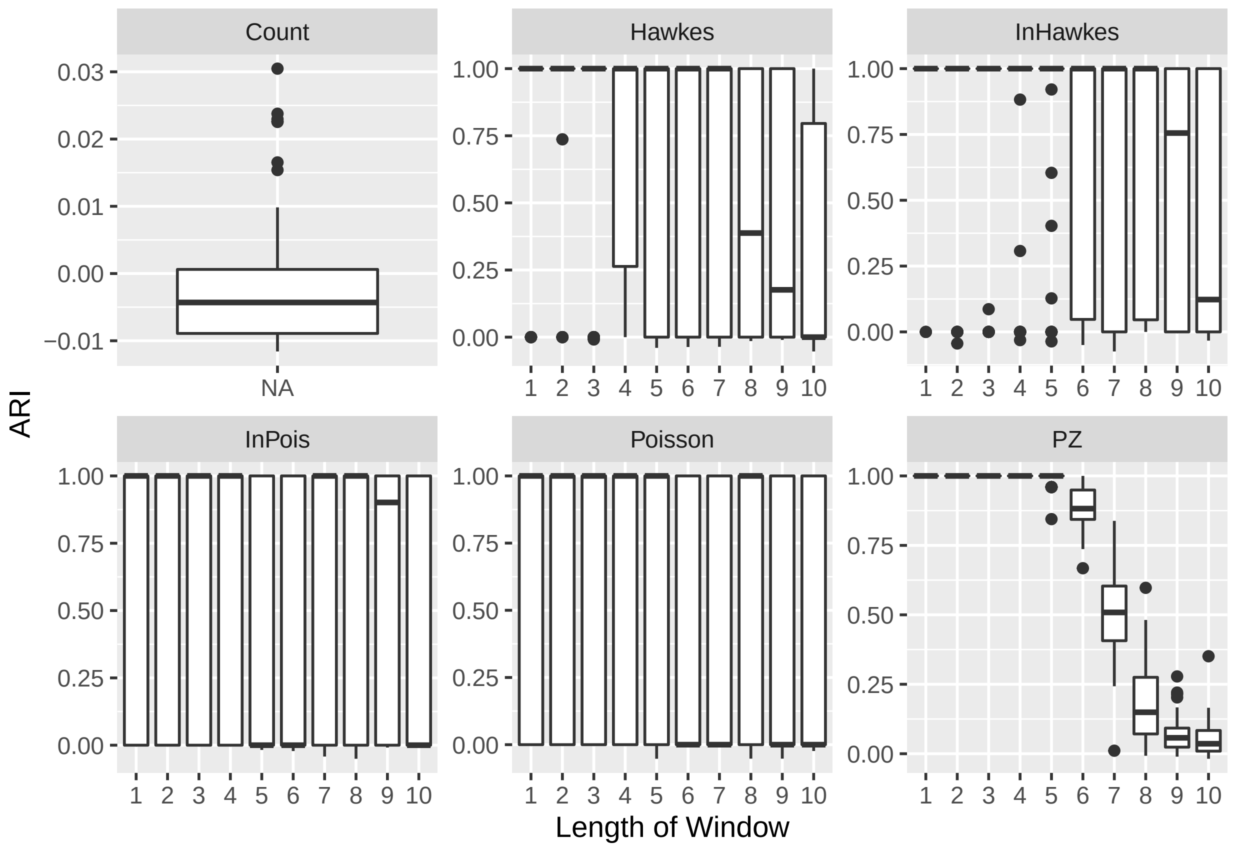
```
fig_1_files <- list.files(path = here("Experiments/exp_results/"),
                                    pattern = "fig_1_exp_1")

fig_1_files %>%
  map_dfr(~readRDS(here("Experiments/exp_results/", .x))) %>%
  group_by(Method) %>%
  summarise(mean(ARI), sd(ARI), median(ARI))
```

```
## # A tibble: 6 x 4
##   Method   ‘mean(ARI)‘ ‘sd(ARI)‘ ‘median(ARI)‘
##   <chr>          <dbl>     <dbl>         <dbl>
## 1 Count      -0.000931    0.0109      -0.00430
## 2 Hawkes      0.652       0.466        1
## 3 InHawkes    0.718       0.434        1
## 4 InPois      0.518       0.497        1
## 5 Poisson     0.537       0.499        1
## 6 PZ          0.669       0.406        0.960
```

Looking at the performance as we vary the window size, we see that the method of Pensky and Zhang can work well, but only in some scenarios. Note we have not used the initialization procedure for our methods here.

```
fig_1_files %>%
  map_dfr(~readRDS(here("Experiments/exp_results/", .x))) %>%
  group_by(Method, window_size) |>
  mutate(Method = as.factor(Method),
         window_size = as.factor(window_size)) |>
  ggplot(aes(window_size, ARI)) +
  geom_boxplot() +
  facet_wrap(~Method, scales = "free") +
  labs(x = "Length of Window")
```


Count
Hawkes
InHawkes
InPois
Poisson
PZ
ARI
Length of Window
0.03
0.02
0.01
0.00
−0.01
NA
1.00
0.75
0.50
0.25
0.00
1 2 3 4 5 6 7 8 9 10