CA analise

Cellular automaton data EDA

Authors: Chen Bistra, Yasmin Reich

Discription:

we created a Cellular automaton, that you can run through the file in the dist dir: “to_submit.exe”
in the git hub: https://github.com/chenbis/machine_learning_biology
To collect information on the automaton we ran the program 10 times on each set of parameters, as you can see in the file: “get_analize_csv.py” in the git.
the parameters we used:
p_values = [0.25, 0.5, 0.75, 1, 0.55, 0.6, 0.7]
l_values = [0, 5, 10, 20, 40, 80, 100]
the weights distribution:
right skew: s1= 0.1, s2= 0.1, s3= 0.3, s4= 0.5
left skew: s1= 0.5, s2= 0.3, s3= 0.1, s4= 0.1
Gaussian: s1= 0.1, s2= 0.4, s3= 0.4, s4= 0.1
uniform: s1= 0.25, s2= 0.25, s3= 0.25, s4=0.25

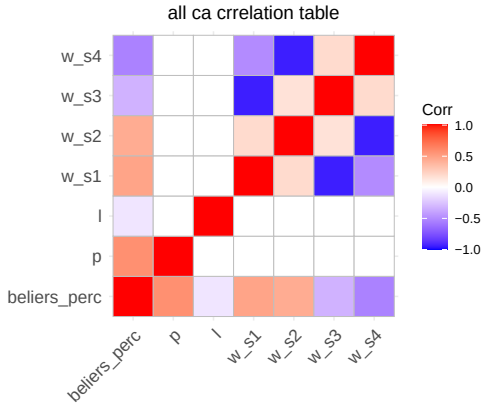
Initial cleaning of the data:

importing the data:

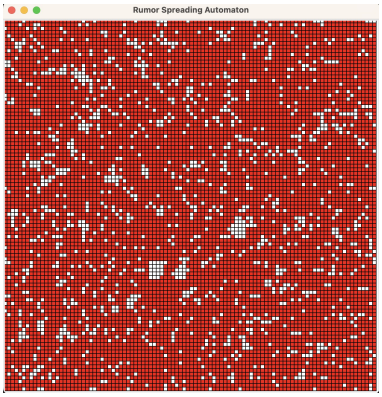
```
uniform_ca <-
  read.csv(file = '/Users/User/.vscode/comp_bio/machine_learning_biology/belief_data_uniform.csv',
                    header=T, na.strings=c("","NA"))
gaussian_ca <-
  read.csv(file = '/Users/User/.vscode/comp_bio/machine_learning_biology/belief_data_gaussian.csv',
                    header=T, na.strings=c("","NA"))
left_skew_ca <-
  read.csv(file = '/Users/User/.vscode/comp_bio/machine_learning_biology/belief_data_many_belivers.csv'
                    header=T, na.strings=c("","NA"))
right_skew_ca <-
  read.csv(file = '/Users/User/.vscode/comp_bio/machine_learning_biology/belief_data_many_non_belivers.
                    header=T, na.strings=c("","NA"))
# combine the data frames into one
all_ca <- rbind(uniform_ca, gaussian_ca, left_skew_ca, right_skew_ca)

# Calculate the mean belief ratio for each experiment
mean_belief_ratio <- summarize(all_ca, mean_beliers_perc = mean(beliers_perc))
```

first, we wanted to see the correlation between the different features of the data:
We expect to see s1 in a direct correlation to the percentage of believers and S4 in an inverse correlation.

```
library(randomForest)
library(ggcorrplot)
# ensure the results are repeatable
set.seed(7)
# calculate correlation matrix
correlationMatrix <- cor(all_ca)
# display the correlation matrix
ggcorrplot(correlationMatrix) + ggtitle("all ca crrelation table") +
  theme(plot.title = element_text(hjust = 0.5))
```

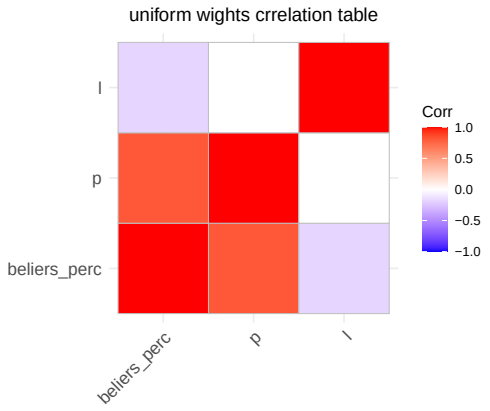


we can see that p is in a direct correlation to the percentage of believers and l is in a light inverse correlation. the l value is in inverse correlation because, if the l is low then the cells can spread the rumor several times, which creates the phenomenon of "waves of contagion". and if it is high then the cells stay believers but don't spread the rumor, Which creates a phenomenon of "vaccinated" (non believers) in the population of believers.
if l is lower: with the parameters of: p=1,l=20, gens=100, weights in uniform distribution. you can see the waves of spreading.
if l is high: with the parameters of: p=1,l=100, gens=100, weights in uniform distribution. you can see the non believers groups.



next, we wanted to see that the correlation is the same in all the different distribution groups. so when we make our conclusions we can consider all the data.
we ran this graph for each group: uniform_ca, Gaussian_ca, left_skew_ca, right_skew_ca, and it was the same.
for example, only the uniform data:

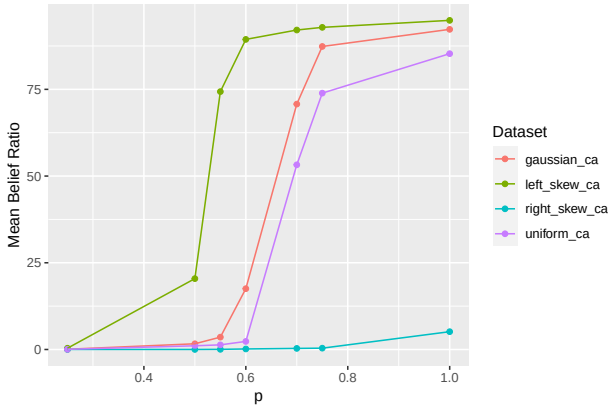
```r
df_select <- uniform_ca[, c("beliers_perc", "p", "l")] #the whights do not change
# ensure the results are repeatable
set.seed(7)
# calculate correlation matrix
correlationMatrix <- cor(df_select)
# display the correlation matrix
ggcorrplot(correlationMatrix) + ggtitle("uniform wights crrelation table") +
  theme(plot.title = element_text(hjust = 0.5))
```



first part results:

after that, to research witch parameters are the best for each distribution of the types of cells for medium spreading speed, we wanted to present how the change in p and l effect the spreading.
in this graph we presented the change in p, while l is 40 is all samples groups.

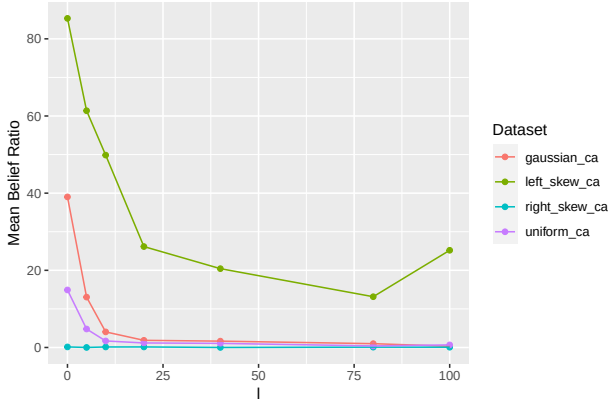
```r
library(ggplot2)
library(dplyr)
ll<-40
# Subset the cleaned_data data frame to only include rows where l = 20 for each dataset
uniform_ca_l40 <- subset(uniform_ca, l == ll) %>% mutate(dataset = "uniform_ca")
gaussian_ca_l40 <- subset(gaussian_ca, l == ll) %>% mutate(dataset = "gaussian_ca")
left_skew_ca_l40 <- subset(left_skew_ca, l == ll) %>% mutate(dataset = "left_skew_ca")
right_skew_ca_l40 <- subset(right_skew_ca, l == ll) %>% mutate(dataset = "right_skew_ca")

# Combine all four data frames into one
all_data <- bind_rows(uniform_ca_l40, gaussian_ca_l40, left_skew_ca_l40, right_skew_ca_l40)

# Group the data by p and dataset and calculate the mean of beliers_perc
mean_data <- all_data %>%
  group_by(p, dataset) %>%
  summarize(mean_beliers_perc = mean(beliers_perc))

# Create a scatter plot with a line for the mean beliers_perc for each p value and dataset
ggplot(mean_data, aes(x = p, y = mean_beliers_perc, color = dataset)) +
  geom_point() +
  geom_line() +
  labs(x = "p", y = "Mean Belief Ratio", color = "Dataset")
```



We conclude that for L=40, in a uniform division of the weights the appropriate P in order to maintain a reasonable rate of spread is 0.7.
parameters to run: p = 0.7, l = 40, s1 = 0.25, s2 = 0.25, s3 = 0.25, s4 = 0.25

in the same way we can set the p to be 0.5 and look at the l values:

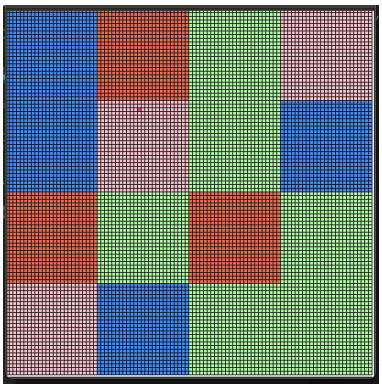
```
# Subset the cleaned_data data frame to only include rows where l = 20 for each dataset
pp <- 0.5
uniform_ca_p0.7 <- subset(uniform_ca, p == pp) %>% mutate(dataset = "uniform_ca")
gaussian_ca_p0.7 <- subset(gaussian_ca, p == pp) %>% mutate(dataset = "gaussian_ca")
left_skew_ca_p0.7 <- subset(left_skew_ca, p == pp) %>% mutate(dataset = "left_skew_ca")
right_skew_ca_p0.7 <- subset(right_skew_ca, p == pp) %>% mutate(dataset = "right_skew_ca")

# Combine all four data frames into one
all_data <- bind_rows(uniform_ca_p0.7, gaussian_ca_p0.7, left_skew_ca_p0.7,
                      right_skew_ca_p0.7)

# Group the data by p and dataset and calculate the mean of beliers_perc
mean_data <- all_data %>%
  group_by(l, dataset) %>%
  summarize(mean_beliers_perc = mean(beliers_perc))

# Create a scatter plot with a line for the mean beliers_perc for each p value and dataset
ggplot(mean_data, aes(x = l, y = mean_beliers_perc, color = dataset)) +
  geom_point() +
  geom_line() +
  labs(x = "l", y = "Mean Belief Ratio", color = "Dataset")
```



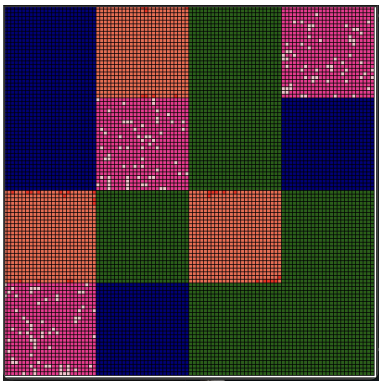
We conclude that for P=0.5, in a left skew (the most s1) of the weights the appropriate L in order to maintain a reasonable rate of spread is 10.
parameters to run: p=0.5, l=10, s1 = 0.5, s2 = 0.3, s3 = 0.1, s4 = 0.1
notice, because the model is not deterministic, the values will not be the best all the time. It depends on the arrangement of the cell types, and the selection of the starting distributor location.

second part results:

As for our approach in this section, we have opted to divide the grid into 25 by 25 blocks, and subsequently assign a specific population type to each block based on the initial distribution entered. To illustrate, if a uniform distribution is selected, each block will have an equal probability of being assigned any one of the four population types.
this is the grid at the begining of the run:



this is the grid at the end of the run:



When a uniform distribution is applied at the beginning of the simulation, we observe that S1 and S2 blocks did, in fact, spread the rumor quickly. S3 blocks displayed a degree of skepticism, but a majority of their citizens ultimately believed the rumor. In contrast, for S4 blocks, only those residing along the block's perimeter accepted the rumor, likely due to the fact that they had an increased opportunity to hear it from multiple sources.