201A: HW 3

Nicolas Nunez-Sahr

2023-11-27

1. Simulation of Markov Process

Assuming we start at State 1, we can carry out matrix multiplication on the left of a one-hot with a 1 on the first position and the transition probabilities matrix. The result is shown in the diagram below.

```r
library(ggplot2)
library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
P = matrix(data = c(c(0.2, 0.2, 0.2), c(0.7, 0.5, 0.4), c(0.1, 0.3, 0.4)), nrow = 3, ncol = 3)   # Trans
X_0 = as.vector(c(1, 0, 0))

X_1 = X_0 %*% P

X_1_df = data.frame(X_1 = X_1)
colnames(X_1_df) = c('1', '2', '3')
X_1_df = X_1_df %>% gather('State', 'Probability', '1', '2', '3')

ggplot(X_1_df, aes(x = State, y = Probability)) +
  geom_col() +
  labs(title = paste0('Probability of State at Turn ', 1, ' starting at ', match(1, X_0))) +
  theme(plot.title = element_text(hjust=0.5))
```

Probability of State at Turn 1 starting at 1
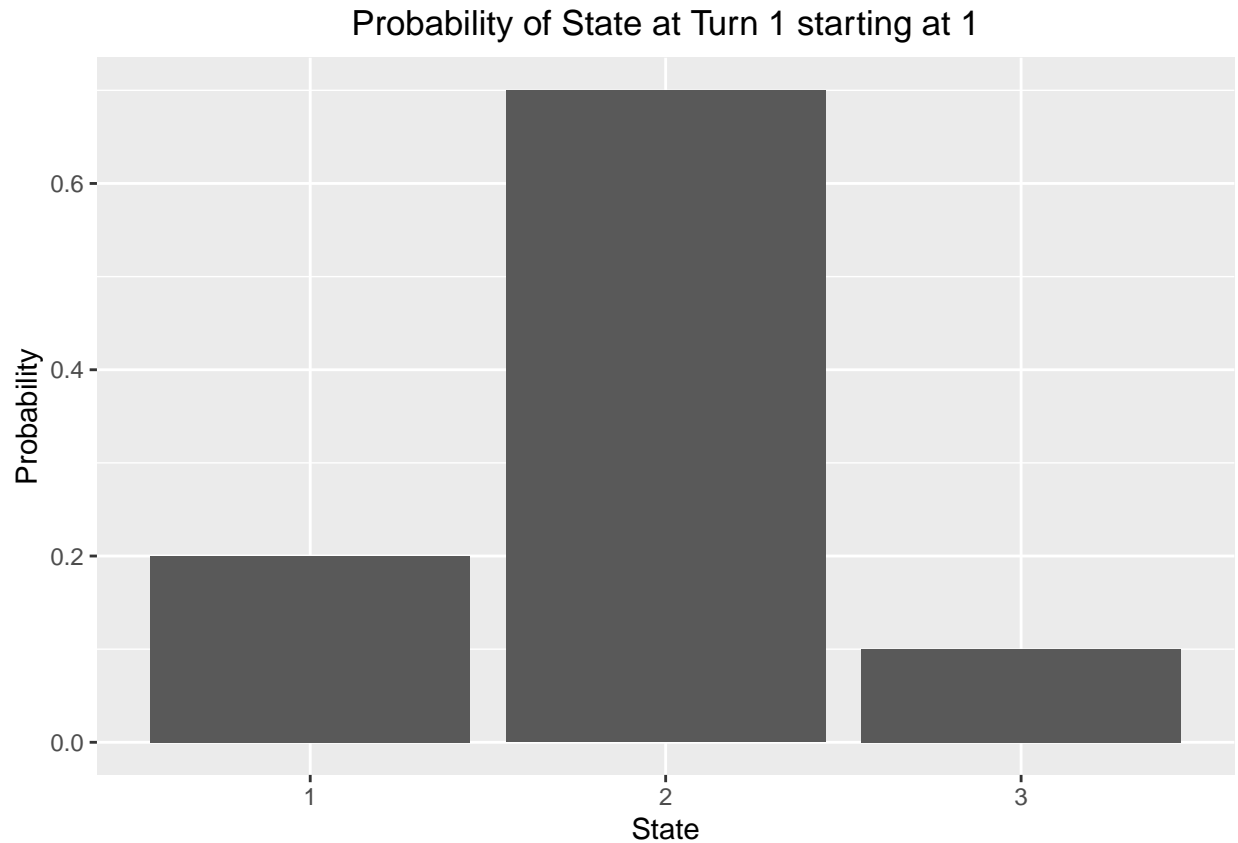
To move inside the Markov chain, we can sample from the rows of P: we pick a row according to where we are at, and get a value of 1, 2, or 3 according to those probabilities.

```
X_1 =  sample(x = c(1, 2, 3), size = 1, prob = as.vector(c(1, 0, 0)) %*% P)
print(paste0('Starting at State 1, after 1 iteration we are now at State ', X_1, '.'))
```

```
## [1] "Starting at State 1, after 1 iteration we are now at State 3."
```

```
X_1_vec = c(rep(0, X_1 - 1), 1, rep(0, 3 - X_1))
```

2. Stationary Distribution

```
# Uniform
X_0 = c(1/3, 1/3, 1/3)

num_iterations = 10

position_probabilities = c(X_0)
for (t in 1:num_iterations) {
  if (t == 1) {
    X_t = X_0 %*% P
  } else {
    X_t = X_t %*% P
```

2

```
  }
  position_probabilities = append(position_probabilities, X_t)
}
position_probabilities_matrix = t(matrix(data = position_probabilities, nrow = 3, ncol = num_iterations
position_probabilities_df = as.data.frame(position_probabilities_matrix)
colnames(position_probabilities_df) = c('1', '2', '3')
position_probabilities_df$iteration = 0:num_iterations
position_probabilities_df = position_probabilities_df %>% gather('State', 'Probability', '1', '2', '3')

ggplot(position_probabilities_df, aes(x=iteration, y=Probability, group=State, color=State)) +
  geom_line(linewidth=2) +
  geom_point(size=2) +
  scale_y_continuous(limits = c(0, 1)) +
  theme_minimal() +
  scale_x_continuous(name = 'Iteration', breaks = seq(0, 10)) +
  theme(plot.title = element_text(hjust=0.5)) +
  labs(title = 'From Initial Distribution to Limiting Distribution')
```


From Initial Distribution to Limiting Distribution
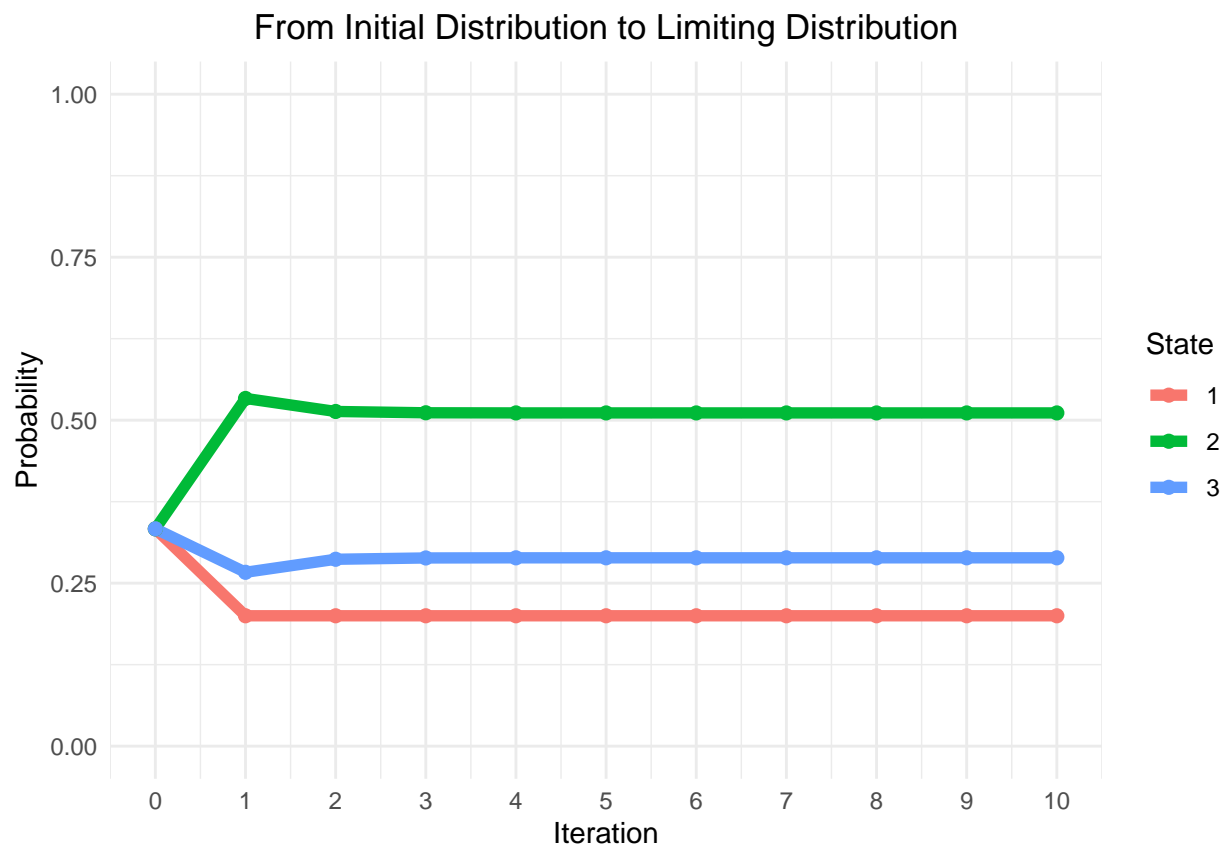
The initial distribution was set to the uniform distribution: since there's only 3 states, each was assigned a probability of one third. As can be observed in the chart above, the probability of being in each state quickly converged to the limiting distribution, which is 0.200 for State 1, 0.511 for State 2, and 0.289 for State 3. If we pick different starting conditions, it may take a few more iterations for the distribution to converge, but it always converges quickly, within 5 iterations. The limiting distribution is the stationary distribution if the graph is irreducible and aperiodic. Our graph is both of those things since it is fully connected, and no transition probabilities equal 1 or 0. Hence, the distribution described above is also the stationary distribution; furthermore, it is unique because the graph is irreducible and there is a finite

3

number of states.

```r
limiting_distribution = position_probabilities_matrix[11,]
difference_from_limiting_distribution = sweep(x = position_probabilities_matrix, MARGIN = 2, STATS = pos
compute_norm = function(x) {
  return (x[1]^2 + x[2]^2 + x[3]^2)
}
norm_of_difference = apply(difference_from_limiting_distribution, 1, compute_norm)

norm_df = data.frame(norm = norm_of_difference, iteration = 0:num_iterations)

ggplot(norm_df, aes(x=iteration, y = log(norm))) +
  geom_line(size=2) +
  geom_point(size=2) +
  theme_minimal() +
  scale_x_continuous(name = 'Iteration', breaks = seq(0, 10)) +
  scale_y_continuous(name = 'Log of Norm', limits = c(-60, 0)) +
  theme(plot.title = element_text(hjust=0.5)) +
  labs(title = 'Log of Squared L-2 Norm of Difference of Initial and Limiting Distributions')
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```


Log of Squared L−2 Norm of Difference of Initial and Limiting Distributions
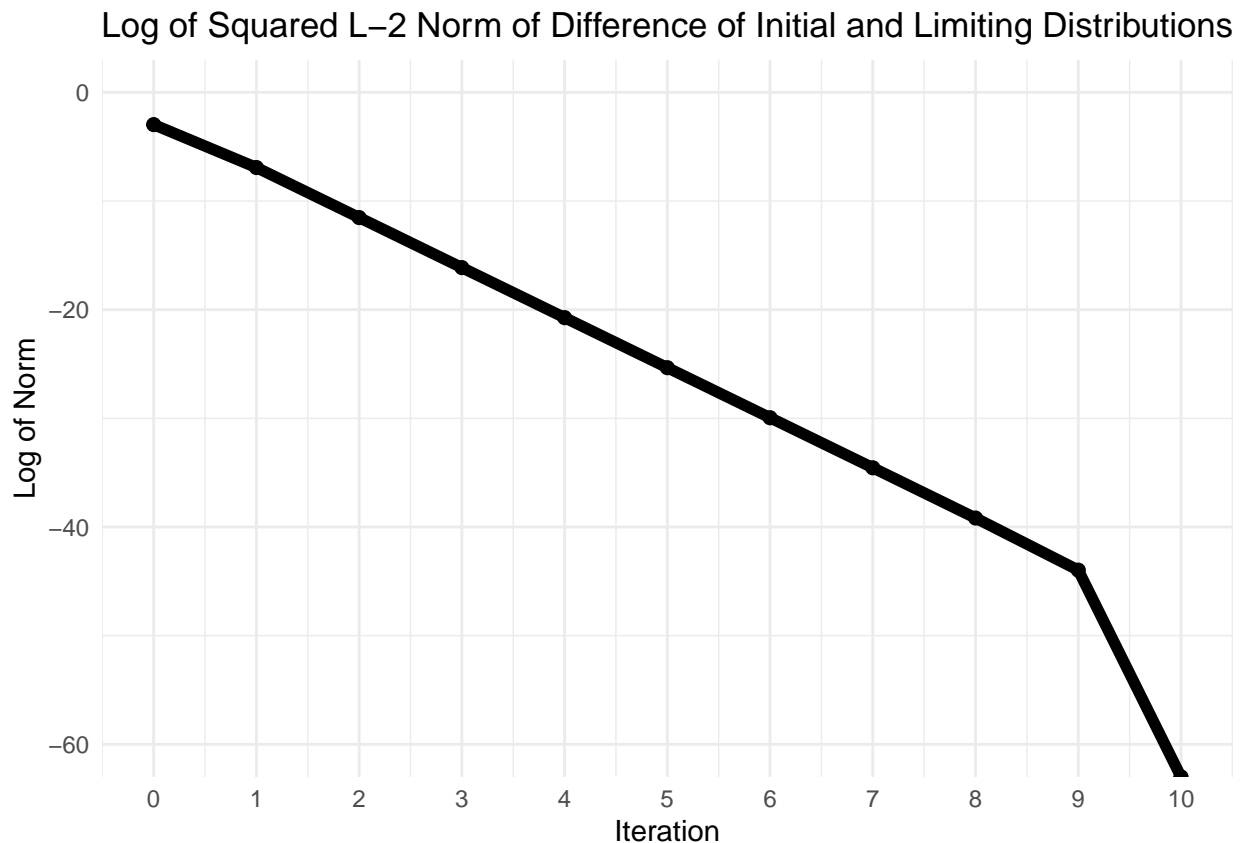
As can be observed in the chart above, the squared distance to the stationary distribution decreases toward 0 as the iterations increase.

3. Absorbing State

```r
num_steps = 100
num_iterations = 1000
s_list_of_lists = c()
for (starting_point in 1:2) {
  s_list = c()
  for (t in 1:num_iterations) {
    X_t = starting_point
    X_t_vec = c(rep(0, X_t - 1), 1, rep(0, 3 - X_t))
    for (s in 1:num_steps) {
      X_t_plus_1 = sample(x = c(1, 2, 3), size = 1, prob = as.vector(X_t_vec) %*% P)
      X_t_plus_one_vec = c(rep(0, X_t_plus_1 - 1), 1, rep(0, 3 - X_t_plus_1))
      if (X_t_plus_1 == 3) {
        s_list = append(s_list, s)
        break
      }
      X_t = X_t_plus_1
      X_t_vec = X_t_plus_one_vec
    }
  }
  s_list_of_lists = append(s_list_of_lists, s_list)
}

steps_df = data.frame('state1' = s_list_of_lists[1:1000], 'state2' = s_list_of_lists[1001:2000])
steps_df_g = steps_df %>% gather('starting_state', 'steps', 'state1', 'state2')
steps_df_g_counts = steps_df_g %>%
  group_by(starting_state, steps) %>%
  summarise(count = n())
```

```
## 'summarise()' has grouped output by 'starting_state'. You can override using
## the '.groups' argument.
```

```r
ggplot(steps_df_g_counts, aes(x=steps, y = count, group=starting_state, fill=starting_state)) +
  facet_wrap(~starting_state, nrow = 1) +
  geom_col() +
  labs(title = 'Number of Steps to get to Absorbing State 3') +
  theme_minimal() +
  theme(plot.title = element_text(hjust=0.5))
```

Number of Steps to get to Absorbing State 3
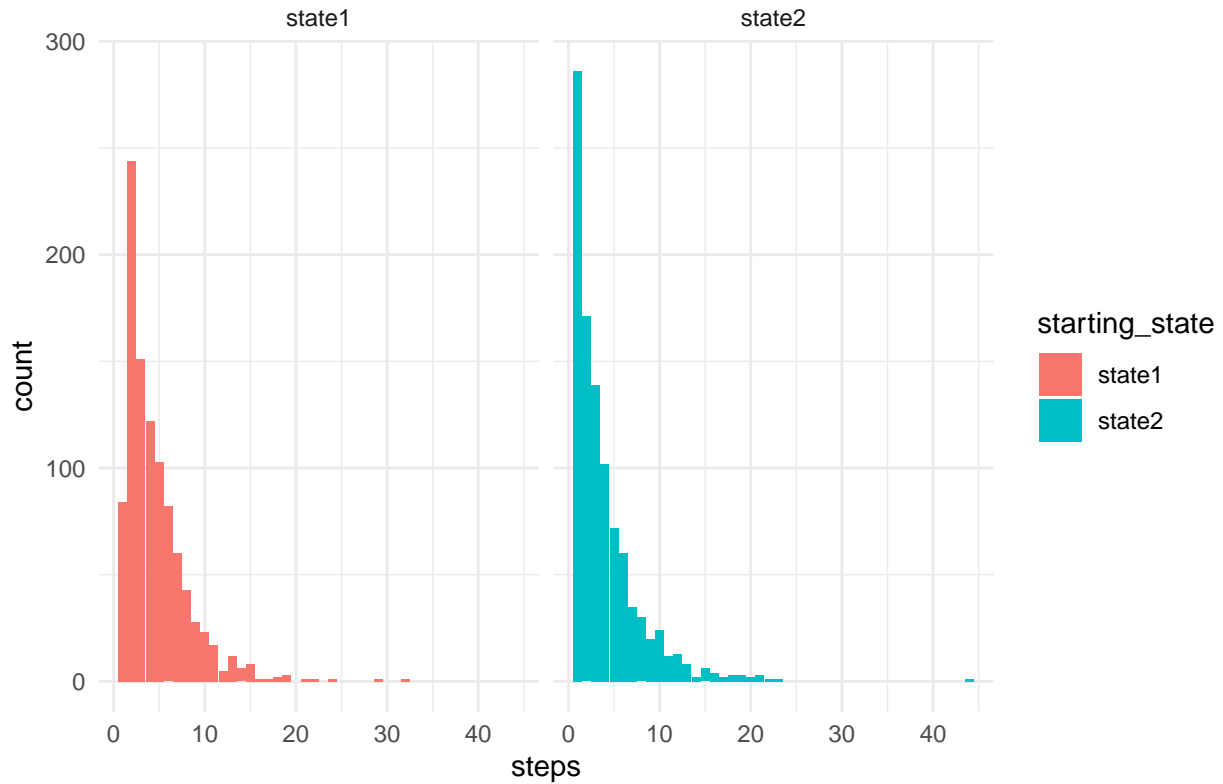
As can be observed in the chart above, it takes longer to get to State 3 from State 1 than from State 2. This makes sense, since the probability of transitioning to 3 from 1 is 0.1, while the probability of transitioning from 2 to 3 is 0.3. If we transition from State 1 to 2, then it would be the same as Starting from 2 but incurring a cost of 1 step. In fact, the distribution of State 1 starting from 1 step looks similar to the distribution of State 2, although we are not taking into account the possibility of staying in State 1. This is a discrete distribution, and therefore it can be modeled using a Poisson distribution.