Computational Eigenvector Simulation

Ali Taqi

Example 1: A Symmetric Stochastic Matrix

```r
#################################
### Step 0: Setup the matrix ###
#################################

# Set seed
set.seed(23)
# Set parameters
M <- 2
# Generate matrix
P <- RM_stoch(M, symm = F, sparsity = F)
if(bool_loud){P}

#################################
#### Step 1: Get the batch ####
#################################

# Set batch parameters
B <- 100
# Create batch
batch <- make_batch(M = M, B = B)
if(bool_loud){head(batch)}

###################################
#### Step 2: Evolve the batch ####
###################################

# Set evolution parameters
steps <- 10
# Evolve and index batch
evolved_batch <- evolve_batch(batch, steps, with_steps = T)
# Index the batch
evolved_batch <- indexed_batch(evolved_batch, steps)
if(bool_loud){head(evolved_batch)}
```

```r
#####################################
#### Step 3: Animate the batch! ####
#####################################

# Plot the evolution arrays of the batch elements
batch_data1 <- evolved_batch
batch_scatterplot1 <- batch_2d_plot(batch_data1)
```

```r
# Add transition time
batch_animation1 <- batch_scatterplot1 + transition_time(time = time)

# Set me to true!
bool_animate <- F
if(bool_animate){batch_animation1}
```

Example 2: Normal Matrix

```r
################################
### Step 0: Setup the matrix ###
################################

# Set seed
set.seed(23)
# Set parameters
M <- 3
mu <- 0
sd <- 1
normal_args <- c(mu, sd)
# Generate matrix
P <- RM_normal(M, normal_args)
if(bool_loud){P}

################################
#### Step 1: Get the batch ####
################################

# Set batch parameters
B <- 100
# Create batch
batch <- make_batch(M = M, B = B)
if(bool_loud){head(batch)}

##################################
#### Step 2: Evolve the batch ####
##################################

# Set evolution parameters
steps <- 10
# Evolve batch
evolved_batch <- evolve_batch(batch, steps, with_steps = T)
# Index the batch
evolved_batch <- indexed_batch(evolved_batch, steps)
```

```r
####################################
#### Step 3: Analyze the batch! ####
####################################

# Get the final evolved batch elements after all steps
fully_evolved <- time_array(evolved_batch, at_time = steps)
fully_evolved
```

```
##              x1           x2          x3 time element_index
## 1    63.920140   67.6506428   76.789305   10             1
## 2   -57.775270  -62.9011770  -69.894827   10             2
## 3   128.524380  137.2751532  157.948287   10             3
## 4   -59.852591  -65.8243321  -74.597995   10             4
## 5  -102.002107 -108.7784419 -125.480776   10             5
## 6     0.529376   -0.4838825    1.287504   10             6
## 7   201.452976  214.7777913  247.778936   10             7
## 8     8.839419   11.5796913   11.829641   10             8
## 9    95.912372  102.0187870  118.377880   10             9
## 10 -132.342996 -141.8125487 -163.183550   10            10
## 11   72.616876   78.1618706   89.693394   10            11
## 12    5.604876    4.2937003    5.855973   10            12
## 13 -182.219177 -194.9923297 -224.749634   10            13
## 14  101.351093  106.5387413  122.726988   10            14
## 15  -27.449574  -29.3328108  -32.879648   10            15
## 16 -135.938655 -145.2257381 -166.127292   10            16
## 17  115.522597  124.7574592  142.918233   10            17
## 18   82.951312   87.2078752   99.999519   10            18
## 19  -62.767304  -66.2024001  -78.479166   10            19
## 20   37.734551   41.9271764   47.057706   10            20
## 21  195.959048  208.6793444  239.907833   10            21
## 22  176.719843  188.7296259  218.065393   10            22
## 23 -175.641817 -187.4807638 -216.491661   10            23
## 24 -151.722792 -162.3693516 -187.545343   10            24
## 25  -97.688496 -104.3638596 -118.103587   10            25
## 26  -31.769689  -33.4864338  -39.227301   10            26
## 27   -3.157744   -3.2750441   -4.593282   10            27
## 28  111.278150  119.8830293  137.638102   10            28
## 29 -197.084254 -209.3770043 -241.321861   10            29
## 30  133.259477  141.5147321  162.434843   10            30
## 31 -100.102773 -106.6676259 -124.135389   10            31
## 32  -58.435617  -62.8852491  -71.815528   10            32
## 33  120.504380  127.7831589  147.385084   10            33
## 34  120.450266  128.8762339  148.120160   10            34
## 35   99.376255  106.8077224  122.312822   10            35
## 36  -12.409839  -13.8377263  -14.395010   10            36
## 37   36.267668   37.0065868   44.542539   10            37
## 38 -156.524374 -166.5459759 -190.333528   10            38
## 39  -40.085587  -40.9237033  -48.077058   10            39
## 40  140.108688  149.3618086  173.263972   10            40
## 41  -30.924940  -33.7459903  -36.433392   10            41
## 42  -42.621062  -45.8697833  -52.071126   10            42
## 43  -82.583125  -89.9273490 -100.885736   10            43
## 44  -65.372814  -68.5027526  -79.594548   10            44
## 45  121.590749  128.6038091  148.532601   10            45
## 46 -106.912149 -115.9300813 -132.141893   10            46
## 47   30.772681   34.2887259   39.109906   10            47
## 48   -4.658729   -5.1402773   -6.928459   10            48
## 49   45.715698   46.8205215   55.347878   10            49
## 50  125.368392  132.6029772  152.471973   10            50
## 51 -111.311111 -117.7342719 -137.264297   10            51
## 52  -21.601287  -24.1942763  -26.741200   10            52
## 53 -109.265509 -117.7420023 -133.043999   10            53
## 54    23.553624   24.5477574   29.748601   10            54
## 55   202.620010  215.2627114  246.854022   10            55
## 56   124.946891  132.4981161  151.578376   10            56
## 57  -135.231300 -144.3666392 -165.691036   10            57
## 58   -67.341432  -71.3669481  -82.035024   10            58
## 59   -22.557432  -26.1465849  -28.967691   10            59
## 60   179.326392  189.8437195  218.263167   10            60
## 61    79.737292   84.6806266   96.981281   10            61
## 62   184.814516  196.2839872  226.401691   10            62
## 63    61.653823   66.6640213   77.219333   10            63
## 64  -109.897363 -117.5878754 -135.119125   10            64
## 65   158.464824  169.2350070  194.757824   10            65
## 66    74.586061   81.3425066   92.841670   10            66
## 67    54.788112   59.9819465   68.806022   10            67
## 68    39.899195   41.3195597   47.439698   10            68
## 69   -28.687658  -33.1360843  -36.445517   10            69
## 70    29.601362   32.6770767   35.910421   10            70
## 71   -82.643576  -88.3518573 -101.811172   10            71
## 72    15.766338   17.7258428   18.478284   10            72
## 73    92.399640   99.2167233  114.363027   10            73
## 74   -99.896869 -105.0860741 -121.380012   10            74
## 75    89.388896   95.6562163  110.973514   10            75
## 76   -26.954269  -28.4480370  -33.199741   10            76
## 77    11.732265   12.8107882   14.675216   10            77
## 78  -126.205009 -134.7683368 -153.261911   10            78
## 79   127.947508  136.5823993  155.066544   10            79
## 80   123.316876  132.4239236  152.004794   10            80
## 81   -61.876446  -66.1772929  -76.204037   10            81
## 82   156.756064  166.0943056  190.862979   10            82
## 83    70.793629   76.9340738   85.716582   10            83
## 84    96.886725  102.0768187  117.942075   10            84
## 85    -3.220489   -5.2474142   -5.383675   10            85
## 86    11.278568   14.5259530   15.256053   10            86
## 87    79.308963   85.1295036   98.971417   10            87
## 88   120.558785  128.0352538  149.165499   10            88
## 89   -90.392170  -97.7979724 -111.875667   10            89
## 90    43.638264   47.6017584   52.829589   10            90
## 91  -111.690951 -120.4725769 -136.110287   10            91
## 92   -10.490053  -11.9680219  -14.183544   10            92
## 93    53.645228   58.9343266   66.082471   10            93
## 94   187.058396  199.7754412  230.830789   10            94
## 95    34.587398   38.4885609   40.813729   10            95
## 96  -127.517756 -135.5374689 -154.564845   10            96
## 97   -48.685973  -53.8553407  -59.389655   10            97
## 98     2.962367    3.3797833    5.339486   10            98
## 99    25.268283   28.6871413   32.013742   10            99
## 100 -120.525306 -129.9821866 -147.774485   10           100
```

```r
scalar_12 <- lm(formula = x1 ~ x2, data = fully_evolved)
summary(scalar_12)
```

```
##
## Call:
## lm(formula = x1 ~ x2, data = fully_evolved)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.45364 -0.69146 -0.01835  0.85219  2.22688
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.1252245  0.1049344   1.193    0.236
## x2          0.9367361  0.0009676 968.123   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.041 on 98 degrees of freedom
## Multiple R-squared:  0.9999, Adjusted R-squared:  0.9999
## F-statistic: 9.373e+05 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
scalar_13 <- lm(formula = x1 ~ x3, data = fully_evolved)
summary(scalar_13)
```

```
##
## Call:
## lm(formula = x1 ~ x3, data = fully_evolved)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.46573 -0.76497 -0.07726  0.83960  1.50127
##
## Coefficients:
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept) -0.0074583  0.0898707   -0.083    0.934
## x3           0.8154802  0.0007213 1130.599   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8914 on 98 degrees of freedom
## Multiple R-squared:  0.9999, Adjusted R-squared:  0.9999
## F-statistic: 1.278e+06 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
scalar_23 <- lm(formula = x2 ~ x3, data = fully_evolved)
summary(scalar_23)
```

```
##
## Call:
## lm(formula = x2 ~ x3, data = fully_evolved)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.01590 -0.77884  0.04769  0.69645  3.10099
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.140766   0.109025  -1.291      0.2
## x3           0.870500   0.000875 994.848   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.081 on 98 degrees of freedom
## Multiple R-squared:  0.9999, Adjusted R-squared:  0.9999
## F-statistic: 9.897e+05 on 1 and 98 DF,  p-value: < 2.2e-16
```

```r
####################################
#### Step 4: Animate the batch! ####
####################################

# Plot the evolution arrays of the batch elements
batch_data2 <- evolved_batch
# Pairwise scatter plots
plot_12 <- batch_2d_customplot(batch_data2, 1, 2)
batch_animation2_1 <- plot_12 + transition_time(time = time)

plot_23 <- batch_2d_customplot(batch_data2, 2, 3)
batch_animation2_2 <- plot_23 + transition_time(time = time)

plot_13 <- batch_2d_customplot(batch_data2, 1, 3)
batch_animation2_3 <- plot_13 + transition_time(time = time)
```

```r
# Set me to true!
bool_animate1 <- F
if(bool_animate1){batch_animation2_1}
```

```r
# Set me to true!
bool_animate2 <- F
if(bool_animate2){batch_animation2_2}
```

```r
# Set me to true!
bool_animate3 <- F
if(bool_animate3){batch_animation2_3}
```