Computational Eigenvector Simulation

Ali Taqi

Example 1: A Symmetric Stochastic Matrix

Step 0: Setup the matrix

```
# Set seed
set.seed(23)
# Set parameters
N <- 5
# Generate matrix
P <- RM_stoch(N, symm = T, sparsity = F)
```

The Matrix

```
##            [,1]      [,2]         [,3]       [,4]       [,5]
## [1,] 0.00000000 0.1451091 3.996388e-01 0.07104614 0.38420596
## [2,] 0.15891616 0.0000000 3.545053e-01 0.28944134 0.19713716
## [3,] 0.35591174 0.2882864 1.110223e-16 0.26964291 0.08615895
## [4,] 0.06107857 0.2272143 2.602931e-01 0.00000000 0.45141403
## [5,] 0.32393981 0.1517733 8.156915e-02 0.44271773 0.00000000
```

Eigenvalues of the Symmetric Stochastic Matrix

```
spectrum(P)
```

```
##       Re Im  Norm Order
## 1  1.000  0 1.000     1
## 2 -0.653  0 0.653     2
## 3 -0.327  0 0.327     3
## 4 -0.058  0 0.058     4
## 5  0.037  0 0.037     5
```

Step 1: Get the batch

```
# Set batch parameters
B <- 100
# Create batch
batch <- generate_batch(N = N, batch_size = B)
head(batch)
```

```
##              x1         x2         x3         x4         x5
## 1  0.6409013881  0.3698745  0.7667785 -0.7761584  0.5576680
## 2  0.2420218312 -0.4516971 -0.3970607 -0.3019124 -0.3417379
## 3  0.6190078631 -0.4510357  0.8781898  0.8045341 -0.6458938
## 4  0.5965025560 -0.8344038 -0.4286165  0.3226320  0.5695485
## 5  0.5531537463 -0.7563570 -0.2553605 -0.4044565  0.5054127
## 6 -0.0008447589  0.6356071 -0.3917425  0.6498076  0.7860833
```

Step 2: Evolve the batch

```
# Set evolution parameters
steps <- 20
# Evolve batch
evolved_batch <- evolve_batch(batch, P, steps)
head(evolved_batch)
```

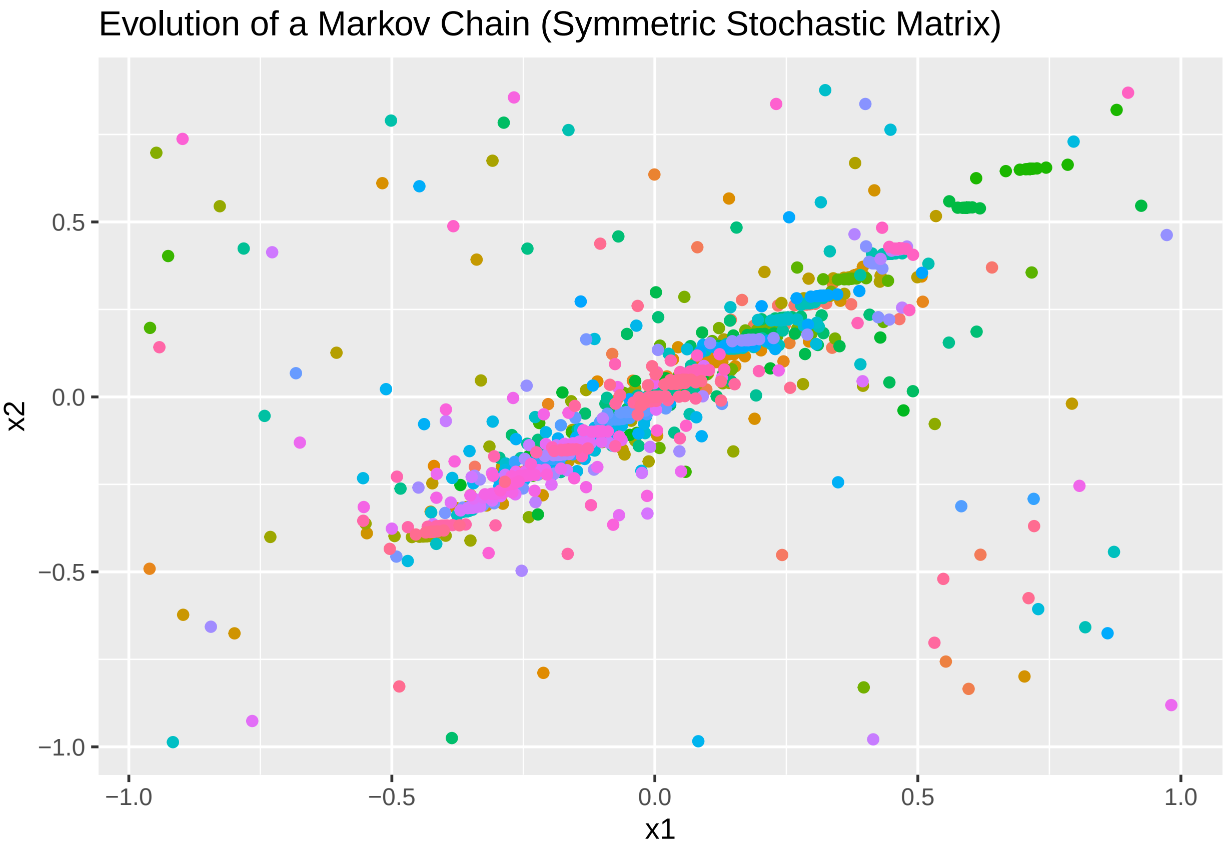
```
##          x1        x2        x3         x4         x5 time element_index
## 1 0.6409014 0.3698745 0.7667785 -0.7761584 0.55766801    0             1
## 2 0.4649287 0.2223373 0.2307113  0.6062365 0.03485016    1             1
## 3 0.1657633 0.2770113 0.4252652  0.1750235 0.51600084    2             1
## 4 0.3732218 0.2647349 0.2520946  0.4350678 0.23394490    3             1
## 5 0.2341515 0.2611935 0.3753317  0.2746883 0.41369906    4             1
## 6 0.3258840 0.2673821 0.2914151  0.3765932 0.29778966    5             1
##        r_x1      r_x2      r_x3       r_x4        r_x5
## 1        NA        NA        NA         NA          NA
## 2 0.7254295 0.6011155 0.3008839 -0.7810731  0.06249267
## 3 0.3565348 1.2459054 1.8432782  0.2887049 14.80626706
## 4 2.2515351 0.9556829 0.5927938  2.4857685  0.45338086
## 5 0.6273788 0.9866229 1.4888529  0.6313689  1.76836110
## 6 1.3917658 1.0236934 0.7764200  1.3709840  0.71982193
```

```
tail(evolved_batch)
```

```
##               x1          x2          x3          x4          x5 time
## 2095 -0.02475557 -0.02261650 -0.02782726 -0.02879687 -0.02939656   15
## 2096 -0.02477977 -0.02261915 -0.02780444 -0.02882274 -0.02936667   16
## 2097 -0.02476396 -0.02261743 -0.02781934 -0.02880584 -0.02938620   17
## 2098 -0.02477428 -0.02261855 -0.02780961 -0.02881688 -0.02937344   18
## 2099 -0.02476754 -0.02261782 -0.02781597 -0.02880967 -0.02938178   19
## 2100 -0.02477195 -0.02261830 -0.02781181 -0.02881438 -0.02937633   20
##      element_index      r_x1      r_x2      r_x3      r_x4      r_x5
## 2095           100 0.9985056 0.9998191 1.0012581 0.9986275 1.0015588
## 2096           100 1.0009775 1.0001174 0.9991797 1.0008983 0.9989830
## 2097           100 0.9993621 0.9999236 1.0005361 0.9994136 1.0006651
## 2098           100 1.0004169 1.0000499 0.9996500 1.0003834 0.9995658
## 2099           100 0.9997277 0.9999675 1.0002287 0.9997497 1.0002838
## 2100           100 1.0001779 1.0000212 0.9998507 1.0001636 0.9998147
```

Step 3: Analyze the batch

```r
# Plot the evolution arrays of the batch elements
batch_data <- evolved_batch
# 2d plot
.batch_2d_plot(batch_data, "(Symmetric Stochastic Matrix)")
```



Example 2: A Symmetric Normal Matrix

Step 0: Setup the matrix

```
# Set seed
set.seed(6)
# Set parameters
N <- 7
# Generate matrix
P <- RM_norm(N, symm = T)
```

The Matrix

```
##             [,1]        [,2]      [,3]        [,4]        [,5]        [,6]
## [1,]  0.26960598 -0.62998541  0.8686598  1.72719552  0.02418764  0.36802518
## [2,] -0.62998541  0.04487299 -1.0483972  1.72785109 -1.17859974  0.65320671
## [3,]  0.86865983 -1.04839720  1.7076774 -1.09437298 -0.28928182  2.20741296
## [4,]  1.72719552  1.72785109 -1.0943730  0.19038081 -1.16973591 -0.03808156
## [5,]  0.02418764 -1.17859974 -0.2892818 -1.16973591 -1.17939052 -1.05871745
## [6,]  0.36802518  0.65320671  2.2074130 -0.03808156 -1.05871745 -0.88516413
## [7,] -1.30920430 -0.36856649  0.5187490  2.35420426  1.13790261 -0.43233430
##             [,7]
## [1,] -1.30920430
## [2,] -0.36856649
## [3,]  0.51874901
## [4,]  2.35420426
## [5,]  1.13790261
## [6,] -0.43233430
## [7,]  0.01423374
```

The Eigenvalues

```
spectrum(P)
```

```
##       Re Im  Norm Order
## 1  3.901  0 3.901     1
## 2  2.897  0 2.897     2
## 3  1.828  0 1.828     3
## 4  0.969  0 0.969     4
## 5 -2.093  0 2.093     5
## 6 -2.695  0 2.695     6
## 7 -4.646  0 4.646     7
```

Step 1: Get the batch

```
# Set batch parameters
B <- 100
# Create batch
batch <- generate_batch(N = N, batch_size = B)
head(batch)
```

```
##           x1          x2          x3         x4          x5          x6
## 1  0.1491569 -0.72196564 -0.04317135  0.5733141  0.35564018 -0.60489532
## 2  0.1247489  0.79350360  0.68179065 -0.7479271  0.48834713  0.72191159
## 3 -0.8737344 -0.28094312 -0.88445288  0.9142893 -0.09083563  0.33213757
## 4 -0.5794810  0.16956027 -0.05121148  0.4202677  0.20333061 -0.71063676
## 5 -0.8926129  0.01258537 -0.26475065  0.5571887  0.16237925 -0.09197025
## 6 -0.5357196 -0.63697162 -0.75745028 -0.5809727  0.94120440  0.41979377
##            x7
## 1 -0.44933925
## 2 -0.59750937
## 3  0.08854953
## 4  0.28939230
## 5 -0.17751117
## 6 -0.27196257
```

Step 2: Evolve the batch

```
# Set evolution parameters
steps <- 20
# Evolve batch
evolved_batch <- evolve_batch(batch, P, steps)
# View
head(evolved_batch)
```

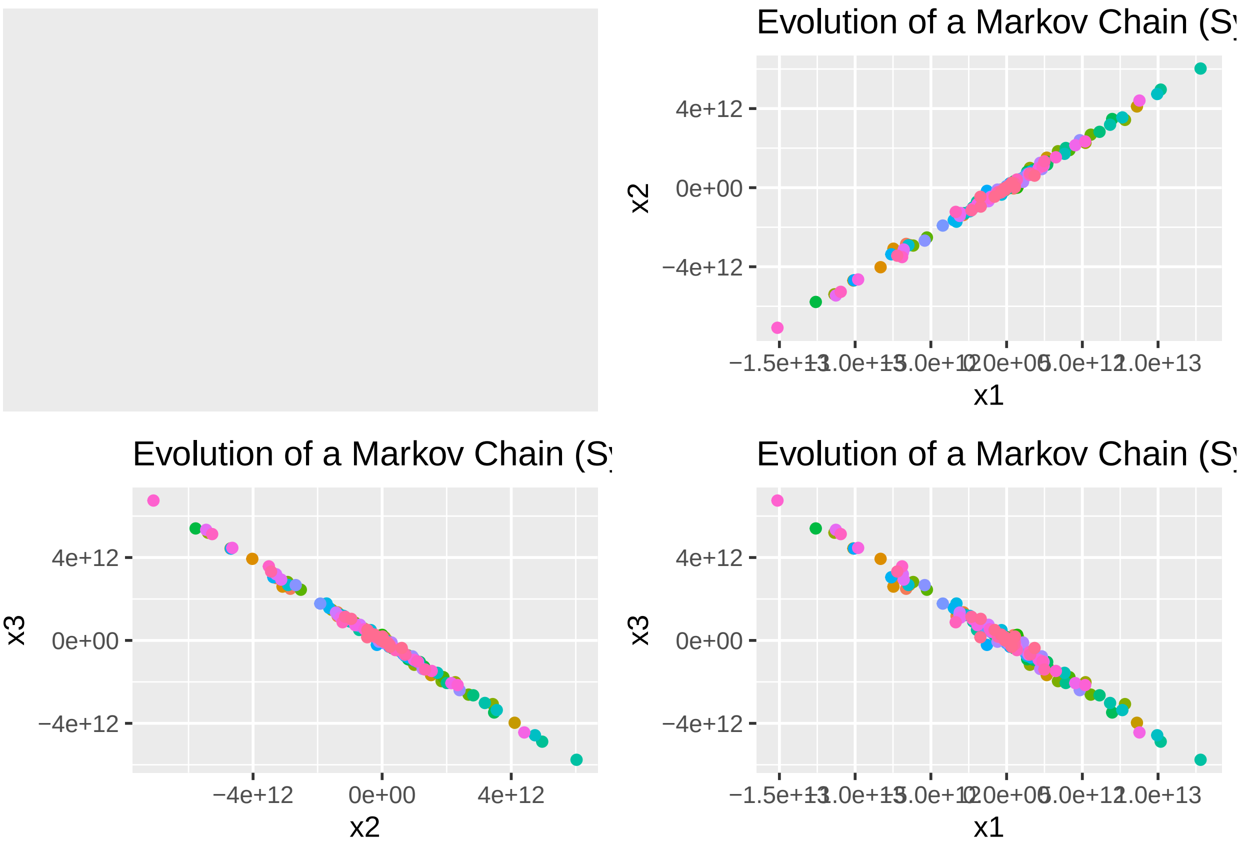
```
##             x1          x2            x3           x4            x5          x6
## 1    0.1491569  -0.7219656   -0.04317135    0.5733141    0.35564018  -0.6048953
## 2    1.8220279   0.2608309   -1.48589712   -2.2842386   -0.09395081  -0.1806551
## 3   -7.6721568  -4.2909171    1.96758802    9.7504765    5.48223866  -2.9824069
## 4   29.5175801  14.1963815  -22.12238190  -47.5769752  -20.23058557  -1.0878630
## 5 -159.5703278 -69.6889135   50.80666529  215.6151037  119.98238054 -23.0966125
## 6  659.6550609 329.9554166 -398.64258467 -995.6849580 -521.08898912 -25.0505174
##             x7 time element_index      r_x1        r_x2       r_x3      r_x4
## 1   -0.4493393    0             1        NA          NA         NA        NA
## 2    2.0579239    1             1 12.215515  -0.3612789  34.418591 -3.984271
## 3   -8.6294238    2             1 -4.210779 -16.4509523  -1.324175 -4.268589
## 4   43.0060281    3             1 -3.847364  -3.3084726 -11.243401 -4.879451
## 5 -189.2967137    4             1 -5.405942  -4.9089209  -2.296618 -4.531921
## 6  912.3723822    5             1 -4.133946  -4.7346902  -7.846265 -4.617881
##          r_x5       r_x6      r_x7
## 1          NA         NA        NA
## 2  -0.2641738  0.2986551 -4.579889
## 3 -58.3522223 16.5088467 -4.193267
## 4  -3.6902052  0.3647601 -4.983650
## 5  -5.9307419 21.2311771 -4.401632
## 6  -4.3430459  1.0845970 -4.819800
```

```
tail(evolved_batch)
```

```
##                 x1            x2            x3            x4            x5
## 2095  6.570474e+08     697439254    -980674198    -649730990    -652934258
## 2096 -3.905870e+09    -324596307    -928349668    6855129460    2648624160
## 2097  1.479978e+10   12862760797  -17109704795  -16903174151  -13787251178
## 2098 -8.186500e+10  -15484066439   -4106173778  136768883948   58304391095
## 2099  3.290555e+11  244508276283 -307267691319 -408395382044 -293149596831
## 2100 -1.729250e+12 -462917816492  154159220493 2782480695396 1275327233711
##                 x6            x7 time element_index      r_x1       r_x2
## 2095    -275550212  1.185720e+09   15           100 -3.491798 64.1092673
## 2096   -1020075690 -3.762554e+09   16           100 -5.944578 -0.4654116
## 2097   -4234321201  2.429135e+10   17           100 -3.789111 -39.6269475
## 2098  -15432862582 -8.629803e+10   18           100 -5.531502  -1.2037903
## 2099  -65272731210  5.045253e+11   19           100 -4.019490 -15.7909602
## 2100 -231883098112 -1.939934e+12   20           100 -5.255194  -1.8932603
##            r_x3       r_x4      r_x5     r_x6      r_x7
## 2095 10.7282693  -1.846013 -5.481967 4.117781 -7.467824
## 2096  0.9466443 -10.550720 -4.056494 3.701959 -3.173222
## 2097 18.4302374  -2.465770 -5.205439 4.150987 -6.456079
## 2098  0.2399909  -8.091314 -4.228863 3.644708 -3.552624
## 2099 74.8306594  -2.986026 -5.027916 4.229464 -5.846313
## 2100 -0.5017098  -6.813203 -4.350431 3.552526 -3.845068
```

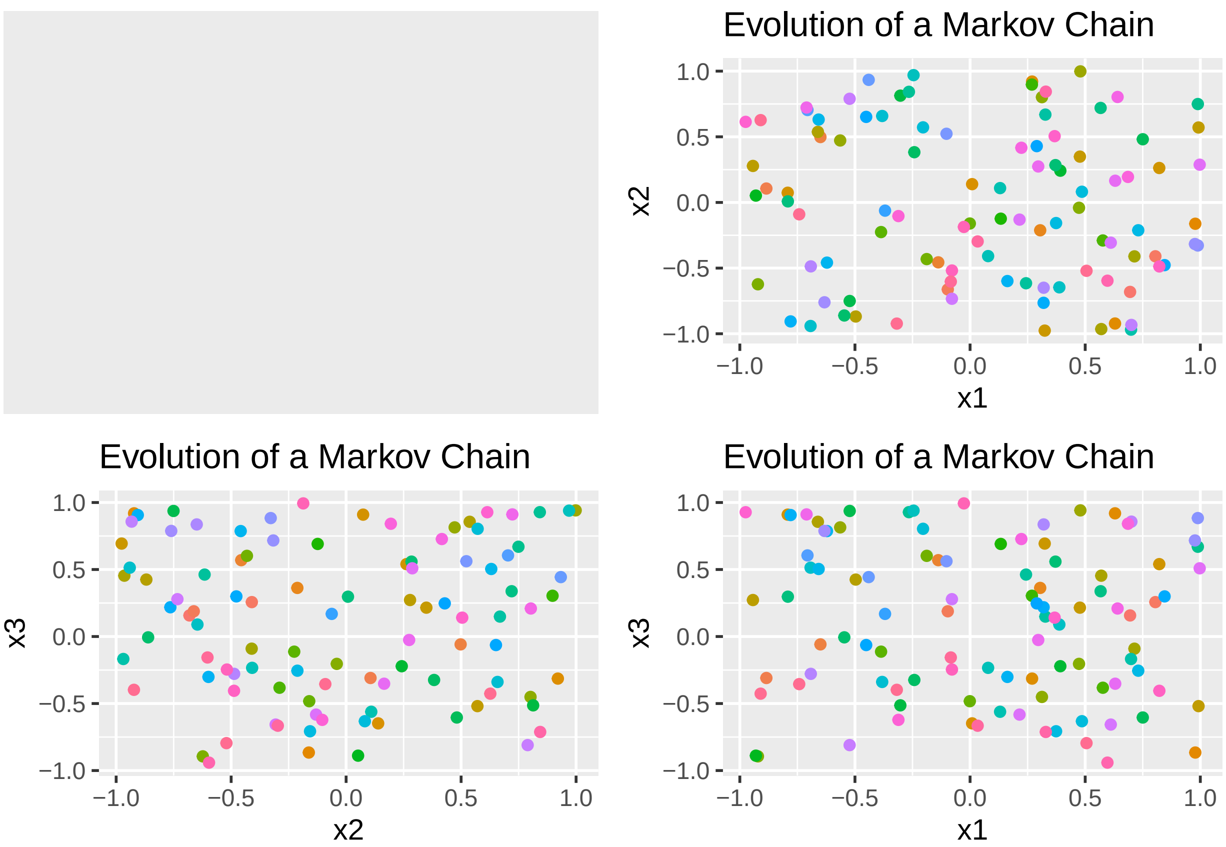
Step 3: Analyze the batch

```r
# Plot the evolution arrays of the batch elements
batch_data <- evolved_batch
# 3d plot
.batch_3d_plot(batch_data, "(Symmetric Normal Matrix)")
```



```r
set.seed(27)
N <- 5
P <- RM_stoch(N, symm = T, sparsity = T)
# Set batch parameters
B <- 100
# Create batch
batch <- generate_batch(N = N, batch_size = B)
# Set evolution parameters
steps <- 10
# Evolve batch
evolved_batch <- evolve_batch(batch, P, steps)
```

```r
.batch_3d_plot(evolved_batch)
```



```
#.batch_2d_customplot(evolved_batch, n1 = 1, n2 = 2) + transition_time(V4)
```