# Markov Chains 2

Mihir Arjunwadkar

Centre for Modeling and Simulation
Savitribai Phule Pune University

A homogeneous Markov is completely specified by the time-independent state-to-state *transition probability matrix* TPM

$$\mathcal{P} = \begin{array}{c c} & \begin{array}{cccc} 1 & 2 & \dots & k \end{array} \\ \begin{array}{c} 1 \\ 2 \\ \vdots \\ k \end{array} & \begin{bmatrix} P(X_2 = 1|X_1 = 1) & P(X_2 = 2|X_1 = 1) & \dots & P(X_2 = k|X_1 = 1) \\ P(X_2 = 1|X_1 = 2) & P(X_2 = 2|X_1 = 2) & \dots & P(X_2 = k|X_1 = 2) \\ \vdots & \vdots & \vdots & \vdots \\ P(X_2 = 1|X_1 = k) & P(X_2 = 2|X_1 = k) & \dots & P(X_2 = k|X_1 = k) \end{bmatrix} \end{array}$$

Rows :: $X_1$ :: the immediate past (the present)

Columns :: $X_2$ :: the present (immediate future)

Row & column labels :: the states of the Markov chain

Given the one-step TPM $\mathcal{P}$, what is the probability of going from state $i$ to state $j$ in 2 steps?

Consider all possible *independent paths* of reaching $j$ from $i$ in 2 steps:

| Path | Probability |
|---|---|
| $i \to 1 \to j$ | $P(X_3 = j \vert X_2 = 1) \times P(X_2 = 1 \vert X_1 = i)$ |
| $i \to 2 \to j$ | $P(X_3 = j \vert X_2 = 2) \times P(X_2 = 2 \vert X_1 = i)$ |
| $\vdots$ | $\vdots$ |
| $i \to i \to j$ | $P(X_3 = j \vert X_2 = i) \times P(X_2 = i \vert X_1 = i)$ |
| $\vdots$ | $\vdots$ |
| $i \to j \to j$ | $P(X_3 = j \vert X_2 = j) \times P(X_2 = j \vert X_1 = i)$ |
| $\vdots$ | $\vdots$ |
| $i \to k \to j$ | $P(X_3 = j \vert X_2 = k) \times P(X_2 = k \vert X_1 = i)$ |

and sum up their probabilities to get $P(X_3 = j | X_1 = i)$:

$$\begin{aligned}
P(X_3 = j | X_1 = i) \quad = & \quad\; P(X_3 = j | X_2 = 1) P(X_2 = 1 | X_1 = i) \\
& + P(X_3 = j | X_2 = 2) P(X_2 = 2 | X_1 = i) \\
& + \vdots \\
& + P(X_3 = j | X_2 = i) P(X_2 = i | X_1 = i) \\
& + \vdots \\
& + P(X_3 = j | X_2 = j) P(X_2 = j | X_1 = i) \\
& + \vdots \\
& + P(X_3 = j | X_2 = k) P(X_2 = k | X_1 = i) \\
= & \sum_{m=1}^{k} P(X_3 = j | X_2 = m) P(X_2 = m | X_1 = i) \\
= & \sum_{m=1}^{k} \mathcal{P}_{mj} \mathcal{P}_{im} = \sum_{m=1}^{k} \mathcal{P}_{im} \mathcal{P}_{mj} = (\mathcal{P} \times \mathcal{P})_{ij} = (\mathcal{P}^2)_{ij}
\end{aligned}$$

**2-step** TPM

$\mathcal{P}^2$ :: The 2-step TPM for a homogeneous Markov chain is the square of the 1-step TPM $\mathcal{P}$.

Note: This is matrix multiplication. This is *not* squaring the matrix elementwise.

***n*-step** TPM

$\mathcal{P}^n$ :: Following similar arguments, induction, etc., one can show that the $n$-step TPM for a homogeneous Markov chain is the $n$th power of the 1-step TPM $\mathcal{P}$.

Note: This is repeated matrix multiplication. This is *not* taking powers elementwise.

Suppose that
– state space is $\mathcal{X} = \{1, 2, \ldots, k\}$
– $k \times k$ TPM $\mathcal{P}$ is given (valid, row-normalized, etc.)

Choose a starting state $i_1$ from the state space $\mathcal{X}$.

Consider the $i_1$th row of $\mathcal{P}$ as a discrete PMF.
Sample a state randomly using this PMF.
Suppose this is state $i_2$ from $\mathcal{X}$ at $t = 2$.

Consider the $i_2$th row of $\mathcal{P}$ as a discrete PMF.
Sample a state randomly using this PMF.
Suppose this is state $i_3$ from $\mathcal{X}$ at $t = 3$.

etc. etc. etc.

```r
is.markov <- function( P )
 {
  # Check whether matrix P is a valid Markov transition probability matrix

  (   is.matrix( P )
   && ( nrow( P ) == ncol( P ) )
   && all( P >= 0 )
   && all( rowSums( P ) == 1 )
  )
 }

sample.mc <- function( n, P, initial = 1 )
 {
  # Generate a length-n random realization of a Markov chain
  # starting from given initial state. States of the Markov
  # chain are taken to be 1:ncol( P ).

  stopifnot( is.markov( P ), length( n ) == 1, as.integer( n ) == n, n > 1 )

  states <- 1:ncol( P )

  stopifnot( length( initial ) == 1, initial %in% states )

  mc <- rep( initial, n )
  for ( i in 2:n ) mc[i] <- sample( states, 1, prob = P[mc[i-1],] )

  mc
 }
```

# Estimating Markov matrix from data

Assume

- homogeneous Markov chain with TPM $\mathcal{P}$
- $\mathcal{X} = \{1, \ldots, k\}$
- index set $T = \{1, 2, \ldots\}$

Data

$X_1, \ldots, X_n$ : an observed realization (data) of the Markov chain

Unknown parameters

- Initial distribution $P(X_1 = i)$, $i = 1, \ldots, k$
- Transition probabilities $\mathcal{P}_{ij} = P(X_t = j | X_{t-1} = i)$, $i, j = 1, \ldots, k$

Single observation $X_1 \implies$ can't estimate the initial PMF

# Estimating Markov matrix from data

Suppose

- $n_{ij}$ : $\#$ of observed $i \to j$ transitions in one time step
- $n_i = \sum_{j=1}^{k} n_{ij}$

These are computed from the data.

Likelihood function (ignoring the initial distribution)

$$\mathcal{L}(\mathcal{P}) = \prod_{i=1}^{k} \prod_{j=1}^{k} \mathcal{P}_{ij}^{n_{ij}}$$

Maximize $\mathcal{L}$ with respect to the unknown parameters $\mathcal{P}_{ij}$ subject to the constraints (a) $\mathcal{P}_{ij} \geq 0$ for all $i, j$ and (b) $\sum_{j=1} \mathcal{P}_{ij} = 1$ for all $i$:

$$\widehat{\mathcal{P}}_{ij} = \begin{cases} \frac{n_{ij}}{n_i} & \text{if } n_i > 0 \\ 0 & \text{if } n_i = 0 \text{ (i.e., no observed transition from state } i\text{)} \end{cases}$$

*All of Statistics* by Larry Wasserman, Springer (2004) Ch 23

# Estimating Markov matrix from data

```
tpm.hat <- function( x, states = NULL )
 {
  # Estimate TPM from Markov chain data x

  x <- as.vector( x )

  if ( is.null( states ) ) states <- unique( x )
  k <- length( states )

  P <- matrix( 0, ncol = k, nrow = k, dimnames = list( states, states ) )
  for ( i in 2:length( x ) ) P[x[i-1],x[i]] <- P[x[i-1],x[i]] + 1

  norm <- 1 / rowSums( P )
  norm[!is.finite( norm )] <- 0

  for ( i in 1:k ) P[i,] <- P[i,] * norm[i]

  P
 }
```

# Estimating Markov matrix from data

```r
source( 'markov.r' )
source( 'markov-estimation.r' )

# An arbitrary Markov chain whose states are the DNA alphabet
#
# This chain avoids repeated letters AA, CC, GG, TT;
# all other transitions occur with equal probabilities.

states    <- c( 'A', 'C', 'G', 'T' )
P         <- matrix( 1/3, 4, 4, dimnames = list( states, states ) )
diag( P ) <- 0

stopifnot( is.markov( P ) )

# A random data realization from this Markov chain
n <- 100
x <- sample.mc( n, P, sample( 1:ncol( P ), 1 ) )
x <- rownames( P )[x] # sequence of DNA letters

# Estimate the Markov TPM from this data
P.hat <- tpm.hat( x, rownames( P ) )

# Exercise: compare P and P.hat
# -- for many realizations of data for fixed data size n: boxplots
# -- long-run behaviour of the estimator for increasing n: LLN? CLT?
```

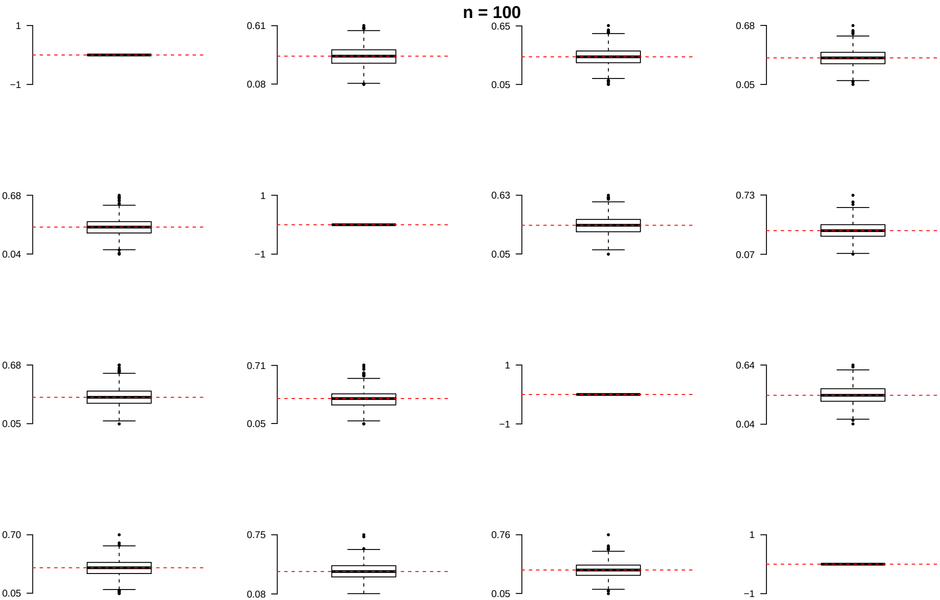
# Estimating Markov matrix from data

```
> P
          A         C         G         T
A 0.0000000 0.3333333 0.3333333 0.3333333
C 0.3333333 0.0000000 0.3333333 0.3333333
G 0.3333333 0.3333333 0.0000000 0.3333333
T 0.3333333 0.3333333 0.3333333 0.0000000

> x
 [1] "G" "T" "G" "T" "A" "T" "G" "A" "G" "A" "T" "G" "A" "G" "C" "T" "A" "G"
[19] "C" "G" "T" "C" "T" "C" "A" "G" "A" "C" "T" "G" "A" "C" "A" "C" "A" "T"
[37] "C" "G" "C" "T" "C" "G" "A" "G" "T" "G" "A" "G" "C" "T" "C" "A" "G" "A"
[55] "T" "C" "T" "C" "A" "T" "G" "A" "C" "A" "T" "C" "G" "T" "C" "T" "G" "C"
[73] "T" "G" "C" "G" "A" "T" "C" "T" "C" "T" "C" "G" "A" "C" "T" "A" "G" "C"
[91] "G" "A" "C" "A" "C" "A" "G" "A" "C" "G"
> P.hat
          A         C         G         T
A 0.0000000 0.3333333 0.3750000 0.2916667
C 0.2962963 0.0000000 0.2962963 0.4074074
G 0.5200000 0.2800000 0.0000000 0.2000000
T 0.1304348 0.5217391 0.3478261 0.0000000

> x
 [1] "G" "C" "G" "A" "T" "G" "C" "T" "C" "T" "G" "T" "G" "A" "T" "A" "G" "T"
[19] "A" "G" "T" "C" "T" "C" "A" "T" "C" "T" "G" "T" "G" "A" "T" "A" "T" "C"
[37] "T" "C" "T" "A" "T" "C" "G" "T" "G" "A" "G" "T" "C" "A" "T" "A" "C" "T"
[55] "G" "C" "G" "T" "A" "G" "T" "A" "G" "C" "T" "A" "T" "A" "C" "A" "G" "A"
[73] "T" "A" "C" "A" "C" "A" "T" "A" "G" "A" "G" "T" "A" "C" "T" "G" "C" "T"
[91] "A" "C" "A" "G" "T" "C" "A" "C" "T" "G"
> P.hat
          A         C         G         T
A 0.0000000 0.2692308 0.3461538 0.3846154
C 0.3333333 0.0000000 0.1428571 0.5238095
G 0.2857143 0.2380952 0.0000000 0.4761905
T 0.4193548 0.2903226 0.2903226 0.0000000
```

# Estimating Markov matrix from data



# Estimating Markov matrix from data