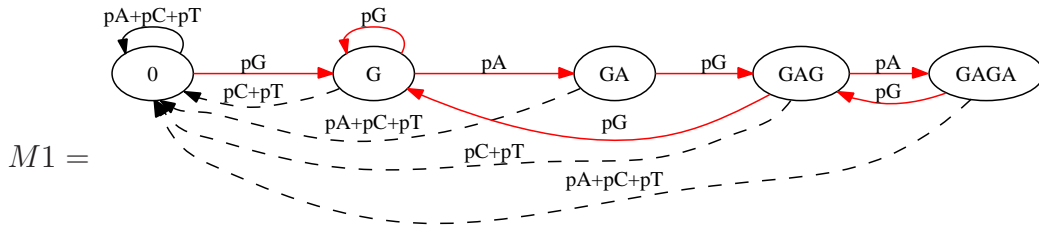


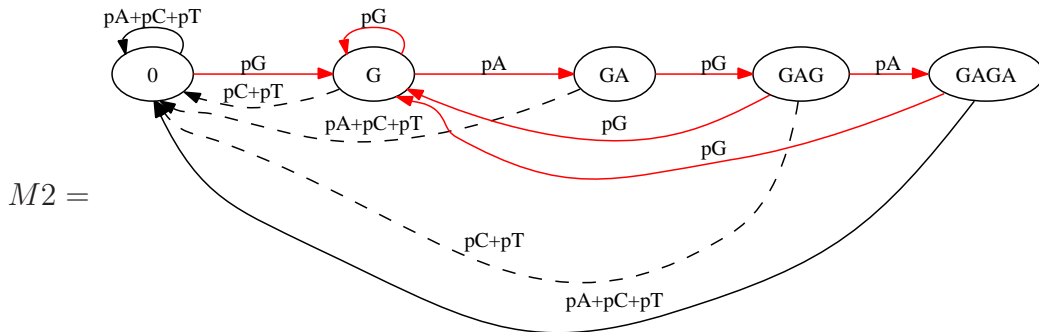
Math 283, Spring 2006, Prof. Tesler – May 22, 2006  
 Markov chains and the number of occurrences of a word in a sequence  
 (Chapter 4.5–4.9; 11.1,2,4,6)

1. FINITE STATE MACHINE FOR COUNTING OCCURRENCES OF A WORD OR MOTIF

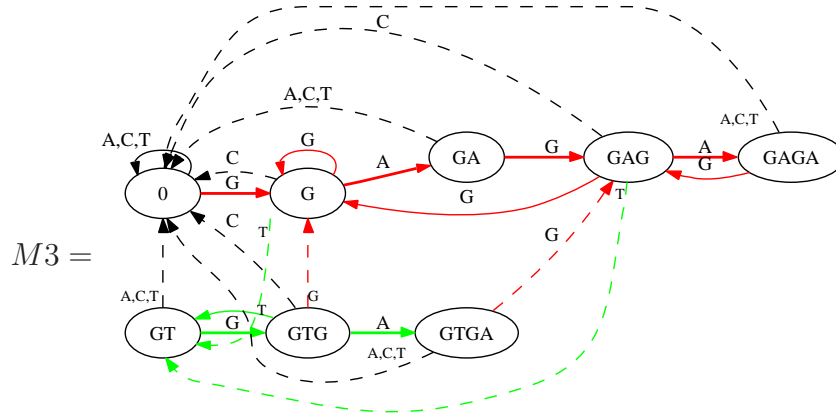
We revisit the problem of counting the number of occurrences of a word in a sequence. This is a finite state machine for analyzing occurrences of the word  $w = \text{GAGA}$  allowing for overlaps:



This one does not allow overlaps:



This one detects overlapping occurrences of the motif  $\{\text{GAGA}, \text{GTGA}\}$ :



For non-overlapping occurrences, replace the outgoing edges from **GAGA** and **GTGA** by copies of 0's outgoing edges.

More generally, finite state machines can be constructed for any regular expression.

**Using these finite state machines:** Start in the state 0. Scan a sequence  $\tau = \tau_1\tau_2 \dots \tau_N$  one character at a time from left to right. When examining  $\tau_j$ , move from the current state to the next state according to which edge  $\tau_j$  is on.

(In a general Markov chain, there is not necessarily one starting state, but instead, a probability of choosing each state as the starting state.)

$M3$  is shown in the regular notation for finite automata, with  $\tau_j$  appearing on one of the edges;  $M1, M2$  are shown in the notation for a Markov chain, with a probability appearing on the edge instead ( $pA, \dots, pT$  are the probabilities of each letter). In machines  $M1$  or  $M2$ , the number of times we are in the state **GAGA** is the desired count of overlapping or non-overlapping occurrences. In  $M3$ , use the number of times in states **GAGA** or **GTGA**.

For example, suppose  $\tau = \text{CAGAGGTCGAGAGT} \dots$ ; this is what happens in machine  $M1$ :

Time $t$	State at $t$	$\tau_{t+1}$	Time $t$	State at $t$	$\tau_{t+1}$	Time $t$	State at $t$	$\tau_{t+1}$
1	0	C	6	GAG	G	11	GA	G
2	0	A	7	G	T	12	GAG	A
3	0	G	8	0	C	13	GAGA	G
4	G	A	9	0	G	14	GAG	T
5	GA	G	10	G	A	15	0	...

In machine  $M2$ , the only difference is that  $t = 14$  is state  $G$ .

*Names of states in overlapping case (M1):* If we are in state  $w_1 \dots w_r$  after examining  $\tau_j$ , it means that the last  $r$  letters of  $\tau$  ending at  $\tau_j$  agreed with the first  $r$  letters of  $w$ , and that there is not a longer prefix of  $w$  that ends at  $\tau_j$ .

*Names of states in non-overlapping case (M2):* It's almost the same as  $M1$ , except the outgoing edges of **GAGA** are replaced by copies of the outgoing edges of  $0$  (same probabilities and endpoints but starting at **GAGA** instead of starting at  $0$ ), so that the second **GA** cannot count towards an overlapping occurrence of **GAGA**.

*Names of states in motif, overlapping case (M3):* State  $w_1 \dots w_r$  means the last  $r$  letters of  $\tau$  ending at  $\tau_j$  are  $w_1 \dots w_r$ ; some word in the motif starts with  $w_1 \dots w_r$ ; and there isn't a longer word ending at  $\tau_j$  that is a prefix of any word in the motif.

This is how to design a machine like  $M1$  for any word (to analyze overlapping occurrences of the word  $w = w_1 w_2 \dots w_k$ ):

- Make a graph with  $k + 1$  nodes (called "states"):  $\emptyset, w_1, w_1 w_2, w_1 w_2 w_3, \dots, w_1 w_2 \dots w_k$ .
- Each node has four outgoing edges (which may later be merged into fewer). For each node  $w_1 \dots w_r$  and each letter  $x = A, C, G, T$ , determine the longest suffix  $s$  (possibly  $\emptyset$ ) of  $w_1 \dots w_r x$  that is one of the states. Draw an edge from  $w_1 \dots w_r$  to  $s$ , with probability  $p_x$  on that edge.  
Note that if  $r < k$  and  $x = w_{r+1}$ , this will be an edge to  $w_1 \dots w_{r+1}$ .
- If there are several edges in the same direction between two nodes, combine them into one edge by adding the probabilities together.

For non-overlapping occurrences of  $w$  (as in  $M2$ ), replace the outgoing edges from  $w$  by copies of  $0$ 's outgoing edges.

For motifs (overlapping or non-overlapping), the states are all the prefixes of all the words. If a prefix is common to more than one word, it is only used as one state.

## 2. MARKOV CHAINS

A Markov chain is similar to a finite state machine, but incorporates probabilities.

Let  $S$  be a set of "states." We will take  $S$  to be a discrete finite set, such as  $S = \{1, 2, \dots, s\}$ .

Let  $t = 1, 2, \dots$  denote the "time."

Let  $X_1, X_2, \dots$  denote a sequence of random variables, which take on values from  $S$ .

The  $X_t$ 's form a *Markov chain* if they obey these rules:

- The probability of being in a certain state at time  $t + 1$  only depends on the state at time  $t$ , not on any earlier states:  
$$P(X_{t+1} = x_{t+1} | X_0 = x_0, \dots, X_t = x_t) = P(X_{t+1} = x_{t+1} | X_t = x_t):$$
- The probability of transitioning from state  $i$  at time  $t$  to state  $j$  at time  $t + 1$  only depends on  $i$  and  $j$ , and does not depend on the time: there are values  $p_{ij}$  such that  $P(X_{t+1} = j | X_t = i) = p_{ij}$  at all times  $t$ . These are put into an  $s \times s$  *transition matrix*.

The *transition matrix*,  $P1$ , of the Markov chain  $M1$  is

$$\begin{array}{l}
\text{To state 1} \quad 2 \quad 3 \quad 4 \quad 5 \\
\text{From state 1: } 0 \\
\text{From state 2: } \mathbf{G} \\
\text{From state 3: } \mathbf{GA} \\
\text{From state 4: } \mathbf{GAG} \\
\text{From state 5: } \mathbf{GAGA}
\end{array}
\begin{bmatrix}
p_A + p_C + p_T & p_G & 0 & 0 & 0 \\
p_C + p_T & p_G & p_A & 0 & 0 \\
p_A + p_C + p_T & 0 & 0 & p_G & 0 \\
p_C + p_T & p_G & 0 & 0 & p_A \\
p_A + p_C + p_T & 0 & 0 & p_G & 0
\end{bmatrix}
=
\begin{bmatrix}
P_{11} & P_{12} & P_{13} & P_{14} & P_{15} \\
P_{21} & P_{22} & P_{23} & P_{24} & P_{25} \\
P_{31} & P_{32} & P_{33} & P_{34} & P_{35} \\
P_{41} & P_{42} & P_{43} & P_{44} & P_{45} \\
P_{51} & P_{52} & P_{53} & P_{54} & P_{55}
\end{bmatrix}$$

Notice that the entries in each row sum up to  $p_A + p_C + p_G + p_T = 1$ . A matrix with all entries  $\geq 0$  and all row sums equal to 1 is called a *stochastic matrix*. The transition matrix of a Markov chain is always a stochastic matrix. (For state machines in other contexts than Markov Chains, the transition matrix would have weights = 0 for no edge or  $\neq 0$  for an edge, but the nonzero weights may be defined differently, and the matrices are usually not stochastic.)

For  $M2$ , since edges leaving  $\mathbf{GAGA}$  were changed to copy 0's outgoing edges, the bottom row changes to a copy of the top row.

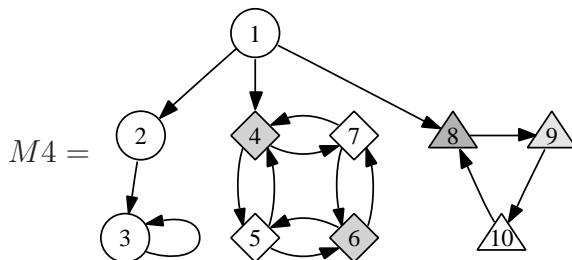
If all nucleotides have equal probabilities  $p_A = p_C = p_G = p_T = \frac{1}{4}$ , matrices  $P1$  and  $P2$  become

$$P1 = \begin{bmatrix} 3/4 & 1/4 & 0 & 0 & 0 \\ 1/2 & 1/4 & 1/4 & 0 & 0 \\ 3/4 & 0 & 0 & 1/4 & 0 \\ 1/2 & 1/4 & 0 & 0 & 1/4 \\ 3/4 & 0 & 0 & 1/4 & 0 \end{bmatrix}
\quad
P2 = \begin{bmatrix} 3/4 & 1/4 & 0 & 0 & 0 \\ 1/2 & 1/4 & 1/4 & 0 & 0 \\ 3/4 & 0 & 0 & 1/4 & 0 \\ 1/2 & 1/4 & 0 & 0 & 1/4 \\ 3/4 & 1/4 & 0 & 0 & 0 \end{bmatrix}$$

We will examine several questions that apply to Markov chains, using  $P1$  or  $P2$  as examples.

- (1) What is the probability of being in a particular state after  $n$  steps?
- (2) What is the probability of being in a particular state as  $n \rightarrow \infty$ ?
- (3) What is the “reverse” Markov chain?
- (4) If you are in state  $i$ , what is the expected number of time steps until the next time you are in state  $j$ ? What is the variance of this? What is the complete probability distribution?
- (5) Starting in state  $i$ , what is the expected number of visits to state  $j$  before reaching state  $k$ ?

**2.1. Technical conditions.** This Markov chain has a number of complications:



A Markov chain is *irreducible* if every state can be reached from every other state after enough steps. But  $M4$  is *reducible* since there are states that cannot be reached from each other: after sufficient time, you are either stuck in state 3, the component  $\{4, 5, 6, 7\}$ , or the component  $\{8, 9, 10\}$ .

State  $i$  has period  $d$  if the Markov chain can only go from state  $i$  to itself in multiples of  $d$  steps, where  $d$  is the maximum number that satisfies that. If  $d > 1$  then state  $i$  is periodic. A Markov chain is *periodic* if at least one state is periodic and is *aperiodic* if no states are periodic. All states in a component have the same period. In  $M4$ , component  $\{4, 5, 6, 7\}$  has period 2 and component  $\{8, 9, 10\}$  has period 3, so  $M4$  is periodic.

An *absorbing state* has all its outgoing edges going to itself. An irreducible Markov chain with two or more states cannot have any absorbing states. In  $M4$ , state 3 is an absorbing state.

There are generalizations to infinite numbers of discrete or continuous states and to continuous time, which we do not consider. We will work with Markov chains that are finite, discrete, irreducible, and aperiodic, unless otherwise stated. **For a finite discrete Markov chain on two or more states, being irreducible and aperiodic with no absorbing states is equivalent to  $P$  or a power of  $P$  having all entries be greater than 0.**

A Markov chain is  $k$ th order if the probability of  $X_t = i$  depends on the values of  $X_{t-1}, \dots, X_{t-k}$ . However, it can be converted to a first order Markov chain by making new states that essentially record the last  $k$  states into one new state. We will only look at first order Markov chains.

### 3. TRANSITION PROBABILITIES AFTER $n$ STEPS

The two-step transition probability for going from state  $i$  at time  $t$  to state  $j$  at time  $t + 2$  is computed by considering all the states it could be at during time  $t + 1$ :

$$\begin{aligned} P(X_{t+2} = j | X_t = i) &= \sum_r P(X_{t+1} = r | X_t = i) P(X_{t+2} = j | X_{t+1} = r, X_t = i) \\ &= \sum_r P(X_{t+1} = r | X_t = i) P(X_{t+2} = j | X_{t+1} = r) = \sum_r P_{ir} P_{rj} = (P^2)_{ij} \end{aligned}$$

(where  $P^2$  is the square of the matrix  $P$ ). In the same way, the transition matrix from time  $t$  to time  $t + n$  is  $P^n$  (for  $n \geq 0$ ):

$$\begin{aligned} P(X_{t+n} = j | X_t = i) &= \sum_{r_1, \dots, r_{n-1}} P(X_{t+1} = r_1 | X_t = i) P(X_{t+2} = r_2 | X_{t+1} = r_1) \cdots P(X_{t+n} = j | X_{t+n-1} = r_{n-1}) \\ &= \sum_{r_1, \dots, r_{n-1}} P_{ir_1} P_{r_1 r_2} \cdots P_{r_{n-1} j} = (P^n)_{ij} \end{aligned}$$

In this example, with  $P = P^1$ :

$$\begin{aligned} P^0 = I &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} & P^1 &= \begin{bmatrix} \frac{3}{4} & \frac{1}{4} & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ \frac{3}{4} & 0 & 0 & \frac{1}{4} & 0 \\ \frac{1}{2} & \frac{1}{4} & 0 & 0 & \frac{1}{4} \\ \frac{3}{4} & 0 & 0 & \frac{1}{4} & 0 \end{bmatrix} & P^2 &= \begin{bmatrix} \frac{11}{16} & \frac{1}{4} & \frac{1}{16} & 0 & 0 \\ \frac{11}{16} & \frac{3}{16} & \frac{1}{16} & \frac{1}{16} & 0 \\ \frac{11}{16} & \frac{1}{4} & 0 & 0 & \frac{1}{16} \\ \frac{11}{16} & \frac{3}{16} & \frac{1}{16} & \frac{1}{16} & 0 \\ \frac{11}{16} & \frac{1}{4} & 0 & 0 & \frac{1}{16} \end{bmatrix} \\ P^3 &= \begin{bmatrix} \frac{11}{16} & \frac{15}{64} & \frac{1}{16} & \frac{1}{64} & 0 \\ \frac{11}{16} & \frac{15}{64} & \frac{3}{64} & \frac{1}{64} & \frac{1}{64} \\ \frac{11}{16} & \frac{15}{64} & \frac{1}{64} & \frac{1}{64} & 0 \\ \frac{11}{16} & \frac{15}{64} & \frac{3}{64} & \frac{1}{64} & \frac{1}{64} \\ \frac{11}{16} & \frac{15}{64} & \frac{1}{16} & \frac{1}{64} & 0 \end{bmatrix} & P^4 &= \begin{bmatrix} \frac{11}{16} & \frac{15}{64} & \frac{15}{256} & \frac{1}{64} & \frac{1}{256} \\ \frac{11}{16} & \frac{15}{64} & \frac{15}{256} & \frac{1}{64} & \frac{1}{256} \\ \frac{11}{16} & \frac{15}{64} & \frac{15}{256} & \frac{1}{64} & \frac{1}{256} \\ \frac{11}{16} & \frac{15}{64} & \frac{15}{256} & \frac{1}{64} & \frac{1}{256} \\ \frac{11}{16} & \frac{15}{64} & \frac{15}{256} & \frac{1}{64} & \frac{1}{256} \end{bmatrix} & P^n &= P^4 \text{ for } n \geq 5 \end{aligned}$$

Usually  $P^n$  just approaches a limit as  $n$  increases, rather than reaching it. We'll see other examples later (like  $P^2$ ).

In this example, no matter what the starting state, the probabilities of being in states 1, 2, 3, 4, 5 at time  $t$  when  $t$  is large enough are  $\frac{11}{16}, \frac{15}{64}, \frac{15}{256}, \frac{1}{64}, \frac{1}{256}$ . (Usually these would be reached asymptotically as  $n \rightarrow \infty$  rather than exactly.)

Suppose at time  $t$ , the probability of being in state  $i$  is  $\alpha_i(t) = P(X_t = i)$ . (For example, sometimes the starting state is not known, but instead, the probabilities of each state at time  $t = 0$  are specified.) Consider the column vector

$$\vec{\alpha}(t) = \begin{pmatrix} \alpha_1(t) \\ \vdots \\ \alpha_s(t) \end{pmatrix} \quad \text{or make it a row vector by using the transpose} \\ \vec{\alpha}(t)' = (\alpha_1(t), \dots, \alpha_s(t))$$

The probabilities at time  $t + n$  are

$$\alpha_j(t+n) = P(X_{t+n} = j | \vec{\alpha}(t)) = \sum_i P(X_{t+n} = j | X_t = i) P(X_t = i) = \sum_i \alpha_i(t) (P^n)_{ij} = (\vec{\alpha}(t)' P^n)_j,$$

so  $\vec{\alpha}(t+n)' = \vec{\alpha}(t)' P^n$  (row vector times matrix), or equivalently,  $(P')^n \vec{\alpha}(t) = \vec{\alpha}(t+n)$  (matrix times column vector).

#### 4. STATIONARY DISTRIBUTION (A.K.A. STEADY STATE DISTRIBUTION)

If  $P$  is irreducible and aperiodic,  $P^n$  approaches a limit with this format as  $n \rightarrow \infty$ :

$$\lim_{n \rightarrow \infty} P^n = \begin{bmatrix} \varphi_1 & \varphi_2 & \cdots & \varphi_s \\ \varphi_1 & \varphi_2 & \cdots & \varphi_s \\ \cdots & \cdots & \cdots & \cdots \\ \varphi_1 & \varphi_2 & \cdots & \varphi_s \end{bmatrix}$$

In other words, no matter what the starting state, the probability of being in state  $j$  after  $n$  steps approaches  $\varphi_j$ . The row vector  $\vec{\varphi}' = (\varphi_1, \dots, \varphi_n)$  is called the *stationary distribution* of the Markov chain. It is “stationary” because these probabilities stay the same from one time to the next: in matrix notation,  $\vec{\varphi}'P = \vec{\varphi}'$ , or  $P'\vec{\varphi} = \vec{\varphi}$ . So  $\vec{\varphi}'$  is a left eigenvector of  $P$  with eigenvalue 1. Since it represents probabilities of being in each state, the components of  $\vec{\varphi}$  add up to 1.

A general method to compute  $\vec{\varphi}$  is to find a left eigenvector of  $P$  with eigenvalue 1 (or right eigenvector of  $P'$ ), and then divide by the sum of the elements of the vector. It turns out that in a Markov chain, all eigenvalues have absolute value  $\leq 1$ . If  $P$  is irreducible and aperiodic, there is just one eigenvalue equal to 1 and the others have absolute value  $\leq 1$ .

In the  $M1$  example: solve  $\vec{\varphi}'P = \vec{\varphi}'$ , or  $(\varphi_1, \dots, \varphi_5)P = (\varphi_1, \dots, \varphi_5)$ :

$$\begin{aligned} \varphi_1 &= \frac{3}{4}\varphi_1 + \frac{1}{2}\varphi_2 + \frac{3}{4}\varphi_3 + \frac{1}{2}\varphi_4 + \frac{3}{4}\varphi_5 \\ \varphi_2 &= \frac{1}{4}\varphi_1 + \frac{1}{4}\varphi_2 + 0\varphi_3 + \frac{1}{4}\varphi_4 + 0\varphi_5 \\ \varphi_3 &= 0\varphi_1 + \frac{1}{4}\varphi_2 + 0\varphi_3 + 0\varphi_4 + 0\varphi_5 \\ \varphi_4 &= 0\varphi_1 + 0\varphi_2 + \frac{1}{4}\varphi_3 + 0\varphi_4 + \frac{1}{4}\varphi_5 \\ \varphi_5 &= 0\varphi_1 + 0\varphi_2 + 0\varphi_3 + \frac{1}{4}\varphi_4 + 0\varphi_5 \end{aligned}$$

and put in the total probability equation  $\varphi_1 + \varphi_2 + \varphi_3 + \varphi_4 + \varphi_5 = 1$ . This is 6 equations in 5 unknowns, so it is overdetermined; knock out the  $\varphi_5 = \dots$  equation and solve the rest of them to get  $\vec{\varphi}' = (\frac{11}{16}, \frac{15}{64}, \frac{15}{256}, \frac{1}{64}, \frac{1}{256}) \approx (0.6875, 0.2344, 0.0586, 0.0156, 0.0039)$ .

(Actually, the first 5 equations are underdetermined; they add up to  $\varphi_1 + \dots + \varphi_5 = \varphi_1 + \dots + \varphi_5$ .)

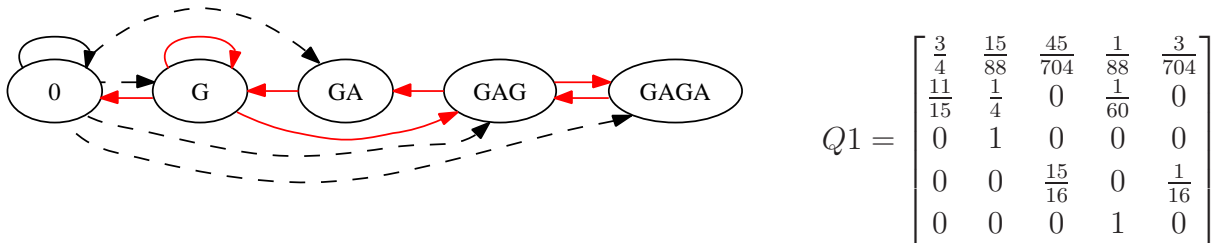
#### 5. REVERSE MARKOV CHAIN

Given a Markov chain that accurately models the “forwards” progression of time, a “reverse” version of it can be constructed to make predictions about the past. For example, this is done in models of nucleotide evolution. The graph of the reverse Markov chain has the same nodes as the forwards chain; the same edges but reverse directions; and new probabilities for these edges.

The transition matrix  $P$  of the forwards Markov chain was defined so that  $P(X_{t+1} = j | X_t = i) = p_{ij}$  at all times  $t$ . The reverse Markov chain has transition matrix  $Q$ , where

$$q_{ij} = P(X_t = j | X_{t+1} = i) = \frac{P(X_{t+1} = i | X_t = j)P(X_t = j)}{P(X_{t+1} = i)} = \frac{p_{ji}\varphi_j}{\varphi_i}$$

We assumed the forwards machine has run long enough to reach the stationary distribution,  $P(X_t = i) = \varphi_i$ . For  $M1$ , the graph and transition matrix of the reverse Markov chain are



For example, the edge  $0 \rightarrow \text{GA}$  in the reverse chain has  $q_{13} = p_{31}\varphi_3/\varphi_1 = (\frac{3}{4})(\frac{15}{256})/(\frac{11}{16}) = \frac{45}{704}$ . This means that in the steady state of the forwards chain, when 0 is entered, there is a probability  $\frac{45}{704}$  that the previous state was GA.

6. EXPECTED TIME FROM STATE  $i$  TILL NEXT TIME IN STATE  $j$ 

What is the expected number of steps from state  $i$  to the next time in state  $j$ ?

Fix  $j$ . For  $i = 1, \dots, s$ , let  $N_i$  be the number of steps from state  $i$  to the next time in state  $j$  (so if  $i = j$  we are counting until the next time at state  $j$ , and  $N_j \geq 1$ ). A system of equations for  $E(N_i)$  can be found as follows:

Start in state  $i$ . There is a probability  $P_{ir}$  of going one step to state  $r$ . If that is state  $j$  we are done, and otherwise the expected number of steps after that first step is  $E(N_r)$ , giving a total (including the first step) of  $E(N_r + 1) = E(N_r) + 1$ .

$$E(N_i) = P_{ij} \cdot 1 + \sum_{\substack{r=1 \\ r \neq j}}^s P_{ir} E(N_r + 1) = P_{ij} + \sum_{\substack{r=1 \\ r \neq j}}^s P_{ir} \cdot (E(N_r) + 1) = \sum_{r=1}^s P_{ir} + \sum_{\substack{r=1 \\ r \neq j}}^s P_{ir} \cdot E(N_r) = 1 + \sum_{\substack{r=1 \\ r \neq j}}^s P_{ir} \cdot E(N_r)$$

Doing this for all  $s$  states,  $i = 1, \dots, s$ , gives  $s$  equations in the  $s$  unknowns  $E(N_1), \dots, E(N_s)$ . For machine  $M1$ , the expected times until reaching the final state are found as follows:

$$\begin{aligned} E(N_1) &= 0 + \frac{3}{4}(E(N_1) + 1) + \frac{1}{4}(E(N_2) + 1) &&= 1 + \frac{3}{4}E(N_1) + \frac{1}{4}E(N_2) \\ E(N_2) &= 0 + \frac{1}{2}(E(N_1) + 1) + \frac{1}{4}(E(N_2) + 1) + \frac{1}{4}(E(N_3) + 1) &&= 1 + \frac{1}{2}E(N_1) + \frac{1}{4}E(N_2) + \frac{1}{4}E(N_3) \\ E(N_3) &= 0 + \frac{3}{4}(E(N_1) + 1) + \frac{1}{4}(E(N_4) + 1) &&= 1 + \frac{3}{4}E(N_1) + \frac{1}{4}E(N_4) \\ E(N_4) &= \frac{1}{4} + \frac{1}{2}(E(N_1) + 1) + \frac{1}{4}(E(N_2) + 1) &&= 1 + \frac{1}{2}E(N_1) + \frac{1}{4}E(N_2) \\ E(N_5) &= 0 + \frac{3}{4}(E(N_1) + 1) + \frac{1}{4}(E(N_4) + 1) &&= 1 + \frac{3}{4}E(N_1) + \frac{1}{4}E(N_4) \end{aligned}$$

This is five equations in five unknowns  $E(N_i)$ ,  $i = 1, 2, 3, 4, 5$ . In matrix format,

$$\begin{bmatrix} -1/4 & 1/4 & 0 & 0 & 0 \\ 1/2 & -3/4 & 1/4 & 0 & 0 \\ 3/4 & 0 & -1 & 1/4 & 0 \\ 1/2 & 1/4 & 0 & -1 & 0 \\ 3/4 & 0 & 0 & 1/4 & -1 \end{bmatrix} \begin{bmatrix} E(N_1) \\ E(N_2) \\ E(N_3) \\ E(N_4) \\ E(N_5) \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}$$

(General pattern on left side: Start with  $P$ . Zero out column  $j$ . Then subtract 1 from each diagonal entry. Right side: column vector of  $-1$ 's.)

Solving gives  $E(N_1) = 272$ ,  $E(N_2) = 268$ ,  $E(N_3) = 256$ ,  $E(N_4) = 204$ ,  $E(N_5) = 256$ .

We may compute  $E(g(N_i))$  for any function  $g$  by setting up recurrences in the same way:

$$E(g(N_i)) = P_{ij}g(1) + \sum_{r \neq j} P_{ir}E(g(N_r + 1)) = \text{expansion depending on } g$$

**6.1. Variance of  $N_i$ 's:** To compute  $\text{Var}(N_i) = E(N_i^2) - E(N_i)^2$ , we set up a system of equations for  $E(N_i^2)$ :

$$\begin{aligned} E(N_i^2) &= P_{ij} \cdot 1^2 + \sum_{\substack{r=1 \\ r \neq j}}^s P_{ir} E((N_r + 1)^2) = P_{ij} + \sum_{\substack{r=1 \\ r \neq j}}^s P_{ir} (E(N_r^2) + 2E(N_r) + 1) \\ &= \left( P_{ij} + \sum_{\substack{r=1 \\ r \neq j}}^s P_{ir} \cdot 1 \right) + 2 \sum_{\substack{r=1 \\ r \neq j}}^s P_{ir} \cdot E(N_r) + \sum_{\substack{r=1 \\ r \neq j}}^s P_{ir} \cdot E(N_r^2) \\ &= 1 + 2 \sum_{\substack{r=1 \\ r \neq j}}^s P_{ir} \cdot E(N_r) + \sum_{\substack{r=1 \\ r \neq j}}^s P_{ir} \cdot E(N_r^2) \end{aligned}$$

Solve first for  $E(N_1), \dots, E(N_s)$  and plug in. That leaves  $s$  unknowns,  $E(N_1^2), \dots, E(N_s^2)$ , and  $s$  equations, which could be solved.

**6.2. Complete distribution of  $N_i$ 's:** The complete distribution of  $N_i$  is encoded in its probability generating function  $E(x^{N_i}) = \sum_{n=0}^{\infty} P(N_i = n)x^n$ . We can solve for this through recursions:

$$E(x^{N_i}) = P_{ij} \cdot x^1 + \sum_{\substack{r=1 \\ r \neq j}}^s P_{ir} E(x^{N_r+1}) = P_{ij} \cdot x + \sum_{\substack{r=1 \\ r \neq j}}^s P_{ir} x E(x^{N_r}) \quad i = 1, \dots, s$$

Write  $f_i(x) = E(x^{N_i})$  to change this into

$$f_i(x) = P_{ij} \cdot x + \sum_{\substack{r=1 \\ r \neq j}}^s P_{ir} x f_r(x) = x \left( P_{ij} + \sum_{\substack{r=1 \\ r \neq j}}^s P_{ir} f_r(x) \right) \quad i = 1, \dots, s$$

For machine M1 and the expected times until reaching state  $j = 5$ , this gives equations

$$\begin{aligned} f_1(x) &= x \left( \frac{3}{4}f_1(x) + \frac{1}{4}f_2(x) + 0f_3(x) + 0f_4(x) + 0 \right) \\ f_2(x) &= x \left( \frac{1}{2}f_1(x) + \frac{1}{4}f_2(x) + \frac{1}{4}f_3(x) + 0f_4(x) + 0 \right) \\ f_3(x) &= x \left( \frac{3}{4}f_1(x) + 0f_2(x) + 0f_3(x) + \frac{1}{4}f_4(x) + 0 \right) \\ f_4(x) &= x \left( \frac{1}{2}f_1(x) + \frac{1}{4}f_2(x) + 0f_3(x) + 0f_4(x) + \frac{1}{4} \right) \\ f_5(x) &= x \left( \frac{3}{4}f_1(x) + 0f_2(x) + 0f_3(x) + \frac{1}{4}f_4(x) + 0 \right) \end{aligned}$$

which in matrix form is

$$\begin{bmatrix} \frac{3}{4}x - 1 & \frac{1}{4}x & 0 & 0 & 0 \\ \frac{1}{2}x & \frac{1}{4}x - 1 & \frac{1}{4}x & 0 & 0 \\ \frac{3}{4}x & 0 & -1 & \frac{1}{4}x & 0 \\ \frac{1}{2}x & \frac{1}{4}x & 0 & -1 & 0 \\ \frac{3}{4}x & 0 & 0 & \frac{1}{4}x & -1 \end{bmatrix} \begin{bmatrix} f_1(x) \\ f_2(x) \\ f_3(x) \\ f_4(x) \\ f_5(x) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\frac{1}{4}x \\ 0 \end{bmatrix}$$

Solving gives

$$\begin{aligned} f_1(x) &= x^4/D(x) \\ f_2(x) &= x^3(4 - 3t)/D(x) \\ f_3(x) &= x^2(x^2 - 16x + 16)/D(x) \\ f_4(x) &= x(-3x^3 + 4x^2 - 64x + 64)/D(x) \\ f_5(x) &= x^2(x^2 - 16x + 16)/D(x) \end{aligned}$$

where  $D(x) = x^4 - 16x^3 + 16x^2 - 256x + 256$ . (Since this involves symbolic computation instead of numerical computation, Maple or Mathematica are much better suited to solving this than Matlab.)

Several things can be done with these. We will just consider  $f_1(x)$ , giving the distribution of times from state 1 until state 5. The Taylor series of  $f_1(x)$  about  $t = 0$  can be computed:

$$f_1(x) = \frac{1}{256}x^4 + \frac{1}{256}x^5 + \frac{15}{4096}x^6 + \frac{15}{4096}x^7 + \frac{15}{4096}x^8 + \frac{239}{65536}x^9 + \dots$$

so for example,  $P(N_1 = 9) = 239/65536$ .

It's an irreducible Markov chain, so the probability of reaching state  $j$  is 1:  $\sum_{n=0}^{\infty} P(N_1 = i) = 1$ . Indeed, plugging in  $x = 1$  gives  $f_1(1) = 1$ .

The mean is  $\mu_1 = f_1'(1) = 272$  and the variance is  $f_1''(1) + \mu_1 - \mu_1^2 = 145856 + 272 - 272^2 = 72144$ , so the standard deviation is  $\sqrt{72144} \approx 268.596$ .

If it wasn't irreducible, state 5 might not ever be reached, so  $f_i(1)$  could be less than 1. Replace  $f_i(x)$  by  $g_i(x) = f_i(x)/f_i(1)$  to get a conditional pgf, mean, and variance (conditioned on eventually reaching state 5).

## 7. EXPECTED NUMBER OF VISITS TO A STATE BEFORE REACHING ANOTHER STATE

For  $M1$  or  $M2$ , starting at 0, what is the expected number of times **GAG** will be visited before the destination **GAGA** is first reached?

For  $M3$ , starting at 0, what is the expected number of times any particular intermediate word will be visited before either destination **GAGA** or **GTGA** is first reached?

Choose some destination states (and include all absorbing states in the list). Order all states so that destination states are last. Make a new machine  $\widetilde{M}$  by modifying  $M$  so that the destination states are turned into absorbing states, and call the transition matrix  $\widetilde{P}$ . We have block decompositions

$$P = \left[ \begin{array}{c|c} R & T \\ \hline S & U \end{array} \right] \quad \widetilde{P} = \left[ \begin{array}{c|c} R & T \\ \hline 0 & I \end{array} \right]$$

where  $R$  is the the submatrix of  $P$  with rows/columns of non-destination states. Row/column names in  $R, S, T$  are the same as in  $P$  even though some states are missing from each (i.e., do not renumber them to start at 1). If all states can reach some destination state, then some row sums of  $R$  are  $< 1$  and all eigenvalues of  $R$  have absolute value  $< 1$ .

Fix  $i$  and  $j$ . Start in state  $i$  at  $t = 0$ . In  $\widetilde{M}$ , let  $D_t$  be an indicator random variable,  $D_t = 1$  if  $X_t = j$ ,  $D_t = 0$  otherwise. The total number of visits to state  $j$  is  $N = \sum_t D_t$ , and

$$E(N) = \sum_{t=0}^{\infty} E(D_t) = \sum_{t=0}^{\infty} P(X_t = j | X_0 = i) = \sum_{t=0}^{\infty} (R^t)_{ij} = ((I - R)^{-1})_{ij} = H_{ij}$$

where  $H = (I - R)^{-1}$ . (We used  $R$  instead of  $P$  because  $P$  has one or more eigenvalues equal to 1, so  $I - P$  can't be inverted.  $R$  has all eigenvalues  $< 1$  in absolute value so the geometric series  $\sum_{t=0}^{\infty} R^t = (I - R)^{-1}$  converges.) For  $\widetilde{M1}$  and  $\widetilde{M2}$ , we have

$$R = \begin{bmatrix} 3/4 & 1/4 & 0 & 0 \\ 1/2 & 1/4 & 1/4 & 0 \\ 3/4 & 0 & 0 & 1/4 \\ 1/2 & 1/4 & 0 & 0 \end{bmatrix} \quad H = (I - R)^{-1} = \begin{bmatrix} 188 & 64 & 16 & 4 \\ 184 & 64 & 16 & 4 \\ 176 & 60 & 16 & 4 \\ 140 & 48 & 12 & 4 \end{bmatrix}$$

**Interpretation of first row of  $H$  (from state 0):** Starting in state 0 and counting until **GAGA** is first encountered, the expected number of visits to 0 (including being there to start) is 188, and the expected number of visits to **G**, **GA**, and **GAG**, respectively, are 64, 16, and 4.

In  $M1$  (original, not modified  $\widetilde{M1}$ ), we could also ask for the expected number of visits to state  $j$  between two visits to **GAGA**. In  $M$  (unmodified), if  $k$  is a destination state and  $j$  is a non-destination state, the expected number of visits to  $j$  after leaving  $k$  and before entering a destination state again, is  $(SH)_{kj}$ . For  $M1$ ,  $SH = [3/4 \ 0 \ 0 \ 1/4]H = [176 \ 60 \ 15 \ 4]$ . Notice it's almost the 3rd row of  $H$  except for the 3rd entry because **GA** and **GAGA** have the same outgoing edges in  $M1$ ;  $H_{33}$  includes 1 for starting at **GA** while  $(SH)_{53}$  only counts later visits to **GA**.

Back to  $\widetilde{M}$ . The probability of ending in absorbing state  $k$  is  $(HT)_{ik}$ . This is obtained by considering the time  $t$  of the last state  $j$  before  $k$ , where  $j$  runs over non-absorbing states:

$$\begin{aligned} P(X_{\infty} = k | X_0 = i) &= \sum_{t=0}^{\infty} \sum_j P(X_t = j | X_0 = i) P(X_{t+1} = k | X_t = j) \\ &= \sum_{t=0}^{\infty} \sum_j (R^t)_{ij} T_{jk} = \sum_{t=0}^{\infty} (R^t T)_{ik} = (HT)_{ik} \end{aligned}$$

$$\text{For } \widetilde{M1}: \quad HT = \begin{bmatrix} 188 & 64 & 16 & 4 \\ 184 & 64 & 16 & 4 \\ 176 & 60 & 16 & 4 \\ 140 & 48 & 12 & 4 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$



so starting from any of the first 4 states, the probability of reaching **GAGA** is 1. For  $\widetilde{M3}$  with 6 non-absorbing and 2 absorbing states,  $HT$  would be  $6 \times 2$  and the row sums would all equal 1.

We previously computed  $E(N)$ . For the variance of  $N$ , we need

$$\begin{aligned}
E(N^2) &= \sum_t E((D_t)^2) + 2 \sum_{t_1=0}^{\infty} \sum_{t_2=t_1+1}^{\infty} E(D_{t_1} D_{t_2}) \\
&= \sum_t E(D_t) + 2 \sum_{t_1=0}^{\infty} \sum_{t_2=t_1+1}^{\infty} P(X_{t_1} = j | X_0 = i) P(X_{t_2} = j | X_{t_1} = j) \\
&= E(N) + 2 \sum_{t_1=0}^{\infty} \sum_{u_2=1}^{\infty} (R^{t_1})_{ij} (R^{u_2})_{jj} \quad \text{where } u_2 = t_2 - t_1 > 0 \\
&= E(N) + 2 \left( \sum_{t_1=0}^{\infty} R^{t_1} \right)_{ij} \left( \sum_{u_2=1}^{\infty} R^{u_2} \right)_{jj} \\
&= H_{ij} + 2H_{ij}(R(I - R)^{-1})_{jj} = H_{ij}(1 + 2(H_{jj} - 1)) = H_{ij}(2H_{jj} - 1)
\end{aligned}$$

so  $\text{Var}(N) = E(N^2) - (E(N))^2 = H_{ij}(2H_{jj} - 1) - (H_{ij})^2$ . Let  $N_{ij}$  be the number of visits to  $j$  before absorption if we start in state  $i$ , where  $i, j$  are non-absorbing states. Then in matrix form,

$$[\text{Var}(N_{ij})] = \begin{bmatrix} 35156 & 4032 & 240 & 12 \\ 35144 & 4032 & 240 & 12 \\ 35024 & 4020 & 240 & 12 \\ 32900 & 3792 & 228 & 12 \end{bmatrix} \quad [\sigma(N_{ij})] = \left[ \sqrt{\text{Var}(N_{ij})} \right] \approx \begin{bmatrix} 187.50 & 63.50 & 15.49 & 3.46 \\ 187.47 & 63.50 & 15.49 & 3.46 \\ 187.15 & 63.40 & 15.49 & 3.46 \\ 181.38 & 61.58 & 15.10 & 3.46 \end{bmatrix}$$

## 8. POWERS OF MATRICES

While  $P1^n$  became constant for  $n \geq 4$ , this is not usually the case. The first several powers of  $P = P2$ , to 6 decimal digits, are

$$\begin{aligned}
P^1 &= \begin{bmatrix} 0.750000 & 0.250000 & 0.000000 & 0.000000 & 0.000000 \\ 0.500000 & 0.250000 & 0.250000 & 0.000000 & 0.000000 \\ 0.750000 & 0.000000 & 0.000000 & 0.250000 & 0.000000 \\ 0.500000 & 0.250000 & 0.000000 & 0.000000 & 0.250000 \\ 0.750000 & 0.250000 & 0.000000 & 0.000000 & 0.000000 \end{bmatrix}, \quad P^2 = \begin{bmatrix} 0.687500 & 0.250000 & 0.0625000 & 0.0000000 & 0.0000000 \\ 0.687500 & 0.187500 & 0.0625000 & 0.0625000 & 0.0000000 \\ 0.687500 & 0.250000 & 0.0000000 & 0.0000000 & 0.0625000 \\ 0.687500 & 0.250000 & 0.0625000 & 0.0000000 & 0.0000000 \\ 0.687500 & 0.250000 & 0.0625000 & 0.0000000 & 0.0000000 \end{bmatrix}, \\
P^3 &= \begin{bmatrix} 0.687500 & 0.234375 & 0.0625000 & 0.0156250 & 0.0000000 \\ 0.687500 & 0.234375 & 0.0468750 & 0.0156250 & 0.0156250 \\ 0.687500 & 0.250000 & 0.0625000 & 0.0000000 & 0.0000000 \\ 0.687500 & 0.234375 & 0.0625000 & 0.0156250 & 0.0000000 \\ 0.687500 & 0.234375 & 0.0625000 & 0.0156250 & 0.0000000 \end{bmatrix}, \quad P^4 = \begin{bmatrix} 0.687500 & 0.234375 & 0.0585938 & 0.0156250 & 0.00390625 \\ 0.687500 & 0.238281 & 0.0585938 & 0.0117188 & 0.00390625 \\ 0.687500 & 0.234375 & 0.0625000 & 0.0156250 & 0.00000000 \\ 0.687500 & 0.234375 & 0.0585938 & 0.0156250 & 0.00390625 \\ 0.687500 & 0.234375 & 0.0585938 & 0.0156250 & 0.00390625 \end{bmatrix}, \\
P^5 &= \begin{bmatrix} 0.687500 & 0.235352 & 0.0585938 & 0.0146484 & 0.00390625 \\ 0.687500 & 0.235352 & 0.0595703 & 0.0146484 & 0.00292969 \\ 0.687500 & 0.234375 & 0.0585938 & 0.0156250 & 0.00390625 \\ 0.687500 & 0.235352 & 0.0585938 & 0.0146484 & 0.00390625 \\ 0.687500 & 0.235352 & 0.0585938 & 0.0146484 & 0.00390625 \end{bmatrix}, \quad P^6 = \begin{bmatrix} 0.687500 & 0.235352 & 0.0588379 & 0.0146484 & 0.00366211 \\ 0.687500 & 0.235107 & 0.0588379 & 0.0148926 & 0.00366211 \\ 0.687500 & 0.235352 & 0.0585938 & 0.0146484 & 0.00390625 \\ 0.687500 & 0.235352 & 0.0588379 & 0.0146484 & 0.00366211 \\ 0.687500 & 0.235352 & 0.0588379 & 0.0146484 & 0.00366211 \end{bmatrix}, \dots
\end{aligned}$$

As  $n \rightarrow \infty$ , the rows all approach the stationary distribution

$$\vec{\varphi}' = \left( \frac{11}{16}, \frac{4}{17}, \frac{1}{17}, \frac{1}{68}, \frac{1}{272} \right) \approx (0.687500, 0.235294, 0.0588235, 0.0147059, 0.00367647).$$

**8.1. Diagonalizable matrices.** Suppose an  $s \times s$  matrix  $P$  has distinct eigenvalues  $\lambda_1, \dots, \lambda_s$ .

Usually some of these are complex numbers, so we will use  $'$  to denote the ‘‘adjoint’’ instead of the ‘‘transpose’’ of a matrix or vector: it is transpose combined with complex conjugate.

Then  $P$  can be written in the format  $P = VDV^{-1}$  where  $D$  is the diagonal matrix

$$D = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 & 0 \\ 0 & \lambda_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \lambda_{s-1} & 0 \\ 0 & 0 & \cdots & 0 & \lambda_s \end{bmatrix}$$

and column  $j$  of  $V$  is a (right) eigenvector of  $P$ ,  $\vec{r}_j$ , with eigenvalue  $\lambda_j$ :  $P\vec{r}_j = \lambda_j\vec{r}_j$ . It turns out that row  $j$  of  $V^{-1}$  is a left eigenvector,  $\vec{\ell}'_j$ , of  $P$  with eigenvalue  $\lambda_j$ :  $\vec{\ell}'_j P = \lambda_j\vec{\ell}'_j$ .

Then  $P^2 = VDV^{-1}VDV^{-1} = VD^2V^{-1}$ , and in general,  $P^n = VD^nV^{-1}$ . Since  $D$  is diagonal,  $D^n$  is easy to compute:

$$D^n = \begin{bmatrix} \lambda_1^n & 0 & \cdots & 0 & 0 \\ 0 & \lambda_2^n & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \lambda_{s-1}^n & 0 \\ 0 & 0 & \cdots & 0 & \lambda_s^n \end{bmatrix}$$

A shortcut to compute  $P^n = VD^nV^{-1}$  is to use the *spectral decomposition*:

$$P^n = \sum_{j=1}^s \lambda_j^n \vec{r}_j \vec{\ell}'_j \quad \text{where } \vec{r}_j \vec{\ell}'_j \text{ is column vector times row vector} \\ = s \times s \text{ square matrix}$$

**8.2. Jordan Canonical Form.** Only some matrices can be diagonalized. There is a generalization of the factorization  $P = VDV^{-1}$  to any matrix. Any matrix  $P$ , including one with repeated eigenvalues, can be written in the format  $P = VJV^{-1}$ , where  $J$  is mostly zeros except for the eigenvalues of  $P$  along its diagonal and some 1's just above the diagonal according to a certain rule (below). Then  $P^n = VJ^nV^{-1}$  and the powers  $J^n$  are easy to compute. Since the Jordan Canonical Form includes diagonalizable matrices as a special case, the best approach is to compute the JCF in the first place, instead of trying to diagonalize and possibly failing to do so. This is the format:

$$J = \begin{bmatrix} B_1 & 0 & 0 & \cdots \\ 0 & B_2 & 0 & \cdots \\ 0 & 0 & B_3 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad \text{and each } B_i \text{ is a block matrix, } B_i = \begin{bmatrix} \lambda_i & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \lambda_i & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \lambda_i & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \lambda_i & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & \lambda_i \end{bmatrix}$$

The dimensions of the  $B_i$ 's are related to multiplicities of eigenvalues. Then  $P^n = VJ^nV^{-1}$ , where

$$J^n = \begin{bmatrix} B_1^n & 0 & 0 & \cdots \\ 0 & B_2^n & 0 & \cdots \\ 0 & 0 & B_3^n & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad B_i^n = \begin{bmatrix} \lambda_i^n & \binom{n}{1}\lambda_i^{n-1} & \binom{n}{2}\lambda_i^{n-2} & \cdots & \cdots & \cdots \\ 0 & \lambda_i^n & \binom{n}{1}\lambda_i^{n-1} & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & \lambda_i^n & \binom{n}{1}\lambda_i^{n-1} \\ 0 & 0 & 0 & \cdots & 0 & \lambda_i^n \end{bmatrix}$$

where a row of  $B_i^n$  with fewer than  $n$  cells right of the diagonal would have the extra binomial terms dropped, and a row with more than  $n$  cells right of the diagonal would be padded with 0's.

The columns of  $V$  (or rows of  $V^{-1}$ ) are "generalized eigenvectors." An ordinary (right) eigenvector satisfies  $P\vec{v} = \lambda\vec{v}$ , or  $(P - \lambda I)\vec{v} = \vec{0}$ . The generalized ones satisfy  $(P - \lambda I)^k\vec{v} = \vec{0}$  for various  $k$ 's.

Although the word "Canonical" is used, the factorization  $P = VJV^{-1}$  for any matrix (or  $P = VDV^{-1}$  for diagonalizable matrices) is not unique, so different programs and different people's hand computations can return different but correct answers. See a linear algebra book for details.

**8.3. Eigenvalues.** In a Markov chain, the absolute values of the eigenvalues are between 0 and 1, and there is at least one eigenvalue equal to 1 (right eigenvector:  $\vec{1}$ ; left eigenvector: stationary distribution). For an irreducible aperiodic Markov chain, there is only one eigenvalue equal to 1 and the others have smaller absolute value. So in  $P^n = VD^nV^{-1}$  or  $P^n = VJ^nV^{-1}$ , as  $n \rightarrow \infty$ , the diagonal term for eigenvalue 1 stays equal to 1, and the others go to 0, making  $P^n$  approach a constant limit. The second largest  $|\lambda_j|$  determines how fast  $P^n$  converges to the stationary distribution, since its  $n$ th powers give the second largest term.

For an irreducible periodic Markov chain with period  $d$ , there are  $d$  eigenvalues equal to the  $d$ th roots of unity,  $1, e^{2\pi i/d}, e^{4\pi i/d}, e^{6\pi i/d}, \dots, e^{2(d-1)\pi i/d}$ . As  $n \rightarrow \infty$ , instead of a stationary distribution, the Markov chain cycles through  $d$  different distributions.

For a reducible Markov chain on  $s$  states, there are eigenvalues contributed from each component, as well as additional ones to bring it up to  $s$  eigenvalues in total (counting multiplicities).

So machine  $M4$  has an eigenvalue 1 for the component with state 3; an eigenvalue 1 and an eigenvalue  $-1$  for the component with  $\{4, 5, 6, 7\}$  (since it has period 2); eigenvalues  $1, e^{2\pi i/3}, e^{4\pi i/3}$  for the component with  $\{8, 9, 10\}$ ; and 4 additional eigenvalues of absolute value less than 1, so that there are 10 eigenvalues in all, counting multiplicities. At least two of those additional eigenvalues equal 0, because  $P^2$  has columns 1 and 2 equal to 0 (no 2-step transitions to states 1 or 2). The two remaining eigenvalues depend on the probabilities of the edges, which were not fully specified. Here is one possibility:

$$P = \begin{bmatrix} 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} = VJV^{-1}$$

$$V = \begin{bmatrix} 1 & 0 & \frac{1}{3} & \frac{1}{3} & -\frac{1}{3} & 0 & -\frac{2}{3} & \frac{1}{3} & \frac{1}{3}e^{-2\pi i/3} & \frac{1}{3}e^{2\pi i/3} \\ 0 & 3 & 1 & 0 & 0 & -2 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & -2 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & e^{2\pi i/3} & e^{-2\pi i/3} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & e^{-2\pi i/3} & e^{2\pi i/3} \end{bmatrix}, J = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{-2\pi i/3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{2\pi i/3} \end{bmatrix}$$

$$V^{-1} = \begin{bmatrix} 1 & 0 & -\frac{1}{3} & 0 & 0 & 0 & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} \\ 0 & \frac{1}{3} & -\frac{1}{3} & \frac{1}{6} & 0 & -\frac{1}{6} & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{4} & 0 & -\frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & -\frac{1}{4} & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3}e^{-2\pi i/3} & \frac{1}{3}e^{2\pi i/3} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3}e^{2\pi i/3} & \frac{1}{3}e^{-2\pi i/3} \end{bmatrix}$$

## 9. MATLAB EXAMPLES

9.1. **Example 1.** Here is a stochastic matrix and some of its powers. (Stochastic because its entries are  $\geq 0$  and its row sums are 1. This one is actually doubly stochastic because the column sums are also 1.)

```
>> M = [[.5, .25, .25]; [.25, .5, .25]; [.25, .25, .5]]
```

```
M =
```

```
0.5000    0.2500    0.2500
0.2500    0.5000    0.2500
0.2500    0.2500    0.5000
```

```
>> M^2
```

```
ans =
0.3750    0.3125    0.3125
0.3125    0.3750    0.3125
0.3125    0.3125    0.3750
```

```
>> M^3
```

```
ans =
0.3438    0.3281    0.3281
0.3281    0.3438    0.3281
0.3281    0.3281    0.3438
```

```
>> M^10
```

```
ans =
0.3333    0.3333    0.3333
0.3333    0.3333    0.3333
0.3333    0.3333    0.3333
```

**Eigenvalues and eigenvectors:** Matlab's `eig(M)` returns  $[V,D]$ . Diagonal elements of  $D$  are eigenvalues. Columns of  $V$  are right eigenvectors *if  $M$  is diagonalizable*, but some columns are garbage if it's not. There is no warning, so you have to check.

```
>> [V,D] = eig(M)
```

```
V =
```

```
0.4082    0.7071    0.5774
0.4082   -0.7071    0.5774
-0.8165     0      0.5774
```

```
D =
```

```
0.2500     0     0
0     0.2500     0
0     0     1.0000
```

$M$  has three eigenvalues:  $\lambda_1 = .25$ ,  $\lambda_2 = .25$ ,  $\lambda_3 = 1$ . Two of them are equal so it is possible that  $M$  is not diagonalizable. Verify that  $M = VDV^{-1}$ :

```
>> V * D * inv(V)
```

```
ans =
```

```
0.5000    0.2500    0.2500
0.2500    0.5000    0.2500
0.2500    0.2500    0.5000
```

So  $M$  is diagonalizable.

The right eigenvectors of  $M$  corresponding to these eigenvalues are the columns of  $V$  in the exact same order as in  $D$ :

$$\vec{r}_1 = \begin{bmatrix} 0.4082 \\ 0.4082 \\ -0.8165 \end{bmatrix} \quad \vec{r}_2 = \begin{bmatrix} 0.7071 \\ -0.7071 \\ 0 \end{bmatrix} \quad \vec{r}_3 = \begin{bmatrix} 0.5774 \\ 0.5774 \\ 0.5774 \end{bmatrix}$$

Remember, any scalar multiple of an eigenvector can be used in place of it, so a different program (or doing it by hand) would potentially return a different result, such as

$$\vec{r}_1 = \begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix} \quad \vec{r}_2 = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} \quad \vec{r}_3 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Also, the eigenvalues could be in a different order on the diagonal of  $D$ , in which case the order of the eigenvectors (as columns of  $V$  and rows of  $V^{-1}$ ) would have to change to match that.

Let's verify that  $\vec{r}_1$  is an eigenvector with eigenvalue  $\lambda_1$ :

```

>> V(:,1)
ans =
    0.4082
    0.4082
   -0.8165

>> D(1,1)
ans =
    0.2500

>> M * V(:,1)
ans =
    0.1021
    0.1021
   -0.2041

>> D(1,1) * V(:,1)
ans =
    0.1021
    0.1021
   -0.2041

>> right1 = V(:,1)
right1 =
    0.4082
    0.4082
   -0.8165

>> lambda1 = D(1,1)
lambda1 =
    0.2500

>> M * right1
ans =
    0.1021
    0.1021
   -0.2041

>> lambda1 * right1
ans =
    0.1021
    0.1021
   -0.2041

```

Next look at  $V^{-1}$  and the left eigenvectors.

```

>> Vi = inv(V)
Vi =
    0.4082    0.4082   -0.8165
    0.7071   -0.7071    0.0000
    0.5774    0.5774    0.5774

```

The rows of  $V^{-1}$  (denoted  $V_i$ ) are the left eigenvectors of  $M$ :

$$\vec{\ell}_1 = (0.4082, 0.4082, -0.8165) \quad \vec{\ell}_2 = (0.7071, -0.7071, 0) \quad \vec{\ell}_3 = (0.5774, 0.5774, 0.5774)$$

Again, they can be rescaled, so by hand you would probably have used

$$\vec{\ell}_1 = (1, 1, -2) \quad \vec{\ell}_2 = (1, -1, 0) \quad \vec{\ell}_3 = (1, 1, 1)$$

Verify  $\vec{\ell}_1$  is a left eigenvector:

```

>> Vi(1,:)
ans =
    0.4082    0.4082   -0.8165

>> Vi(1,:) * M
ans =
    0.1021    0.1021   -0.2041

>> D(1,1) * Vi(1,:)
ans =
    0.1021    0.1021   -0.2041

```

**Stationary distribution:** We need to normalize the left eigenvector whose eigenvalue equals 1. In this case it was  $\lambda_3 = 1$ , so we need to take  $\vec{\ell}'_3$ , and normalize to make its components add to 1; do this by dividing by the sum of its entries:

```

>> Vi(3,:)
ans =
    0.5774    0.5774    0.5774

>> sstate = Vi(3,:) / sum(Vi(3,:))
sstate =
    0.3333    0.3333    0.3333

```

**Spectral decomposition:**  $S1 = \vec{r}_1 \vec{\ell}'_1$ ,  $S2 = \vec{r}_2 \vec{\ell}'_2$ ,  $S3 = \vec{r}_3 \vec{\ell}'_3$ , and  $M^n = \lambda_1^n S1 + \lambda_2^n S2 + \lambda_3^n S3 = .25^n(S1 + S2) + 1^n S3$  which approaches  $S3$  as  $n$  increases,

```
>> S1 = V(:,1) * Vi(1,:)
S1 =
    0.1667    0.1667   -0.3333
    0.1667    0.1667   -0.3333
   -0.3333   -0.3333    0.6667

>> S2 = V(:,2) * Vi(2,:)
S2 =
    0.5000   -0.5000    0.0000
   -0.5000    0.5000   -0.0000
         0         0         0

>> S3 = V(:,3) * Vi(3,:)
S3 =
    0.3333    0.3333    0.3333
    0.3333    0.3333    0.3333
    0.3333    0.3333    0.3333

>> S1+S2
ans =
    0.6667   -0.3333   -0.3333
   -0.3333    0.6667   -0.3333
   -0.3333   -0.3333    0.6667
```

Note that the rows of  $S3$  all equal the stationary distribution. (However, it's a coincidence that the columns do also.) Now we'll verify the spectral decomposition formula using  $M^3$ :

```
>> lambda1 = D(1,1)
lambda1 =
    0.2500

>> lambda2 = D(2,2)
lambda2 =
    0.2500

>> lambda3 = D(3,3)
lambda3 =
    1.0000

>> M^3
ans =
    0.3438    0.3281    0.3281
    0.3281    0.3438    0.3281
    0.3281    0.3281    0.3438

>> lambda1^3*S1 + lambda2^3*S2 + lambda3^3*S3
ans =
    0.3437    0.3281    0.3281
    0.3281    0.3437    0.3281
    0.3281    0.3281    0.3438
```

## 9.2. Example 2: Complex Numbers.

```
>> x = [1+2i,3+4i,5+6i]
```

```
x =
    1.0000 + 2.0000i    3.0000 + 4.0000i    5.0000 + 6.0000i
```

Note that ' is adjoint (conjugate and transpose combined):

```
>> x'
ans =
    1.0000 - 2.0000i
    3.0000 - 4.0000i
    5.0000 - 6.0000i
```

```
>> real(x)
ans =
     1     3     5
```

```
>> imag(x)
ans =
     2     4     6
```

```
>> conj(x)
ans =
    1.0000 - 2.0000i    3.0000 - 4.0000i    5.0000 - 6.0000i
```

### 9.3. Example 3: The matrix $P2$ for machine $M2$ : Complex Eigenvalues and Eigenvectors, and Jordan Canonical Form.

```
>> P2 = [
  [ 3/4, 1/4, 0, 0, 0 ]; %
  [ 2/4, 1/4, 1/4, 0, 0 ]; % G
  [ 3/4, 0, 0, 1/4, 0 ]; % GA
  [ 2/4, 1/4, 0, 0, 1/4 ]; % GAG
  [ 3/4, 1/4, 0, 0, 0 ]; % GAGA
]
```

```
P2 =
    0.7500    0.2500         0         0         0
    0.5000    0.2500    0.2500         0         0
    0.7500         0         0    0.2500         0
    0.5000    0.2500         0         0    0.2500
    0.7500    0.2500         0         0         0
```

#### Eigenvalues and eigenvectors, first try:

```
>> [V2,D2] = eig(P2)
```

```
V2 =
    0.4472         0.2182 + 0.0000i    0.2182 - 0.0000i    0.0000 + 0.1857i    0.0000 - 0.1857i
    0.4472        -0.6547 - 0.0000i   -0.6547 + 0.0000i   -0.1857 - 0.5571i   -0.1857 + 0.5571i
    0.4472         0.2182 - 0.0000i    0.2182 + 0.0000i    0.7428              0.7428
    0.4472        -0.6547              -0.6547              0.0000 + 0.1857i    0.0000 - 0.1857i
    0.4472         0.2182 - 0.0000i    0.2182 + 0.0000i   -0.0000 + 0.1857i   -0.0000 - 0.1857i
```

```
D2 =
    1.0000         0         0         0         0
         0        -0.0000 + 0.0000i         0         0         0
         0         0        -0.0000 - 0.0000i         0         0
         0         0         0    0.0000 + 0.2500i         0
         0         0         0         0    0.0000 - 0.2500i
```

Notice the eigenvalues: 1 corresponds to the right eigenvector  $\vec{1}$  that's a column of 1's, and to the left eigenvector that's the stationary distribution. There are two 0's so  $P2$  has a duplicate eigenvalue and we will have to use Jordan Canonical form. The interpretation of one of them is that two rows of  $P2$  are equal, the top and bottom row, so  $(1, 0, 0, 0, -1)'$  (' turns row vector into column vector) is a right eigenvector of  $P2$  with eigenvalue 0. There are also two complex eigenvalues,  $0 \pm i/4$ , which come in a conjugate pair (they must since  $P2$  is real), and their corresponding eigenvectors (columns 4 and 5 of  $V2$ ) also come in a complex conjugate pair.

$V2$  is not invertible; it's determinant should be 0, but Matlab has a numerical precision error:

```
>> det(V2)
ans =
-4.5552e-009 +1.1903e-017i
```

So we redo this with the Jordan Canonical Form.

```
>> [V2,J2] = jordan(P2)
```

```
V2 =
   -0.0625        -0.5170        -0.1728         0.1176 + 0.0294i    0.1176 - 0.0294i
    0.1875         1.3011        -0.1728        -0.3824 + 0.0294i   -0.3824 - 0.0294i
   -0.0625         0.4830        -0.1728         0.1176 - 0.4706i    0.1176 + 0.4706i
    0.1875         1.3011        -0.1728         0.1176 + 0.0294i    0.1176 - 0.0294i
   -0.0625         0.4830        -0.1728         0.1176 + 0.0294i    0.1176 - 0.0294i
```

```
J2 =
         0         1.0000         0         0         0
         0         0         0         0         0
         0         0         1.0000         0         0
         0         0         0         0 + 0.2500i         0
         0         0         0         0         0 - 0.2500i
```

**Inverse of corrected V2 to get left eigenvectors:**

```
>> V2i = inv(V2)
V2i =
  3.2727         0        -0.0000         4.0000        -7.2727
 -1.0000         0         0.0000         0          1.0000
 -3.9787       -1.3617       -0.3404       -0.0851       -0.0213
  0          -1.0000         0 + 1.0000i     1.0000         0 - 1.0000i
  0          -1.0000         0 - 1.0000i     1.0000         0 + 1.0000i
```

**Stationary distribution:** The eigenvalue equal to 1 is the third one ( $\lambda_3 = 1$ ), so the stationary distribution is row 3 of  $V2i = V2^{-1}$ , suitably normalized:

```
>> V2i(3,:) / sum(V2i(3,:))
ans =
  0.6875   0.2353   0.0588   0.0147   0.0037
```

This agrees with the computation stated earlier in this document,  $\vec{\varphi}' = \left(\frac{11}{16}, \frac{4}{17}, \frac{1}{17}, \frac{1}{68}, \frac{1}{272}\right) \approx (0.687500, 0.235294, 0.0588235, 0.0147059, 0.00367647)$ . It also agrees with a high power of P2:

```
>> P2^10
ans =
  0.6875   0.2353   0.0588   0.0147   0.0037
  0.6875   0.2353   0.0588   0.0147   0.0037
  0.6875   0.2353   0.0588   0.0147   0.0037
  0.6875   0.2353   0.0588   0.0147   0.0037
  0.6875   0.2353   0.0588   0.0147   0.0037
```

**Spectral decomposition:** Since P2 wasn't diagonalizable, the Jordan Canonical Form must be used and the spectral form gets harder. The general formula  $P = VJV^{-1}$  gives  $P = VJ^nV^{-1}$ . Here,  $P = P2$  and  $J = J2$ . The powers of J2 will be (fill in 0's for blanks):

$$(J2)^n = \begin{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}^n & & & \\ & 1^n & & \\ & & (i/4)^n & \\ & & & (-i/4)^n \end{bmatrix} \quad \begin{aligned} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}^0 &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}^1 &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}^n &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \end{aligned} \quad \text{for } n \geq 2$$

By writing  $(J2)^n$  as a sum of the four blocks shown, this gives a block expansion (in terms of left and right eigenvectors;  $\vec{r}_j$  is column  $j$  of V2,  $\vec{\ell}'_j$  is row  $j$  of  $V2^{-1}$ )

$$(P2)^n = (V2)(J2)^n(V2)^{-1} = [\vec{r}_1 \quad \vec{r}_2] \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}^n \begin{bmatrix} \vec{\ell}'_1 \\ \vec{\ell}'_2 \end{bmatrix} + \vec{r}_3(1)^n\vec{\ell}'_3 + \vec{r}_4(i/4)^n\vec{\ell}'_4 + \vec{r}_5(-i/4)^n\vec{\ell}'_5$$

When  $n \geq 2$ , the first term vanishes, so for  $n \geq 2$ , we will have  $(P2)^n = 1^nS3 + (i/4)^nS4 + (-i/4)^nS5 = S3 + (i/4)^nS4 + (-i/4)^nS5$ , where

```
>> S3 = V2(:,3) * V2i(3,:)
S3 =
  0.6875   0.2353   0.0588   0.0147   0.0037
  0.6875   0.2353   0.0588   0.0147   0.0037
  0.6875   0.2353   0.0588   0.0147   0.0037
  0.6875   0.2353   0.0588   0.0147   0.0037
  0.6875   0.2353   0.0588   0.0147   0.0037

>> S4 = V2(:,4) * V2i(4,:)
S4 =
  0          -0.1176 - 0.0294i  -0.0294 + 0.1176i   0.1176 + 0.0294i   0.0294 - 0.1176i
  0          0.3824 - 0.0294i  -0.0294 - 0.3824i  -0.3824 + 0.0294i   0.0294 + 0.3824i
  0          -0.1176 + 0.4706i   0.4706 + 0.1176i   0.1176 - 0.4706i   -0.4706 - 0.1176i
  0          -0.1176 - 0.0294i  -0.0294 + 0.1176i   0.1176 + 0.0294i   0.0294 - 0.1176i
  0          -0.1176 - 0.0294i  -0.0294 + 0.1176i   0.1176 + 0.0294i   0.0294 - 0.1176i
```



```
>> S5 = V2(:,5) * V2i(5,:)
S5 =
    0          -0.1176 + 0.0294i  -0.0294 - 0.1176i   0.1176 - 0.0294i   0.0294 + 0.1176i
    0          0.3824 + 0.0294i  -0.0294 + 0.3824i  -0.3824 - 0.0294i   0.0294 - 0.3824i
    0          -0.1176 - 0.4706i   0.4706 - 0.1176i   0.1176 + 0.4706i  -0.4706 + 0.1176i
    0          -0.1176 + 0.0294i  -0.0294 - 0.1176i   0.1176 - 0.0294i   0.0294 + 0.1176i
    0          -0.1176 + 0.0294i  -0.0294 - 0.1176i   0.1176 - 0.0294i   0.0294 + 0.1176i
```

$S3$  corresponds to the stationary distribution.  $S4$  and  $S5$  are complex conjugates, so  $(i/4)^n S4 + (-i/4)^n S5$  is a sum of two complex conjugates; thus, it is real-valued, even though complex numbers are involved in the computation. Now verify the formula:

```
>> P2^3
ans =
    0.6875    0.2344    0.0625    0.0156         0
    0.6875    0.2344    0.0469    0.0156    0.0156
    0.6875    0.2500    0.0625         0         0
    0.6875    0.2344    0.0625    0.0156         0
    0.6875    0.2344    0.0625    0.0156         0

>> 1^3*S3 + (i/4)^3*S4 + (-i/4)^3*S5
ans =
    0.6875    0.2344    0.0625    0.0156    0.0000
    0.6875    0.2344    0.0469    0.0156    0.0156
    0.6875    0.2500    0.0625   -0.0000    0.0000
    0.6875    0.2344    0.0625    0.0156    0.0000
    0.6875    0.2344    0.0625    0.0156    0.0000
```

#### 9.4. Example 4: The matrix $P1$ for machine $M1$ : Jordan Canonical Form.

```
>> P1 = [
    [ 3/4, 1/4, 0, 0, 0 ]; %
    [ 2/4, 1/4, 1/4, 0, 0 ]; % G
    [ 3/4, 0, 0, 1/4, 0 ]; % GA
    [ 2/4, 1/4, 0, 0, 1/4 ]; % GAG
    [ 3/4, 0, 0, 1/4, 0 ]; % GAGA
]

P1 =
    0.7500    0.2500         0         0         0
    0.5000    0.2500    0.2500         0         0
    0.7500         0         0    0.2500         0
    0.5000    0.2500         0         0    0.2500
    0.7500         0         0    0.2500         0

>> [V1,D1] = eig(P1)
V1 =
    0.4472         0.2182 + 0.0000i   0.2182 - 0.0000i  -0.2182 - 0.0000i  -0.2182 + 0.0000i
    0.4472        -0.6547         -0.6547         0.6547         0.6547
    0.4472         0.2182 - 0.0000i   0.2182 + 0.0000i  -0.2182 + 0.0000i  -0.2182 - 0.0000i
    0.4472        -0.6547 - 0.0000i  -0.6547 + 0.0000i   0.6547 - 0.0000i   0.6547 + 0.0000i
    0.4472         0.2182 - 0.0000i   0.2182 + 0.0000i  -0.2182 + 0.0000i  -0.2182 - 0.0000i

D1 =
    1.0000         0         0         0         0
         0    0.0000 + 0.0000i         0         0         0
         0         0    0.0000 - 0.0000i         0         0
         0         0         0    0.0000 + 0.0000i         0
         0         0         0         0    0.0000 - 0.0000i

>> imag(D1)
ans =
    1.0e-08 *
         0         0         0         0         0
         0    0.3596         0         0         0
         0         0   -0.3596         0         0
         0         0         0    0.1803         0
         0         0         0         0   -0.1803
```

```
>> imag(V1)
ans =
  1.0e-07 *

    0    0.0105   -0.0105   -0.0052    0.0052
    0         0         0         0         0
    0   -0.1151    0.1151    0.0577   -0.0577
    0   -0.0000    0.0000   -0.0000    0.0000
    0   -0.1151    0.1151    0.0577   -0.0577

>> V1 * D1 * inv(V1)
Warning: Matrix is close to singular or badly scaled.
Results may be inaccurate. RCOND = 3.355158e-25.
```

```
ans =
  1.0e+07 *

  0.0000 + 0.0000i  0.0000 - 0.3549i  0.0000 + 0.7099i  0.0000 + 0.3549i  -0.0000 - 0.7099i
  0.0000 + 0.0000i  0.0000 - 0.9693i  0.0000 + 1.9386i  0.0000 + 0.9693i  -0.0000 - 1.9386i
  0.0000 + 0.0000i  0.0000 - 0.3549i  0.0000 + 0.7099i  0.0000 + 0.3549i  -0.0000 - 0.7099i
  0.0000 + 0.0000i  0.0000 - 0.9693i  0.0000 + 1.9386i  -0.0000 + 0.9693i  -0.0000 - 1.9386i
  0.0000 + 0.0000i  0.0000 - 0.3549i  0.0000 + 0.7099i  0.0000 + 0.3549i  -0.0000 - 0.7099i
```

We see that there is one eigenvalue equal to 1 and four eigenvalues equal to 0, and that  $V1$  is not invertible and  $P1$  is not diagonalizable. Use the Jordan Canonical Form instead:

```
>> [V1,J1] = jordan(P1)
V1 =
 -0.0039   -0.0195   -0.0707   -0.2298   -0.0430
  0.0117    0.0430    0.1339    0.4066   -0.0430
 -0.0039    0.0430    0.1793    0.5884   -0.0430
  0.0117    0.0430    0.3839    1.4066   -0.0430
 -0.0039    0.0430    0.1793    1.5884   -0.0430
```

```
J1 =
  0    1    0    0    0
  0    0    1    0    0
  0    0    0    1    0
  0    0    0    0    0
  0    0    0    0    1
```

```
>> V1i = inv(V1)
V1i =
    0   52.3636  -64.0000   11.6364    0
 -16.0000  16.0000  13.0909  -16.0000   2.9091
    0   -4.0000   4.0000   4.0000  -4.0000
    0         0  -1.0000    0   1.0000
 -16.0000  -5.4545  -1.3636  -0.3636  -0.0909
```

```
>> V1 * J1 * V1i
ans =
  0.7500    0.2500   -0.0000   -0.0000    0.0000
  0.5000    0.2500    0.2500   -0.0000    0.0000
  0.7500   -0.0000   -0.0000    0.2500   -0.0000
  0.5000    0.2500    0.0000   -0.0000    0.2500
  0.7500   -0.0000   -0.0000    0.2500   -0.0000
```

**Eigenvalues and eigenvectors:** The eigenvalues are on the diagonal of  $J1$ : 0, 0, 0, 0, 1. The right eigenvector for eigenvalue  $\lambda_5 = 1$  is the 5th column of  $V1$ :

```
>> V1(:,5)
ans =
 -0.0430
 -0.0430
 -0.0430
 -0.0430
 -0.0430
```

which you would probably rescale to all 1's.

Since the first 4 eigenvalues  $\lambda_1 = \dots = \lambda_4 = 0$  are in a  $4 \times 4$  block of  $J1$  with 1's above the diagonal, the first of them corresponds to a right eigenvector

```
>> V1(:,1)
ans =
-0.0039
 0.0117
-0.0039
 0.0117
-0.0039
```

and the other three ( $\lambda_2 = \lambda_3 = \lambda_4 = 0$ ) correspond to “generalized eigenvectors.”

### Stationary distribution:

```
>> sstate = V1i(5,:) / sum(V1i(5,:))
sstate =
 0.6875  0.2344  0.0586  0.0156  0.0039
```

This agrees with the computation from the text:

$$\vec{\varphi}' = \left( \frac{11}{16}, \frac{15}{64}, \frac{15}{256}, \frac{1}{64}, \frac{1}{256} \right) \approx (0.6875, 0.2344, 0.0586, 0.0156, 0.0039).$$

**Powers of P1:** The “spectral decomposition” formula as given was only for diagonalizable matrices. For a Jordan Canonical Form,  $P = VJV^{-1}$  gives  $P^n = VJ^nV^{-1}$ , and  $J^n$  is easy to compute:

<pre>&gt;&gt; J1 J1 =  0  1  0  0  0  0  0  1  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  1</pre>	<pre>&gt;&gt; J1^2 ans =  0  0  1  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1</pre>	<pre>&gt;&gt; J1^3 ans =  0  0  0  1  0  1</pre>
<pre>&gt;&gt; J1^4 ans =  0  1</pre>	<pre>&gt;&gt; J1^5 ans =  0  1</pre>	

### Reverse Markov Chain:

```
>> d_sstate = diag(sstate)
d_sstate =
 0.6875  0  0  0  0
 0  0.2344  0  0  0
 0  0  0.0586  0  0
 0  0  0  0.0156  0
 0  0  0  0  0.0039

>> Q1 = inv(d_sstate) * P1' * d_sstate
Q1 =
 0.7500  0.1705  0.0639  0.0114  0.0043
 0.7333  0.2500  0  0.0167  0
 0  1.0000  0  0  0
 0  0  0.9375  0  0.0625
 0  0  0  1.0000  0
```

**Solving for  $E(N_i)$ :** We had a system of equations  $C\vec{x} = \vec{r}$ . Enter the coefficient matrix  $C$  and the vector  $\mathbf{r}$  and use  $C \setminus \mathbf{r}$  to solve it. You could enter the matrices  $C$  and  $\mathbf{r}$  directly, but I will show how to get  $C$  directly from  $P$  for the case considered in this handout ( $P$  is  $5 \times 5$  and we want the expected time until state 5); the rule to construct  $C$  from  $P$  was to zero out column  $j$  (here  $j = 5$ ) of  $P$  and then subtract 1 from each diagonal entry, and the rule to construct  $\mathbf{r}$  is that it’s a column vector of  $-1$ ’s.

```

>> C = P1
C =
    0.7500    0.2500         0         0         0
    0.5000    0.2500    0.2500         0         0
    0.7500         0         0    0.2500         0
    0.5000    0.2500         0         0    0.2500
    0.7500         0         0    0.2500         0

>> C(:,5) = 0           % we want time until state 5, so zero out the 5th column
C =
    0.7500    0.2500         0         0         0
    0.5000    0.2500    0.2500         0         0
    0.7500         0         0    0.2500         0
    0.5000    0.2500         0         0         0
    0.7500         0         0    0.2500         0

>> eye(5)           % gives a 5 x 5 identity matrix
ans =
     1     0     0     0     0
     0     1     0     0     0
     0     0     1     0     0
     0     0     0     1     0
     0     0     0     0     1

>> C = C - eye(5)   % C is a 5 x 5 matrix
C =
   -0.2500    0.2500         0         0         0
    0.5000   -0.7500    0.2500         0         0
    0.7500         0   -1.0000    0.2500         0
    0.5000    0.2500         0   -1.0000         0
    0.7500         0         0    0.2500   -1.0000

>> rhs = -1 * ones(5,1)
rhs =
   -1
   -1
   -1
   -1
   -1

>> C \ rhs
ans =
  272.0000
  268.0000
  256.0000
  204.0000
  256.0000

```

**Expected number of visits to each state before reaching GAGA:**

```

>> R1 = P1(1:4,1:4)
R =
    0.7500    0.2500         0         0
    0.5000    0.2500    0.2500         0
    0.7500         0         0    0.2500
    0.5000    0.2500         0         0

>> H1 = inv(eye(4)-R1)
H1 =
  188.0000   64.0000   16.0000    4.0000
  184.0000   64.0000   16.0000    4.0000
  176.0000   60.0000   16.0000    4.0000
  140.0000   48.0000   12.0000    4.0000

```

**Variance of that:** In Matlab, `diag(matrix)=vector`, `diag(vector)=matrix`, and  $m \cdot k$  is elementwise  $k$ th powers instead of the matrix  $k$ th power.

```

>> diag(H1)
ans =
  188.0000
   64.0000
   16.0000
    4.0000

```

```

>> diag(diag(H1))
ans =
 188.0000     0         0         0
     0    64.0000     0         0
     0         0    16.0000     0
     0         0         0    4.0000

>> VarN1 = H1 * (2*diag(diag(H1))-eye(4)) - (H1 .^ 2)
VarN1 =
 1.0e+04 *

    3.5156    0.4032    0.0240    0.0012
    3.5144    0.4032    0.0240    0.0012
    3.5024    0.4020    0.0240    0.0012
    3.2900    0.3792    0.0228    0.0012

>> SigmaN1 = VarN1 .^ (1/2)
SigmaN1 =
 187.4993    63.4980    15.4919    3.4641
 187.4673    63.4980    15.4919    3.4641
 187.1470    63.4035    15.4919    3.4641
 181.3836    61.5792    15.0997    3.4641

```

### Probability of eventually reaching GAGA from each state:

```

>> T1 = P1(1:4,5)
T1 =
     0
     0
     0
    0.2500

>> H1 * T1
ans =
    1.0000
    1.0000
    1.0000
    1.0000

```

10. MAPLE EXAMPLE: PGF FOR COMPLETE DISTRIBUTION OF  $N_i$ 'S

Here is how to use Maple to solve the system of equations for  $f_1(x), \dots, f_5(x)$  from Sec. 6.2 in this handout; and then get the pgf for  $N_1$  by computing the Taylor series of  $f_1(x)$ ; and then compute total probability for  $N_1$  ( $f_1(1) = 1$ ), the mean  $E(N_1) = f_1'(1)$ , and variance  $\text{Var}(N_1) = f_1''(1) + f_1'(1) - (f_1'(1))^2$ :

```
> sols :=
  solve({f1(x)=x*(3/4*f1(x) + 1/4*f2(x)),
        f2(x)=x*(1/2*f1(x) + 1/4*f2(x) + 1/4*f3(x)),
        f3(x)=x*(3/4*f1(x) + 1/4*f4(x)),
        f4(x)=x*(1/2*f1(x) + 1/4*f2(x) + 1/4),
        f5(x)=x*(3/4*f1(x) + 1/4*f4(x))},
        {f1(x),f2(x),f3(x),f4(x),f5(x)});
```

$$\text{sols} := \{f_1(x) = \frac{x^4}{x^4}, f_2(x) = -\frac{x^3(-4+3x)}{x^4}, f_3(x) = \frac{x^2(x^2+16-16x)}{x^4},$$

$$f_4(x) = -\frac{x^2(-4x^2+3x^3-64+64x)}{x^4}, f_5(x) = \frac{x^2(x^2+16-16x)}{x^4}\}$$

```
%1 := 16 x^2 - 16 x^3 + 256 - 256 x + x^4
```

```
> g1 := sols[1];
```

$$g_1 := f_1(x) = \frac{x^4}{16x^2 - 16x^3 + 256 - 256x + x^4}$$

```
> g1 := rhs(sols[1]);
```

$$g_1 := \frac{x^4}{16x^2 - 16x^3 + 256 - 256x + x^4}$$

```
> taylor(g1,x=0,10);
```

$$\frac{1}{256}x^4 + \frac{1}{256}x^5 + \frac{15}{4096}x^6 + \frac{15}{4096}x^7 + \frac{15}{4096}x^8 + \frac{239}{65536}x^9 + O(x^{10})$$

```
> N1_totalprob := subs(x=1,g1);
      N1_totalprob := 1
```

```
> diff(g1,x);
```

$$-\frac{x^4(32x^2 - 48x^3 - 256 + 4x^4)}{(16x^2 - 16x^3 + 256 - 256x + x^4)^2} + \frac{4x^3}{(16x^2 - 16x^3 + 256 - 256x + x^4)^3}$$

```
> N1_mean := subs(x=1,diff(g1,x));
      N1_mean := 272
```

```
> N1_var := subs(x=1,diff(g1,x,x)) + N1_mean - N1_mean^2;
      N1_var := 72144
```