Stochastic Calculus Notes, Lecture 2

Last modified September 16, 2004

1 Forward and Backward Equations for Markov chains

1.1. Introduction: Forward and backward equations are useful ways to get answers to quantitative questions about Markov chains. The probabilities $u(k,t) = P(X(t) = k)$ satisfy forward equations that allows us to compute all the numbers $u(k, t+1)$ once the all the numbers $u(j,t)$ are known. This moves us forward from time t to time $t+1$. The expected values $f(k,t) = E[V(X(T)) \mid X(t) = k]$ (for $t < T$) satisfy a backward equation that allows us to calculate the numbers $f(k,t)$ once all the $f(j, t+1)$ are known. A *duality* relation allows us to infer the forward equation from the backward equation, or conversely. The transition matrix is the *generator* of both equations, though in different ways. There are many related problems that have solutions involving forward and backward equations. Two treated here are hitting probabilities and random compound interest.

1.2. Forward equation, functional version: Let $u(k,t) = P(X(t) = k)$. The law of total probability gives

$$\begin{aligned} u(k,t+1) &= P(X(t+1) = k) \\ &= \sum_j P(X(t+1) = k \mid X(t) = j) \cdot P(X(t) = j) \; . \end{aligned}$$

Therefore

$$u(k,t+1) = \sum_j P_{jk} u(j,t) \; . \qquad (1)$$

This is the *forward equation* for probabilities. It is also called the Kolmogorov forward equation or the Chapman Kolmogorov equation. Once $u(j,t)$ is known for all $j \in \mathcal{S}$, (1) gives $u(k,t+1)$ for any k. Thus, we can go forward in time from $t = 0$ to $t = 1$, etc. and calculate all the numbers $u(k,t)$.

Note that if we just wanted one number, say $u(17, 49)$, still we would have to calculate many related quantities, all the $u(j,t)$ for $t < 49$. If the state space is too large, this direct forward equation approach may be impractical.

1.3. Row and column vectors: If A is an $n \times m$ matrix, and B is an $m \times p$ matrix, then AB is $n \times p$. The matrices are compatible for multiplication because the second dimension of A, the number of columns, matches the first dimension of B, the number of rows. A matrix with just one column is a *column vector*.[1]

[1] The physicists' more sophisticated idea that a vector is a physical quantity with certain transformation properties is "inoperative" here.

Just one row makes it a *row vector*. Matrix-vector multiplication is a special case of matrix-matrix multiplication. We often denote genuine matrices (more than one row and column) with capital letters and vectors, row or column, with lower case. In particular, if u is an n dimensional row vector, a $1 \times n$ matrix, and A is an $n \times n$ matrix, then uA is another n dimensional row vector. We do not write Au for this because that would be incompatible. Matrix multiplication is always associative. For example, if u is a row vector and A and B are square matrices, then $(uA)B = u(AB)$. We can compute the row vector uA then multiply by B, or we can compute the $n \times n$ matrix AB then multiply by u.

If u is a row vector, we usually denote the k-th entry by u_k instead of u_{1k}. Similarly, the k-th entry of column vector f is f_k instead of f_{k1}. If both u and f have n components, then $uf = \sum_{k=1}^{n} u_k f_k$ is a 1×1 matrix, i.e. a number. Thus, treating row and column vectors as special kinds of matrices makes the product of a row with a column vector natural, but not, for example, the product of two column vectors.

1.4. Forward equation, matrix version: The probabilities $u(k,t)$ form the components of a row vector, $u(t)$, with components $u_k(t) = u(k,t)$ (an abuse of notation). The forward equation (1) may be expressed (check this)

$$u(t+1) = u(t)P \; . \tag{2}$$

Because matrix multiplication is associative, we have

$$u(t) = u(t-1)P = u(t-2)P^2 = \cdots = u(0)P^t \; . \tag{3}$$

Tricks of matrix multiplication give information about the evolution of probabilities. For example, we can write a formula for $u(t)$ in terms of the eigenvectors and eigenvalues of P. Also, we can save effort in computing $u(t)$ for large t by repeated squaring:

$$P \to P^2 \to \left(P^2\right)^2 = P^4 \to \cdots \to P^{2^k}$$

using just k matrix multiplications. For example, this computes P^{1024} using just ten matrix multiplies, instead of a thousand.

1.5. Backward equation, functional version: Suppose we run the Markov chain until time T then get a "reward", $V(X(T))$. For $t \leq T$, define the conditional expectations

$$f(k,t) = E\left[V(X(T)) \mid X(t) = k\right] \; . \tag{4}$$

This expression is used so often it often is abbreviated

$$f(k,t) = E_{k,t}[V(X(T))] \; .$$

These satisfy a *backward equation* that follows from the law of total probability:

$$f(k,t) \quad = \quad E\left[V(X(T)) \mid X(t) = k\right]$$

$$
\begin{aligned}
&= \sum_{j\in\mathcal{S}} E\left[V(X(T)) \mid X(t)=k \text{ and } X(t+1)=j\right]\cdot P(X(t+1)=j \mid X(t)=k) \\
f(k,t) &= \sum_{j\in\mathcal{S}} f(j,t+1)P_{kj}\ . \qquad (5)
\end{aligned}
$$

The Markov property is used to infer that

$$E[V(X(T)) \mid X(t)=k \text{ and } X(t+1)=j] = E_{j,t+1}[V(X(T))]\ .$$

The dynamics (5) must be supplemented with the *final condition*

$$f(k,T) = V(k)\ . \qquad (6)$$

Using these, we may compute all the numbers $f(k,T-1)$, then all the numbers $f(k,T-2)$, etc.

1.6. Backward equation using modern conditional expectation: As usual, $\mathcal{F}_t$ denotes the $\sigma-$algebra generated by $X(0)$, …, $X(t)$. Define $F(t) = E[V(X(T)) \mid \mathcal{F}_t]$. The left side is a random variable that is measurable in $\mathcal{F}_t$, which means that $F(t)$ is a function of $(X(0),\ldots,X(t))$. The Markov property implies that $F(t)$ actually is measurable with respect to $\mathcal{G}_t$, the $\sigma-$algebra generated by $X(t)$ alone. This means that $F(t)$ is a function of $X(t)$ alone, which is to say that there is a function $f(k,t)$ so that $F(t) = f(X(t),t)$, and

$$f(X(t),t) = E[V(X(T)) \mid \mathcal{F}_t] = E[V(X(T)) \mid \mathcal{G}_t]\ .$$

Since $\mathcal{G}_t$ is generated by the partition $\{k\} = \{X(t)=k\}$, this is the same definition (4). Moreover, because $\mathcal{F}_t \subseteq \mathcal{F}_{t+1}$ and $F(t+1) = E[V(X(T)) \mid \mathcal{F}_{t+1}]$, the tower property gives

$$E[V(X(T)) \mid \mathcal{F}_t] = E[F(t+1) \mid \mathcal{F}_t]\ ,$$

so that, again using the Markov property,

$$F(t) = E[F(t+1) \mid \mathcal{G}_t]\ . \qquad (7)$$

Note that this is a version of the tower property. On the event $\{X(t)=k\}$, the right side above takes the value

$$\sum_{j\in\mathcal{S}} f(j,t+1)\cdot P(x(t+1)=j \mid X(t)=k)\ .$$

Thus, (7) is the same as the backward equation (5). In the continuous time versions to come, (7) will be very handy.

1.7. Backward equation, matrix version: We organize the numbers $f(k,t)$ into a column vector $f(t) = (f(1,t), f(2,t),\cdots)^t$. It is barely an abuse to write $f(t)$ both for a function of k and a vector. After all, any computer programmer knows that a vector really is a function of the index. The backward equation (5) then is equivalent to (check this)

$$f(t) = Pf(t+1) \; . \tag{8}$$

Again the associativity of matrix multiplication lets us write, for example,

$$f(t) = P^{T-t} V \; ,$$

writing V for the vector of values of V.

1.8. Invariant expectation value: We combine the conditional expectations (4) with the probabilities $u(k,t)$ with the law of total probability to get, for any t,

$$\begin{aligned} E[V(X(T))] &= \sum_{k \in \mathcal{S}} P(X(t) = k) \cdot E[V(X(T)) \mid X(t) = k] \\ &= \sum_{k \in \mathcal{S}} u(k,t) f(k,t) \\ &= u(t) f(t) \; . \end{aligned}$$

The last line is a natural example of an inner product between a row vector and a column vector. Note that the product $E[V(X(T))] = u(t)f(t)$ does not depend on t even though $u(t)$ and $f(t)$ are different for different t. For this *invariance* to be possible, the forward evolution equation for u and the backward equation for f must be related.

1.9. Relationship between the forward and backward equations: It often is possible to derive the backward equation from the forward equation and conversely using the invariance of $u(t)f(t)$. For example, suppose we know that $f(t) = Pf(t+1)$. Then $u(t+1)f(t+1) = u(t)f(t)$ may be rewritten $u(t+1)f(t+1) = u(t)Pf(t+1)$, which may be rearranged as (using rules of matrix multiplication)

$$(\; u(t+1) - u(t)P \;)\, f(t+1) = 0 \; .$$

If this is true for enough linearly independent vectors $f(t+1)$, then the vector $u(t+1) - u(t)P$ must be zero, which is the matrix version of the forward equation (2). A theoretically minded reader can verify that enough f vectors are produced if the transition matrix is nonsingular and we choose a linearly independent family of "reward" vectors, V. In the same way, the backward evolution of f is a consequence of invariance and the forward evolution of u.

We now have two ways to evaluate $E[V(X(T))]$: (i) start with given $u(0)$, compute $u(T) = u(0)P^T$, evaluate $u(T)V$, or (ii) start with given $V = f(T)$, compute $f(0) = P^T V$, then evaluate $u(0)f(0)$. The former might be preferable, for example, if we had a number a number of different reward functions to evaluate. We could compute $u(T)$ once then evalute $u(T)V$ for all our V vectors.

1.10. Duality: In it's simplest form, *duality* is the relationship between a matrix and its transpose. The set of column vectors with n components is a vector space of dimension n. The set of n component row vectors is the *dual space*, which has the same dimension but may be considered to be a different space. We can combine an element of a vector space with an element of its dual to get a number: row vector u multiplied by column vector f yields the number uf. Any linear transformation on the vector space of column vectors is represented by an $n \times n$ matrix, P. This matrix also defines a linear transformation, the *dual transformation*, on the dual space of row vectors, given by $u \to uP$. This is the sense in which the forward and backward equations are dual to each other.

Some people prefer not to use row vectors and instead think of organizing the probabilities $u(k,t)$ into a column vector that is the transpose of what we called $u(t)$. For them, the forward equation would be written $u(t+1) = P^t u(t)$ (note the notational problem: the t in P^t means "transpose" while the t in $u(t)$ and $f(t)$ refers to time.). The invariance relation for them would be $u^t(t+1)f(t+1) = u^t(t)f(t)$. The transpose of a matrix is often called its *dual*.

1.11. Hitting probabilities, backwards: The hitting probability for state 1 up to time T is

$$P\left(X(t) = 1 \text{ for some } t \in [0,T]\right) \ . \tag{9}$$

Here and below we write $[a,b]$ for all the *integers* between a and b, including a and/or b if they are integers. Hitting probabilities can be computed using forward or backward equations, often by modifying P and adding *boundary conditions*. For one backward equation approach, define

$$f(k,t) = P\left(X(t') = 1 \text{ for some } t' \in [t,T] \mid X(t) = k\right) \ . \tag{10}$$

Clearly,

$$f(1,t) = 1 \text{ for all } t, \tag{11}$$

and

$$f(k,T) = 0 \text{ for } k \neq 1. \tag{12}$$

Moreover, if $k \neq 1$, the law of total probabilities yields a backward relation

$$f(k,t) = \sum_{j \in \mathcal{S}} P_{kj} f(j,t+1) \ . \tag{13}$$

The difference between this and the plain backward equation (5) is that the relation (13) holds only for *interior* states $k \neq 1$, while the boundary condition (11) supplies the values of $f(1,t)$. The sum on the right of (13) includes the term corresponding to state $j = 1$.

1.12. Hitting probabilities, forward: We also can compute the hitting probabilities (9) using a forward equation approach. Define the *survival probabilities*

$$u(k,t) = P\left(X(t) = k \text{ and } X(t') \neq 1 \text{ for } t' \in [0,t]\right) \ . \tag{14}$$

These satisfy the obvious *boundary condition*

$$u(1,t) = 0 \ , \tag{15}$$

and initial condition

$$u(k,0) = 1 \text{ for } k \neq 1. \tag{16}$$

The forward equation is (as the reader should check)

$$u(k,t+1) = \sum_{j\in\mathcal{S}} u(j,t)P_{jk} \ . \tag{17}$$

We may include or exclude the term with $j = 1$ on the right because $u(1,t) = 0$. Of course, (17) applies only at interior states $k \neq 1$. The overall probability of survival up to time T is $\sum_{k\in\mathcal{S}} u(k,T)$ and the hitting probability is the complementary $1 - \sum_{k\in\mathcal{S}} u(k,T)$.

The matrix vector formulation of this involves the row vector

$$\widetilde{u}(t) = (u(2,t), u(3,t), \ldots)$$

and the matrix $\widetilde{P}$ formed from P by removing the first row and column. The evolution equation (17) and boundary condition (15) are both expressed by the matrix equation

$$\widetilde{u}(t+1) = \widetilde{u}(t)\widetilde{P} \ .$$

Note that $\widetilde{P}$ is not a stochastic matrix because some of the row sums are less than one:

$$\sum_{j\neq 1} P_{kj} < 1 \quad \text{if} \quad P_{k1} > 0 \ .$$

1.13. Absorbing boundaries: *Absorbing boundaries* are another way to think about hitting and survival probabilities. The absorbing boundary Markov chain is the same as the original chain (same transition probabilities) as long as the state is not one of the boundary states. In the absorbing chain, the state never again changes after it visits an absorbing boundary point. If $\overline{P}$ is the transition matrix of the absorbing chain and P is the original transition matrix, this means that $\overline{P}_{jk} = P_{jk}$ if j is not a boundary state, while $\overline{P}_{jk} = 0$ if j is a boundary state and $k \neq j$. The probabilities $u(k,t)$ for the absorbing chain are the same as the survival probabilities (14) for the original chain.

1.14. Running cost: Suppose we have a *running cost* functtion, $W(x)$, and we want to calculate

$$f = E\left[\sum_{t=0}^{T} W(X(t))\right] \ . \tag{18}$$

Sums like this are called *path dependent* because their value depends on the whole path, not just the final value $X(T)$. We can calculate (18) with the forward equation using

$$\begin{aligned} f &= \sum_{t=0}^{T} E\left[W(X(t))\right] \\ &= \sum_{t=0}^{T} u(t)W \ . \end{aligned} \tag{19}$$

Here W is the column vector with components $W_k = W(k)$. We compute the probabilities that are the components of the $u(t)$ using the standard forward equation (2) and sum the products (19).

One backward equation approach uses the quantities

$$f(k,t) = E_{k,t}\left[\sum_{t'=t}^{T} W(X(t'))\right] \ . \tag{20}$$

These satisfy (check this):

$$f(t) = Pf(t+1) + W \ . \tag{21}$$

Starting with $f(T) = W$, we work backwards with (21) until we reach the desired $f(0)$.

1.15. Multiplicitive functionals: For some reason, a function of a function is often called a *functional*. The path, $X(t)$, is a function of t, so a function, $F(X)$, that depends on the whole path is often called a functional. Some applications call for finding the expected value of a multiplicative functional:

$$f = E\left[\prod_{t=0}^{T} V(X(t))\right] \ . \tag{22}$$

For example, $X(t)$ could represent the state of a financial market and $V(k) = 1 + r(k)$ the interest rate for state k. Then (22) would be the expected total interest. We also can write $V(k) = e^{W(k)}$, so that

$$\prod V(X(t)) = \exp\left[\sum W(X(t))\right] = e^{Z} \ ,$$

with $Z = \sum W(x(t))$. This dos not solve the problem of evaluating (22) because $E\left[e^{z}\right] \neq e^{E(Z)}$.

The backward equation approach uses the intermediate quantities

$$f(k,t) = E_{k,t}\left[\prod_{t'=t}^{T} V(X(t'))\right] \ .$$

The $t' = t$ term in the product has $V(X(t)) = V(k)$. The final condition is $f(k,T) = V(k)$. The backward evolution equation is derived more or less as before:

$$\begin{aligned}
f(k,t) &= E_{k,t}\left[V(k)\prod_{t'>t} V(X(t'))\right] \\
&= V(k)E_{k,t}\left[\prod_{t'=t+1}^{T} V(X(t'))\right] \\
&= V(k)E_{k,t}\left[f(X(t+1),t+1)\right] \quad \text{(the tower property)} \\
f(k,t) &= V(k)\big(Pf(t+1)\big)(k)\ . \qquad (23)
\end{aligned}$$

In the last line on the right, $f(t+1)$ is teh column vector with components $f(k,t+1)$ and $Pf(t+1)$ is teh matrix vector product. We write $\big(Pf(t+1)\big)(k)$ for the k^{th} component of the column vector $Pf(t+1)$. We could express the whole thing in matrix terms using diag(V), the diagonal matrix with $V(k)$ in the (k,k) position:

$$f(t) = \mathrm{diag}(V)Pf(t+1)\ .$$

A version of (23) for Brownian motion is called the Feynman-Kac formula.

1.16. Branching processes: One forward equation approach to (22) leads to a different interpretation of the answer. Let $B(k,t)$ be the event $\{X(t)=k\}$ and $I(k,t)$ the indicator function of $B(k,t)$. That is $I(k,t,X)=1$ if $X\in B(k,t)$ (i.e. $X(t)=k$), and $I(k,t,X)=0$ otherwise. It is in keeping with the probabilists' habbit of leaving out the arguents of functions when the argument is the underlying random outcome. We have $u(k,t)=E[I(k,t)]$. The forward equation for the quantities

$$g(k,t) = E\left[I(k,t)\prod_{t'=0}^{t} V(X(t'))\right] \qquad (24)$$

is (see homework):

$$g(k,t) = V(k)\big(g(t-1)P\big)(k)\ . \qquad (25)$$

This is also the forward equation for a *branching process* with branching factors $V(k)$. At time t, the branching process has $N(k,t)$ *particles*, or *walkers*, at state k. The numbers $N(k,t)$ are random. A time step of the branching process has two parts. First, each particle takes one step of the Markov chain. A particle at state j goes to state k with probability P_{jk}. All steps for all particles are independent. Then, each particle at state k does a *branching* or *birth/death* step in which the particle is replaced by a random number of particles with expected number $V(k)$. For example, if $V(k)=1/2$, we could delete the particle (death) with probability half. If $V(k)=2.8$, we could keep the existing particle, one new one, then add a third with probability .8. All particles are treated independently. If there are m particles in state k before the birth/death step, the expected number after the birth/death step is $V(k)m$. The expected number of particles, $g(k,t)=E[N(k,t)]$, satisfies (25).

When $V(k) = 1$ for all k there need be no birth or death. There will be just one particle, the path $X(t)$. The number of particles at state k at time t, $N(k,t)$, will be zero if $X(t) \neq k$ or one if $X(t) = k$. In fact, $N(k,t) = I(k,t)(X)$. The expected values will be $g(k,t) = E[N(k,t)] = E[I(k,t)] = u(k,t)$.

The branching process representation of (22) is possible when $V(k) \geq 0$ for all k. Monte Carlo methods based on branching processes are more accurate than direct Monte Carlo in many cases.

2 Lattices, trees, and random walk

2.1. Introduction: Random walk on a lattice is an important example where the abstract theory of Markov chains is used. It is the simplest model of something randomly moving through space with none of the subtlty of Brownian motion, though random walk on a lattice is a useful approximation to Brownian motion, and vice versa. The forward and backward equations take a specific simple form for lattice random walk and it is often possible to calculate or approximate the solutions by hand. Boundary conditions will be applied at the boundaries of lattices, hence the name.

We pursue forward and backward equations for several reasons. First, they often are the best way to calculate expectations and hitting probabilities. Second, many theoretical qualitative properties of specific Markov chains are understood using backward or forward equatins. Third, they help explain and motivate the partial differential equations that arise as backward and forward equations for diffusion processes.

2.2. Simple random walk: The state space for simple random walk is the integers, positive and negative. At each time, the walker has three choices: (A) move up one, (B) do not move, (C) move down one. The probabilities are $P(A) = P(k \to k+1) = a$, $P(B) = P(X(t+1) = X(t)) = b$, and $P(X(t+1) = X(t)-1) = c$. Naturally, we need a, b, and c to be non-negative and $a+b+c = 1$. The transation matrix[2] has b on the diagonal ($P_{kk} = b$ for all k), a on the *super-diagonal* ($P_{k,k+1} = a$ for all k), and c on the *sub diagonal*. All other matrix elements P_{jk} are zero.

This Markov chain is *homogeneous* or *translation invariant*: The probalities of moving up or down are independent of $X(t)$. A *translation* by k is a shift of everything by k (I do not know why this is called "translation"). Translation invariance means, for example, that the probability of going from m to l in s steps is the same as the probability of going from $m+k$ to $l+k$ in s steps: $P(X(t+s) = l \mid X(t) = m) = P(X(t+s) = l+k \mid X(t) = m+k)$. It is common to simplify general discussions by choosing k so that $X(0) = 0$. Mathematicians often say "without loss of generality" or "w.l.o.g." when doing so.

[2]This "matrix" is infinite when the state space is infinite. Matrix multiplication is still defined. For example, the k component of uP is given by $(uP)_k = \sum_j u_j P_{jk}$. This possibly infinite sum has only three nonzero terms when P is tridiagonal.

Often, particularly when discussing multidimensional random walk, we use x, y, etc. instead of j, k, etc. to denote lattice points (states of the Markov chain). Probabilists often use lower case Latin letters for general possible values of a random variable, while using the capital letter for the random variable itself. Thus, we might write $P_{xy} = P(X(t+1) = x \mid X(t) = y)$. As an execise in definition unwrapping, review Lecture 1 and check that this is the same as $P_{X(t),x} = P(X(t+1) = x \mid \mathcal{F}_t)$.

2.3. Gaussian approximation, drift, and volatility: We can write $X(t+1) = X(t) + Y(t)$, where $P(Y(t) = 1) = a$, $P(Y(t) = 0) = b$, and $P(Y(t) = -1) = c$. The random variables $Y(t)$ are independent of each other because of the Markov property and homogeniety. Assuming (without loss of generality) that $X(0) = 0$, we have

$$X(t) = \sum_{s=0}^{t-1} Y(s) \, , \tag{26}$$

which expresses $X(t)$ as a sum of *iid* (independent and identically distributed) random variables. The central limit theorem then tells us that for large t, $X(t)$ is approximately Gaussian with mean μt and variance $\sigma^2 t$, where $\mu = E[Y(t)] = a - b$ and $\sigma^2 = \text{var}[Y(t)] = a + c - (a-c)^2$. These are called *drift* and *volatility*[3] respectively. The mean and variance of $X(t)$ grow linearly in time with rate μ and σ^2 respectively. Figure 1 shows some probability distributions for simple random walk.

2.4. Trees: Simple random walk can be thought of as a sequence of decisions. At each time you decide: up(A), stay(B), or down(C). A more general sequence of decisions is a *decision tree*. In a general decision tree, making choice A at time 0 then B at time one would have a different result than choosing first B then A. After t decisions, there could be 3^t different decision paths and results.

The simple random walk decision tree is *recombining*, which means that many different decision paths lead to the same $X(t)$ For example, start (w.l.o.g) with $X(0) = 0$, the paths ABB, CAA, BBA, etc. all lead to $X(3) = 1$. A recombining tree is much smaller than a general decision tree. For simple random walk, after t steps there are $2t + 1$ possible states, instead of up to 3^t. For $t = 10$, this is 21 instead of about 60 thousand.

2.5. Urn models: Urn models illustrate several features of more general random walks. Unlike simple random walk, urn models are *mean reverting* and have *steady state probabilities* that determine their large time behavior. We will come back to them when we discuss *scaling* in future lectures.

The simple urn contains n balls that are identical except for their color. There are k red balls and $n - k$ green ones. At each state, someone chooses one of the balls at random with each ball equally likely to be chosen. He or she replaces the chosen ball with a fresh ball that is red with probability p and green

[3] People use the term *volatility* in two distinct ways. In the Black Scholes theory, volatility means something else.

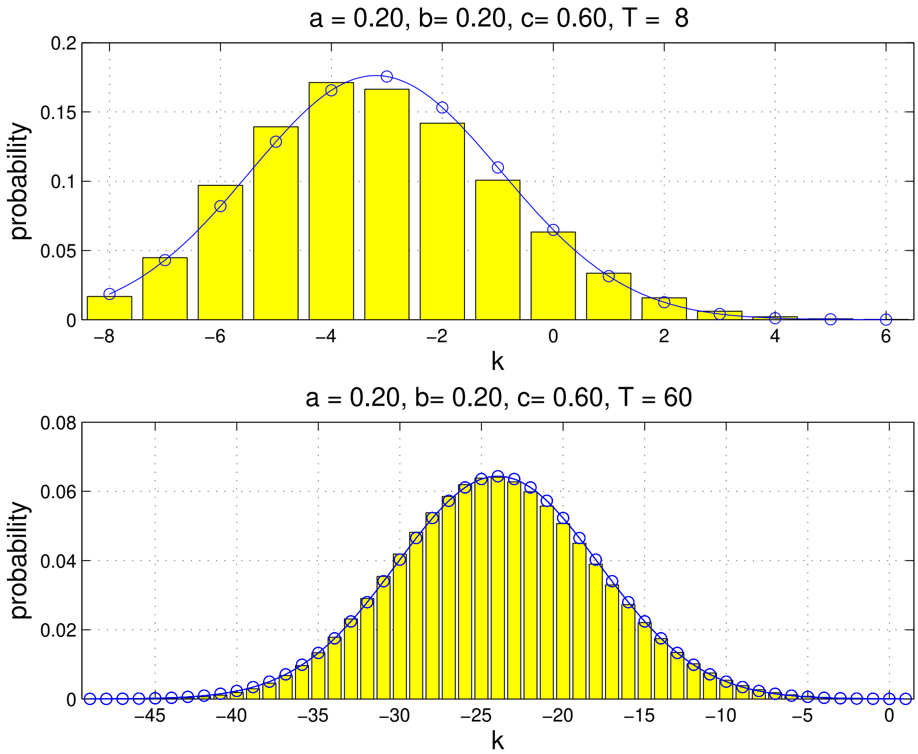

Figure 1: The probability distributions after $T = 8$ (top) and $T = 60$ (bottom) steps for simple random walk. The smooth curve and circles represent the central limit theorem Gaussian approximation. The plots have different probability and k scales. Values not shown have very small probability.

with probability $1-p$. All choices are independent. The number of red balls decreases by one if he or she removes a red ball and returns a green one. This happens with probabilty $(k/n)\cdot(1-p)$. Similarly, the $k \to k+1$ probability is $((n-k)/n)\cdot p$. In formal terms, the state space is the integers from 0 to n and the transition probabilities are

$$P_{k,k-1} = \frac{k(1-p)}{n} \ , \quad P_{kk} = \frac{(2p-1)k + (p-1)n}{n} \ , \quad P_{k,k+1} = \frac{(n-k)p}{n} \ ,$$

$$P_{jk} = 0 \text{ otherwise.}$$

If these formulas are right, then $P_{k,k-1} + P_{kk} + P_{k,k+1} = 1$.

2.6. Urn model steady state: For the simple urn model, the probabilities $u(k,t) = P(X(t) = k)$ converge to steady state probabilities, $v(k)$, as $t \to \infty$. This is illustrated in Figure (2). The steady state probabilities are

$$v(k) = \binom{n}{k} p^k (1-p)^{n-k} \ .$$

The steady state probabilities have the property that if $u(k,t) = v(k)$ for all k, then $u(k,t+1) = v(k)$ also for all k. This is *statistical steady state* because the probabilities have reached steady state values though the states themselves keep changing, as in Figure (3). In matrix vector notation, we can form the row vector, v, with entries $v(k)$. Then v is a statistical steady state if $vP = v$. It is no coincidence that $v(k)$ is the probability of getting k red balls in n independent trials with probability p for each trial. The steady state expected number of red balls is

$$E_v[X] = np \ ,$$

where the notation $E_v[]$ refers to expectation in probability distribution v.

2.7. Urn model mean reversion: If we let $m(t)$ be the expected value if $X(t)$, then a calculation using the transition probabilities gives the relation

$$m(t+1) = m(t) + \frac{1}{n}\left(np - m(t)\right) \ . \tag{27}$$

This relation shows not only that $m(t) = np$ is a steady state value ($m(t) = np$ implies $m(t+1) = np$), but also that $m(t) \to np$ as $t \to \infty$ (if $r(t) = m(t) - np$, then $r(t+1) = \alpha r(t)$ with $|\alpha| = \left|1 - \frac{1}{n}\right| < 1$).

Another way of expression mean reversion will be useful in discussing stochastic differential equations later. Because the urn Model is a Markov chain,

$$E\left[X(t+1) \mid \mathcal{F}_t\right] = E\left[X(t+1) \mid X(t)\right]$$

Again using the transition probabilities, we get

$$E\left[X(t+1) \mid \mathcal{F}_t\right] = X(t) + \frac{1}{n}\left(np - X(t)\right) \ . \tag{28}$$

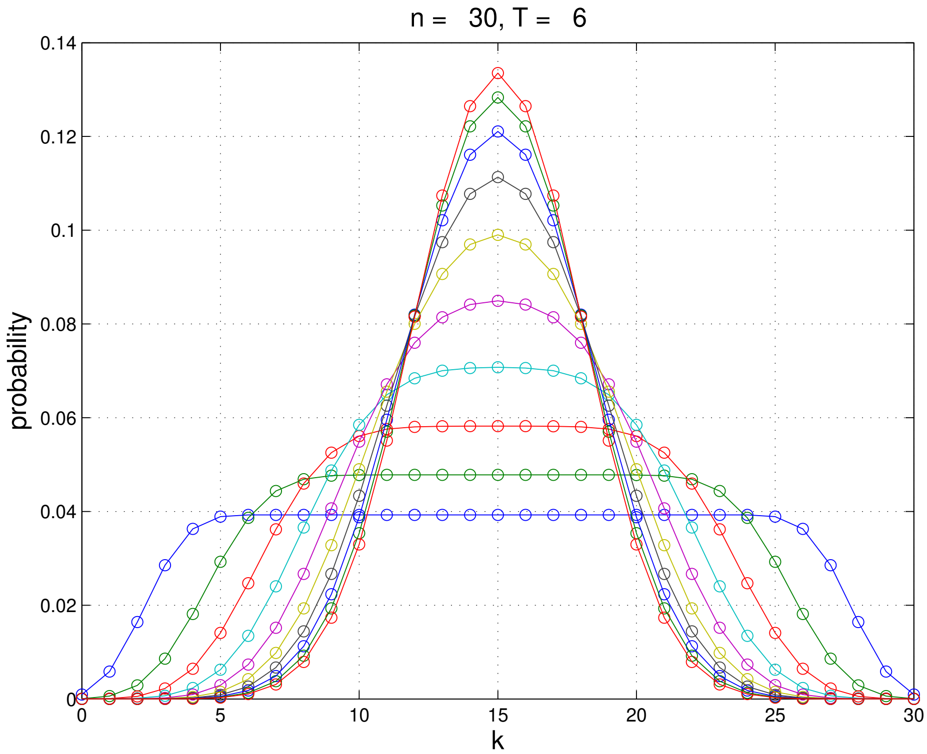

Figure 2: The probability distributions for the simple urn model plotted every T time steps. The first curve is blue, low, and flat. The last one is red and most peaked in the center. The computation starts with each state being equally likely. Over time, states near the edges become less likely.

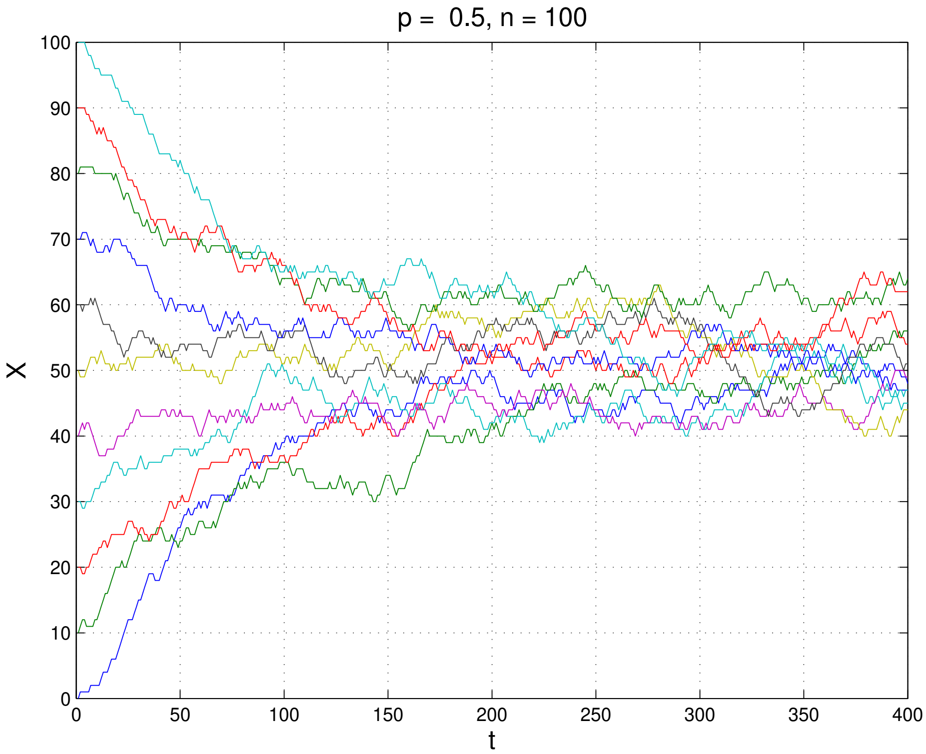

Figure 3: A Monte-Carlo sampling of 11 paths from the simple urn model. At time $t = 0$ (the left edge), the paths are evenly spaced within the state space.

If $X(t) > np$, we have

$$E[\Delta X(t)] = E[X(t+1) - X(t)] - \frac{1}{n}(np - X(t)) \ ,$$

is negative. If $X(t) < np$, it is positive.

2.8. Boundaries: The terms *boundary*, *interior*, *region*, etc. as used in the general discussion of Markov chain hitting probabilities come from applications in lattice Markov chains such as simple random walk. For example, the region $x > \beta$ has boundary $x = \beta$. The quantities

$$u(x,t) = P(X(t) = x \text{ and } X(s) > \beta \text{ for } 0 \leq s \leq t)$$

satisfy the forward equation (just (1) in this special case)

$$u(x,t+1) = au(x-1,t) + bu(x,t) + cu(x+1,t)$$

for $x > \beta$ together with the *absorbing boundary condition* $u(\beta,t) = 0$. We could create a finite state space Markov chain by considering a region $\beta < x < \gamma$ with simple random walk in the *interior* together with absorbing boundaries at $x = \beta$ and $x = \gamma$. Absorbing boundary conditions are also called *Dirichlet boundary conditions*.

Another way to create a finite state space Markov chain is to put *reflecting boundaries* at $x = \beta$ and $x = \gamma$. This chain has the same transition probabilities as ordinary random walk in the interior ($\beta < x < \gamma$). However, transitions from β to $\beta - 1$ are disallowed and replaced by transitions from β to $\beta + 1$. This means changing the transition probabilities starting from $x = \beta$ to

$$P(\beta \to \beta{-}1) = P_{\beta,\beta-1} = 0 \ , \ \ P(\beta \to \beta) = P_{\beta\beta} = b \ , \ \ P(\beta \to \beta{+}1) = P_{\beta,\beta+1} = a{+}c \ .$$

The transition rules at $x = \gamma$ are similarly changed to block $\gamma \to \gamma + 1$ transitions. There is some freedom in defining the reflection rules at the boundaries. We could, for example, make $P(\beta \to \beta) = b + c$ and $P(\beta \to \beta + 1) = a$, which changes the blocked transition to standing still rather than moving right. We return to this point in discussing *oblique reflection* in multidimensional random walks and diffusions.

2.9. Multidimensional lattice: The unit square lattice in d dimensions is the set of $d-$tuples of integers (the set of integers is called Z):

$$x = (x_1, \ldots, x_d) \text{ with } x_j \in Z \text{ for } 1 \leq j \leq d \ .$$

The scaled square lattice, with *lattice spacing* $h > 0$, is the set of points $hx = (hx_1, \ldots, hx_d)$, where x are integer lattice points. In the present discussion, the scaling is irrelevent, so we use the unit lattice. We say that latice points x and y are *neighbors* if

$$|x_j - y_j| \leq 1 \text{ for all coordinates } j = 1, \ldots, d \ .$$