Probability Generating Functions and Branching Processes
Thursday, November 05, 2015
2:02 PM

Reading:
Karlin and Taylor Sec. 1.1E
Resnick Secs. 1.3, 1.4
Lawler Sec. 2.4

Homework 3 will be posted by tomorrow morning; due Monday, November 23 at 2 PM.

We will prepare to discuss a special kind of countable state Markov chain, whose probability transition matrix is awkward to write down, but can be analyzed efficiently because it involves more independence than standard Markov chains do.

We will develop, for this purpose, the technique of probability generating functions, which are often useful when dealing with sums of independent random variables.

Probability generating function of a discrete random variable X is defined as follows:

$$\mathcal{P}_X(s) \equiv \mathbb{E}\, s^X = \sum_{j=0}^{\infty} s^j P(X=j)$$

Here we have a dummy variable $0 \leq s \leq 1$. (Can also extend the concept to the case when X takes on negative values, but we'll leave this out of our consideration.)

- The probability generating function so defined is related to, but not the same as moment generating function or the characteristic function in probability theory.
- The purpose of a generating function is analogous to transforms that are used to solve or analyze differential equations by converting the formulation to one where calculations are simpler
  - Probability generating function : z-transform
  - Characteristic function: Fourier transform
  - Moment generating function : Laplace transform
- Characteristic and moment generating functions are versions of the probability generating function that work for a wider class of random variables; probability generating function is well suited only to discrete random variables.
- Essential to all these concepts is that the transforms are invertible; from the transformed function, one can recover uniquely the function which transforms into it.

  What that means for the probability generating function, is that it is a transform of the probability distribution or probability mass function

$p_j = P(X = j)$ for $j = 0,1,2, \ldots$

$$\mathcal{P}_X(s) = \sum_{j=0}^{\infty} p_j s^j$$

How do we invert the probability generating function to get back the probability mass function:

$$p_j = \frac{1}{j!} \left(\frac{d}{ds}\right)^j \mathcal{P}_X(s) \Big|_{s=0}$$

Well, that assumes the probability generating function is a convergent Taylor series about $s = 0$, but the fact that $\sum_{j=0}^{\infty} p_j = 1, p_j \geq 0$ means that the Taylor series will converge at least on the interval $[-1,1]$.

The purpose of generating functions is they make some calculations simpler.

If we know the probability generating function, we can compute any moment of the random variable by differentiation (rather than summation of the probability mass function):

$$\mathbb{E} X^m = \mathbb{E} \left(s \frac{d}{ds}\right)^m s^X \Big|_{s=1}$$

$$= \left(s \frac{d}{ds}\right)^m \mathbb{E} s^X \Big|_{s=1}$$

$$= \left(s \frac{d}{ds}\right)^m \mathcal{P}_X(s) \Big|_{s=1}$$

Example: Compute mean and variance of the binomial distribution.

$X \sim Bin(n;p)$

$$P(X = j) = p_j = \binom{n}{j} p^j (1-p)^{n-j}$$

n n

$$P(X = j) = \binom{n}{j} p^j (1-p)^{n-j}$$

$$\mathbb{E} X^m = \sum_{j=0}^{n} j^m p_j = \sum_{j=0}^{n} j^m \binom{n}{j} p^j (1-p)^{n-j}$$

... minor pain

Let's instead approach this by using the probability generating function:

$$\mathcal{P}_X(s) = \mathbb{E} s^X = \sum_{j=0}^{\infty} p_j s^j$$

$$= \sum_{j=0}^{n} s^j \binom{n}{j} p^j (1-p)^{n-j}$$

$$= \sum_{j=0}^{n} \binom{n}{j} (ps)^j (1-p)^{n-j}$$

binomial theorem

$$= (ps + 1 - p)^n$$

$$\mathbb{E} X = \left(s \frac{d}{ds}\right) \mathcal{P}_X(s) \Big|_{s=1}$$

$$= \left(s \frac{d}{ds}\right) (ps + 1 - p)^n \Big|_{s=1}$$

$$= s(np)(ps + 1 - p)^{n-1} \Big|_{s=1}$$

$$= np(p + 1 - p)^{n-1} = np$$

$$\mathbb{E}\, X = np$$

$$\left(s\frac{d}{ds}\right)\left(s\frac{d}{ds}\right)f = s^2 f'' + s f'$$

$$\mathbb{E}\, X^2 = \left(s\frac{d}{ds}\right)^2 \phi_X(s)\Big|_{s=1}$$

$$= \left[\left(s\frac{d}{ds}\right)\left(s(np)(ps+1-p)^{n-1}\right)\right]\Big|_{s=1}$$

$$= \left[s\left(np(ps+1-p)^{n-1} + s(np)((n-1)p)(ps+1-p)^{n-2}\right)\right]\Big|_{s=1}$$

$$= np(p+\cancel{1-p})^{n-1} + (np)((n-1)p)(\cancel{p}+1-p)^{n-2}$$

$$\mathbb{E}\, X^2 = np + n(n-1)p^2$$

$$\mathrm{Var}\, X = \mathbb{E}\, X^2 - (\mathbb{E}\, X)^2 = np + n(n-1)p^2 - (np)^2$$

$$= np - np^2 = np(1-p)$$

Actually for computing moments, the moment generating function is a little bit easier to work with, but that's not our ultimate objective here.

More importantly than this, probability generating functions expedite the analysis of sums of independent random variables.

$Y = \sum_{j=1}^{N} X_j$ with X_j as independent random variables with known probability distributions. The probability distribution for Y is related in a somewhat awkward way to the probability distributions for X_j, namely by convolution (which means you have a (n-1)-fold sum).

However, the probability generating function of Y is easily related to the probability generating functions of X_j:

$$\mathcal{P}_Y(s) = \mathbb{E}\, s^Y = \mathbb{E}\, s^{\sum_{j=1}^N X_j}$$

$$= \mathbb{E} \prod_{j=1}^N s^{X_j}$$

independence

$$= \prod_{j=1}^N \mathbb{E}\, s^{X_j}$$

$$\boxed{\mathcal{P}_Y(s) = \prod_{j=1}^N \mathcal{P}_{X_j}(s)}$$

Probability generating functions also work well with "random sums" meaning a sum of a random number of random variables.

So let us define $Z = \sum_{j=1}^N X_j$ where the X_j are iid random variables, and N is a nonnegative integer random variable, which is independent of the X_j.

Then again, the probability generating function for Z can be related to the probability generating functions for N, X_j:

$$\mathcal{P}_Z(s) = \mathbb{E}\, s^Z = \mathbb{E}\, s^{\sum_{j=1}^N X_j}$$

$$= \mathbb{E} \prod_{j=1}^N s^{X_j}$$

We now use the law of total expectation because the calculation simplifies if we "know" the value of the random variable N. Partition on the value of N:

$$= \sum_{n=1}^{\infty} \left(\underbrace{\mathbb{E}\left(\prod_{j=1}^{n} s^{X_j} \,\middle|\, N=n \right)}_{X_j \text{ independent of } N} P(N=n) \right)$$

$$= \sum_{n=0}^{\infty} \underbrace{\mathbb{E}\left[\prod_{j=1}^{n} s^{X_j} \right]}_{\text{independence of } X_j} P(N=n)$$

$$= \sum_{n=0}^{\infty} \prod_{j=1}^{n} \mathbb{E}\, s^{X_j} \quad P(N=n)$$

$$= \sum_{n=0}^{\infty} \prod_{j=1}^{n} \underset{\substack{\equiv \\ P_{X_j}(s) \text{ because iid}}}{P_X(s)} \quad P(N=n)$$

$$= \sum_{n=0}^{\infty} \left(P_X(s) \right)^n P(N=n)$$

$$\boxed{P_Z(s) = P_N\left(P_X(s) \right)}$$

A mathematical shorthand for this conditional expectation calculation:

$$P_Z(s) = \ldots = \mathbb{E} \prod_{j=1}^{N} s^{X_j}$$

$$= \mathbb{E}\left(\mathbb{E}\left[\prod_{j=1}^{N} s^{X_j} \,\middle|\, N \right] \right)$$

$$= \mathbb{E}\left(\mathbb{E}\left[\prod_{j=1}^{N} s^{X_j} \,\middle|\, N\right]\right)$$

$$= \mathbb{E}\left(\prod_{j=1}^{N} \mathbb{E}\, s^{X_j}\right)$$

$$= \mathbb{E}\left(\prod_{j=1}^{N} \mathcal{P}_X(s)\right)$$

$$= \mathbb{E}\left(\left(\mathcal{P}_X(s)\right)^N\right)$$

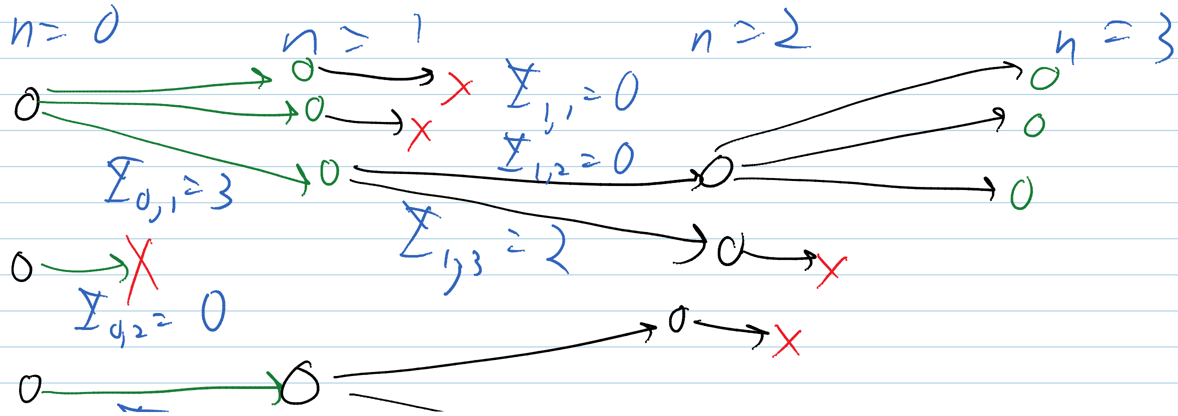
$$= \mathcal{P}_N\left(\mathcal{P}_X(s)\right)$$

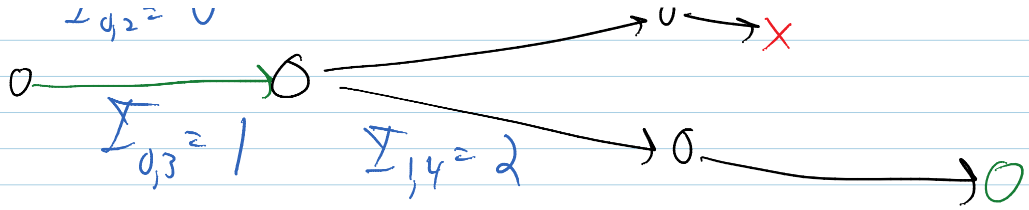
More details on generating functions in Resnick Ch. 0.

Branching Processes (Galton-Watson model)

This is a special version of a countable state Markov chain with state space $S = \{0,1,2,\dots\} = \mathbb{Z}_{\geq 0}$ that is modeled as follows:

At any epoch n, X_n represents the number of active "agents" at that epoch. At each epoch, each agent gives rise to a random number of offspring. If the agent survives to the next epoch, we just count itself as one of its own offspring. Each agent's offspring is assumed to be iid. The process continues at each epoch.





$X_0=3$ $X_1=4$ $X_2=4$ $X_3=4$

This describes a Markov chain because it combines the current state with independent noise at each epoch. The probability transition matrix is awkward, but the stochastic update rule can be compactly represented as a random recursion, namely a random sum:

$X_{n+1} = \sum_{k=1}^{X_n} Y_{n,k}$ where $Y_{n,k}$ represents the number of offspring of agent k at epoch n. The $Y_{n,k}$ are taken as independent, identically distributed random variables with a common probability mass function:

$p_j = P(Y_{n,k} = j)$ for any $j = 0,1,2,\ldots$ This is known as the offspring distribution.

And as always we need to supply the probability distribution for the number of agents that are present at epoch 0, call this:

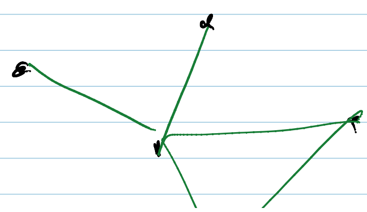
$\phi_j^{(0)} = P(X_0 = j)$ for $j = 0,1,2,\ldots$

We have described the simplest type of branching process, but the ideas can be generalized in a number of ways:

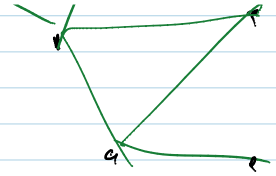
- multiple types of agents
- age structure
- continuous time
- see Branching Processes in Biology, Kimmel and Axelrod

Applications of branching processes:

- genealogy
- genetics
- biomolecular reproduction processes (cells, PCR)
- population
- cell growth to tumors
- queueing models
- dynamics on networks
  - social network
  - network of computers
  - neuronal network



- social network
- network of computers
- neuronal network
- One often uses an approximation for the dynamics on a network in which the spread of an infection/product/buzz/virus/signal is represented by a branching process. This is appropriate so long as loops in the dynamics can be neglected, which is particularly relevant often at early times. The branching process approximation works better than you would think…
  - Melnik, Hackett, Porter, Mucha, and Gleeson, "The Unreasonable Effectiveness of Tree-Based theory for Networks with Clustering," Physical Review E 83 (2011), 036112.



How do we compute with branching processes? The probability transition matrix is awkward to construct, but the stochastic update rule looks like a random sum. So probability generating functions should work well…

If we apply the random sum rule for probability generating functions, we get:

$$P_{X_{n+1}}(s) = P_{X_n}\left(P_Y(s)\right)$$

where $$P_Y(s) = \sum_{j\geq 0}^{\infty} p_j s^j$$

is the pgf of the offspring distribution.

Thus, we can recursively compute the probability generating function for X_n by iterating composition of the pgf of the offspring distribution n times back to the initial pgf for X_0.