Inference, Models and Simulation for Complex Systems CSCI 7000-001
Lecture 3 1 September 2011
Prof. Aaron Clauset

1 Power-law distributions and working with empirical data

Suppose we have some empirical observations $\{x_i\} = \{x_1, x_2, \ldots, x_n\}$ to which we'd like to fit a power-law distribution. Recall from Lecture 2 that there are two parameters we need to know to do this: α and $x_{\min}$. That is, we need to know the scaling exponent and we need to know where the power-law behavior starts.

1.1 Estimating α

For the moment, let's assume we already know $x_{\min}$. In that case, we can use the method of maximum likelihood to derive an MLE for α.

$$\begin{aligned}\ln \mathcal{L}(\{x_i\} \,|\, \alpha, x_{\min}) &= \ln \left[\prod_{i=1}^{n} \frac{\alpha - 1}{x_{\min}} \left(\frac{x_i}{x_{\min}} \right)^{-\alpha} \right] \\ &= n \ln \left(\frac{\alpha - 1}{x_{\min}} \right) - \alpha \sum_{i=1}^{n} \ln \left(\frac{x_i}{x_{\min}} \right) \quad .\end{aligned}$$

Now solving $\partial \mathcal{L} / \partial \alpha = 0$ for α yields

$$\hat{\alpha} = 1 + n \Bigg/ \sum_{i=1}^{n} \ln \left(\frac{x_i}{x_{\min}} \right) \quad , \tag{1}$$

and the standard error estimate in the MLE can be shown to be $\hat{\sigma} = (\hat{\alpha} - 1)/\sqrt{n}$.

Note that the MLE for α depends on the unspecified parameter $x_{\min}$, and, because the formula for the standard error estimate depends on $\hat{\alpha}$, it does too. That is, we can't actually use these equations until we've chosen a value for $x_{\min}$ that tells us where the power-law behavior begins.

1.2 Estimating $x_{\min}$

But how do we choose an $x_{\min}$? Unfortunately, maximum likelihood won't work because increasing $x_{\min}$ decreases the number of observations n in our sample. The likelihood function monotonically decreases with increasing n, so truncating our sample never decrease the likelihood, and, unless the likelihood of some value is $\ell_i(x_i) = 1$, will typically *increase* the likelihood. The consequence of this behavior is that a maximum likelihood approach would simply converge on the trivial—and

unhelpful—solution of $\hat{x}_{\min} = \max_i x_i$, which leaves us with a data set of size $n = 1$.

This means we must use some other estimation technique to choose $\hat{x}_{\min}$. The downside is that we lose the nice properties and guarantees that come with maximum likelihood and likelihood functions in general, so we'll have to worry about bias and consistency.

1.2.1 The Hill plot

In the past, and particularly in quantitative finance, people have often used a visual diagnostic called a "Hill plot" to choose $\hat{x}_{\min}$. The idea is to simply try all values of $x_{\min}$ and choose the one that yields a "stable" or "visually reasonable" fit to the data. Mathematically, it constructs and visualizes the function $\hat{\alpha}(x_{\min})$ on the domain $x_{\min} \in \{x_i\}$. A flat-ish region of this function represents a stable-ish estimate of $\hat{\alpha}$, and the heuristic is to choose $\hat{x}_{\min}$ as the beginning of that region.

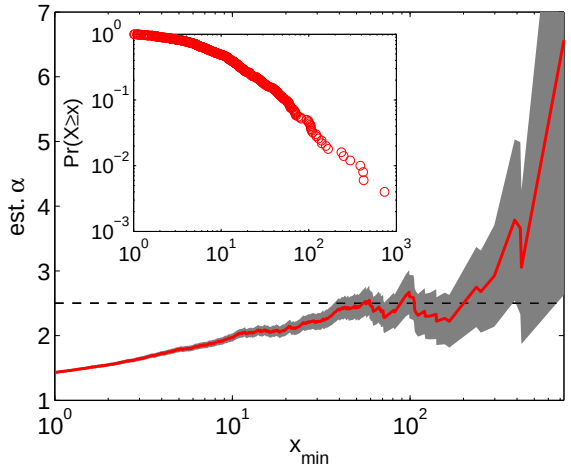

Figure 1: A Hill plot for $n = 500$ observations drawn from a shifted power-law distribution with $k = 15$ and $\alpha = 2.5$. (The inset shows the ccdf of the data.) The dashed line shows the true value of α; the shaded region shows the standard uncertainty around $\hat{\alpha}$. Visually, there's no obvious choice of $x_{\min}$ that yields an accurate estimate of α.

Unfortunately, in addition to being subjective and un-automated, this approach isn't reliably accurate. For instance, Figure 5 shows a Hill plot for $n = 500$ observations drawn iid from a shifted power law with $k = 15$ and $\alpha = 2.5$; the inset shows the ccdf of the actual data.[1] Visually, things

[1] Recall from Lecture 2 that for a shifted power law, there is no *correct* choice of $x_{\min}$, above which the distribution

start getting flat somewhere around $x_{\min} \approx 20$, but this yields $\hat{\alpha} \approx 2$, which is a much heavier tail than is accurate. The deviations from the true value of α for smaller values of $x_{\min}$ are caused by fitting a power-law model to non-power-law data. The deviations for larger values of $x_{\min}$ are caused by variance induced by sampling noise and by a small-sample bias in the MLE for α. For these data, there's no visually obvious choice of $x_{\min}$ that yields an accurate estimate of α because these two regions overlap significantly, and this is why a Hill plot is not a good way to choose $\hat{x}_{\min}$.

An important message: In the wide world of data analysis techniques, there are many visual diagnostic tools with a similar flavor. Be forewarned: keep your statistical wits about you when you encounter them. *Looking* at your data is a very important part of data analysis because there is no better way to get an unfiltered view of your data than through simple visualizations: plot distributions of quantities, scatter plots of attributes, time series, etc. The trouble enters when visualization requires the use of statistical tools that have built-in assumptions, and all data analysis tools have built-in assumptions. Thus, the most important message in this class is that *in order to understand what your data are saying to you*, you must first understand what your statistical tools do—what input they require, what input they *expect*, what operations they perform on the data, why they performs those and not other operations, and finally how to interpret their output correctly.

The Hill plot diagnostic fails to provide a *reliable* way to choose $x_{\min}$ because it does not provide a quantitative and objective answer to the following questions: (i) How do we automatically quantify "flat"-ness? And, (ii) given a method to do so,[2] how "flat" is flat enough?

1.2.2 KS minimization

An objective and fairly accurate solution to choosing $x_{\min}$ is to minimize a so-called "goodness-of-fit" (gof) statistic,[3] such as the Kolmogorov-Smirnov (KS) statistic, between the fitted model and the data. The KS statistic is defined as

$$D = \max_{x \in \{x_i\}} \left| P(x \,|\, \hat{\theta}) - S(x) \right| \quad , \tag{2}$$

follows a pure power law; thus, our goal would be to simply choose some value of $x_{\min}$ that yields an accurate estimate of α. That is, $x_{\min}$ is a nuisance parameter.

[2]For instance, we could fit a first-order polynomial to the $\hat{\alpha}(x_{\min})$ function above some choice of $\hat{x}_{\min}$.

[3]There are a number of other such statistical measures, including the sum-of-squared errors, the weighted-KS statistic, etc. These statistics generally have nice mathematical properties, which is why there are commonly used. In general, all such measures quantify the magnitude and direction of deviations between the observed data and some model of that data.

where $P(x \mid \hat{\theta})$ is the theoretical cdf, with parameters $\hat{\theta}$ and $S(x)$ is the empirical cdf.[4]

The KS statistic (typically denoted D) measures the largest deviation in cumulative density between the fitted model and the empirical data. Figure 6 shows an example of this for a small sample from an exponential distribution. Because D measures the maximum deviation, a small D implies that the fitted model $P(x \mid \hat{\theta})$ is everywhere close to the empirical distribution. (What kind of deviations is the KS statistic most sensitive to? For what kind of questions might this behavior matter?)

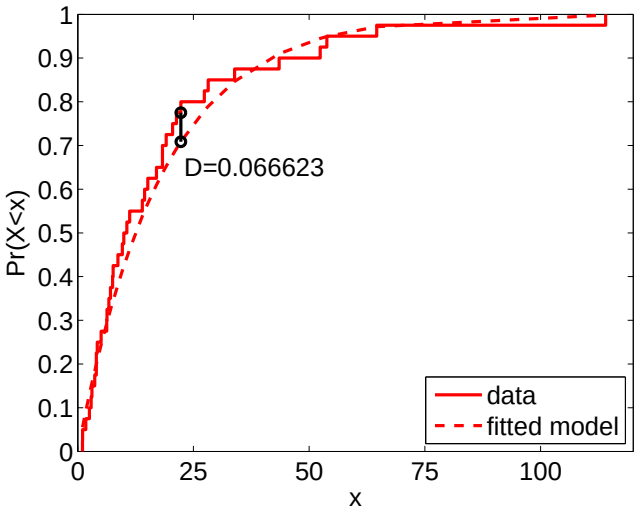

Figure 2: The empirical cdf (solid line; $n = 40$, $\lambda = 0.050$) and the maximum likelihood theoretical cdf (dashed line; $\hat{\lambda} = 0.053 \pm 0.009$) for an exponential distribution. The black line shows the maximum absolute deviation between the two functions of $D = 0.066623$.

Suppose that we choose $x_{\min}$ too small, such that we fit the power-law model to the distribution's "body" where there are significant deviations from the power-law behavior. In this case, D should be large because of a model-misspecification bias coming from the data. On the other hand, if we choose $x_{\min}$ too large, and fit the model to the distribution's extreme tail where there are few observations, statistical noise or variance in the data will make D large. We want to choose an $x_{\min}$ somewhere between these two, that is, we want a balanced tradeoff between bias on the one

[4]The empirical cdf $S(x)$ is a step function, defined as the fraction of the full data set that are less than some value x. If we sort our data so that $x_1 < x_2 < \cdots < x_n$, then the corresponding y values for the empirical cdf, in order, are $\{0, \frac{1}{n}, \frac{2}{n}, \ldots, \frac{n-1}{n}\}$.

hand and variance on the other. This can be done by choosing $\hat{x}_{\min}$ in the following way:

$$\hat{x}_{\min} = \inf_{x_{\min} \in \{x_i\}} \left(\max_{x \geq x_{\min}} |P(x \,|\, \hat{\alpha}, \hat{x}_{\min}) - S(x)| \right) \quad . \tag{3}$$

That is, we estimate $\hat{x}_{\min}$ as the value of x_i that yields the smallest maximum deviation. Note that each time we increase our candidate value for $\hat{x}_{\min}$, we need to truncate the data set, re-estimate α and compute a new theoretical cdf and empirical cdf.

Using the same data in Fig. 5, we can apply the KS minimization technique (see below for example code in Matlab). The results are shown in Figure 7, and we find $\hat{x}_{\min} = 39.79$; at this choice, we get $\hat{\alpha} = 2.41 \pm 0.16$, which is indistinguishable from the true value of $\alpha = 2.5$.

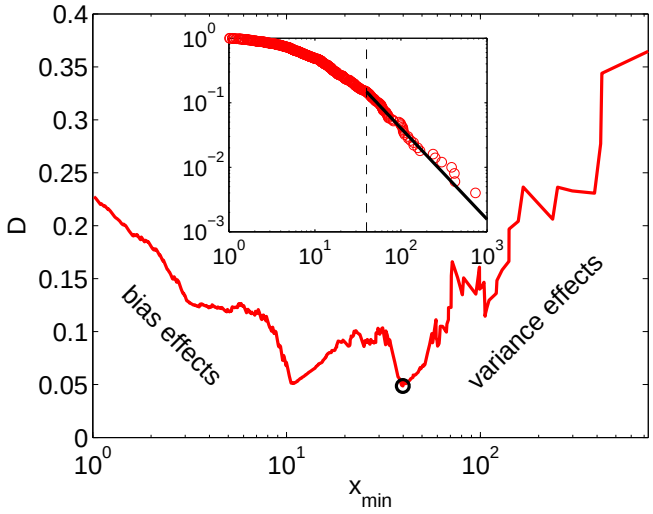

Figure 3: The function $D(x_{\min})$ for the same data in Fig. 5, with the smallest value marked.

This agreement between $\hat{\alpha}$ and α, however, is slightly misleading. Recall that these data are drawn from a shifted power law with $k = 15$; at $\hat{x}_{\min} \approx 30 = 2k$, there should still be non-trivial deviations from the pure power-law model we're fitting (see Lecture 2). In fact, I had to generate several synthetic data sets to get a fit this good. The point here is that KS minimization provides an automatic, objective and fairly reliable way to choose $x_{\min}$, even if it doesn't provide the strong guarantees we expect from a maximum likelihood procedure (nice things like asymptotic consistency and normally-distributed errors). That is, this is a nice example of a reasonable computational procedure for solving a tricky statistical problem.

Numerical experiments on its accuracy show that it's also quite reliable, although it does make some mistakes. It's good that we can measure and quantify these mistakes on synthetic data, where we know what the true underlying structure is: it means we can learn how to interpret its behavior, including potential mistakes, when we apply it to empirical data with unknown structure. In general, these numerical experiments suggest that when applied to data drawn from a distribution that actually exhibits a pure power-law form above an explicit value of $x_{\min}$, KS minimization is slightly conservative, i.e., it tends to choose an $\hat{x}_{\min} \gtrsim x_{\min}$. Unfortunately, it's not known in general how large a bias KS minimization can exhibit or even why, mathematically, it works so well in practice.

1.3 Matlab code

Matlab code for doing the KS minimization procedure for continuous data; the structure of the procedure is very similar for discrete data.

Let `x` be a vector of empirical observations:

```
xmins = unique(x);                      % search over all unique values of x
xmins = xmins(1:end-1);
dat   = zeros(size(xmins));
z     = sort(x);
for xm=1:length(xmins)
     xmin = xmins(xm);                  % choose next xmin candidate
     z    = z(z>=xmin);                 % truncate data below this xmin value
     n    = length(z);                  %
     a    = n ./ sum( log(z./xmin) );   % estimate alpha using direct MLE
     cx   = (0:n-1)'./n;                % construct the empirical CDF
     cf   = 1-(xmin./z).^a;             % construct the fitted theoretical CDF
     dat(xm) = max( abs(cf-cx) );       % compute the KS statistic
end;
D     = min(dat);                       % find smallest D value
xmin  = xmins(find(dat<=D,1,'first'));  % find corresponding xmin value
z     = x(x>=xmin);                     %
n     = length(z);                      %
alpha = 1 + n ./ sum( log(z./xmin) );   % get corresponding alpha estimate
L = n*log((alpha-1)/xmin) - alpha.*sum(log(z./xmin)); % log-likelihood at estimate
```

2 Mechanisms that produce power-law distributions

2.1 Some history

So far, we've learned about some of the mathematical properties of power-law distributions, and statistical methods for inferring their presence. What we have not yet discussed are the mechanisms that produce these strange distributions. In the mid-1990s, when large data sets on social, biological and technological systems were first being put together and analyzed, power-law distributions seemed to be everywhere. From the sizes of forest fires, to the sizes of international wars, to the number of network connections a computer has on the Internet, to the number of chemical reaction partners biological proteins have, to the number of calls received by phone numbers, etc. etc. etc. There were dozens, possibly hundreds of quantities, that all seemed to follow the same pattern: a power-law distribution.

The excitement over power laws was partly due to the enormous diversity they implied for the sizes of observations and partly due to their apparent ubiquity in complex systems. But, a large share of the exuberance, I think, was due to the enthusiasm physicists have for "scale invariance." That is, the appearance of a power-law distribution implies something special about the underlying generative process, that the process operates exactly the same whether on "small" things or "large" things. This invariance strikes a deep chord in the scientific soul because it suggests the existence of a fundamental "law." If we could understand what law produced all of these power-law distributions, perhaps we could develop a single scale-invariant theory of complex systems.

Initially, there were only a few models known to give rise to power-law distributions, and a number of people held out hope that these could be classified into an even smaller number of universal equivalence classes that would explain the appearance of power-law distributions throughout the complex world. Early candidates of these grand theories were self-organized criticality (a.k.a. "SOC," put forward by the colorful physicist Per Bak and collaborators), high-optimized tolerance (a.k.a. "HOT," put forward by the equally colorful engineer John Doyle and his collaborator Jean Carlson, in opposition to SOC), and preferential attachment (a.k.a. "cumulative advantage" etc., rediscovered by another colorful physicist Laszlo Barabasi and his collaborator Reka Albert). Unfortunately for these universalists, there are now dozens or more mechanisms that produce power-law distributions and more rigorous statistics tools suggest that power laws may not be nearly as ubiquitous as initially thought.

There are far too many power-law generating mechanisms to survey completely. Instead, we'll hit a couple of highlights.

2.2 The Reed-Hughes sampling mechanism

One of the simplest ways to generate a power-law distribution is via sampling. Reed and Hughes, in a short paper subtitled "Why power laws are so common in nature"[5] and published in *Physical Review E* in 2002, showed how this works.

Suppose some quantity y follows an exponential distribution

$$\Pr(y) \propto \mathrm{e}^{-\lambda y} \ , \tag{4}$$

where $\lambda > 0$; for instance, y is the waiting time for events generated by a Poisson process. But, suppose that we're not actually interested in y, but rather some *other* quantity x, which is related to y via an exponential function

$$\begin{aligned} x &\propto \mathrm{e}^{\delta y} \\ \mathrm{d}x &\propto \delta \mathrm{e}^{\delta y} \mathrm{d}y \ , \end{aligned} \tag{5}$$

where $\delta > 0$. That is, x is a quantity that grows exponentially with y, but y itself is distributed exponentially. For instance, continuing the Poisson process idea, suppose that x is the amount of money in a bank account, which grows according to some fixed interest rate, but that with constant probability p, each month we withdraw everything from the account. The relevant question is thus, What is the distribution of money withdrawn?

The answer is simply the distribution of x, which is given by setting $\Pr(x)\mathrm{d}x = \Pr(y)\mathrm{d}y$, plugging in Eq. (5), and solving for $\Pr(x)$.

$$\Pr(x) = \Pr(y)\frac{\mathrm{d}y}{\mathrm{d}x} \propto \frac{\mathrm{e}^{-\lambda y}}{\delta \mathrm{e}^{\delta y}} \propto x^{-(1+\lambda/\delta)} \ . \tag{6}$$

(I've omitted a few steps of algebra; do you see how to make the calculation go through?) Thus, the distribution of x follows a power-law distribution with an scaling exponent set by the ratio of the two exponential parameters.

Does this result behave as we expect?[6] For a fixed sampling rate λ, if we increase the exponential growth parameter δ, the power-law tail gets heavier. That is, we see more large-valued events under the sampling, which makes sense because the exponential growth is stronger meaning that we get larger things faster. On the other hand, for a fixed exponential growth rate δ, if we increase the sampling parameter λ, the power-law tail gets lighter. That is, we see fewer large-valued events. This also makes sense because a larger λ implies shorter waiting times because the distribution

[5] This subtitle gives you a sense of what kinds of stakes physicists felt they were playing for around this time.

[6] Answering the question "Does it make sense?" is a very useful way to check whether or not your modeling or statistical results are correct.

decays faster and thus there's less time for the exponential growth to generate large events.

It's worth pointing out that this simple sampling mechanism generalizes to distributions with arbitrarily structured bodies and exponential tails, that is, distributions that can be expressed in the form $L(x)\mathrm{e}^{-\lambda x}$ where $L(x)$ is a slowly varying function (see Lecture 2). This means that if we have an exponentially growing quantity but sample (observe) it using something like a Gamma distribution, the frequency of the largest events will follow a power-law tail.

As a more biological example of where the Reed-Hughes sampling mechanism can generate power-law distributions, consider the following. Suppose we add a single bacterium to each of k petri dishes. To make things mathematically simple (but probably biologically unrealistic) let's further assume that there are no space or resource constraints on the reproducing bacteria—they all have lots of sugars to digest and lots of space to grow. The number of bacteria x in the ith colony grows exponentially, since bacterial reproduction is a simple binary fission process.

But, now suppose we get very distracted by some other pressing issues, e.g., the latest news coverage of a wildfire, a big football game, or working on the CSCI 7000 problem set, which leads us to observe the size of the ith colony at each time step with some very small probability q. Thus, the observation times are a Poisson process while the population size is an exponential growth process, exactly as required by the Reed-Hughes mechanism. The result is that the *observed* distribution of colony sizes follows a power-law distribution.

2.3 Forest fires, self-organized criticality and highly-optimized tolerance

A classic toy model for generating power-law distributions comes from *self-organized criticality* (SOC): the forest-fire model,[7] which is a kind of cellular automata.[8]

Assume an $n \times n$ lattice, where each cell is one of three states: empty, tree or burning. Time proceeds in discrete steps. At each step, with probability p_{grow}, each "empty" cell becomes a "tree" cell; each cell that is "burning" becomes "empty;" and each "tree" cell becomes "burning" with probability $p = 1$ if it is adjacent (on a von Neuman neighborhood) to a burning cell or with probability $p_{\mathrm{lightning}}$ if it is not. Thus, if lightning strikes a tree, it will burn that cell at this time step, and move to all neighboring tree cells at the next time step. In this way, the fire spreads

[7] The other classic model is that of sand piles and avalanches, due to Bak, Tang and Wiesenfeld in 1988, which we'll encounter on the second problem set. There are different, more recent SOC models, too, for instance, models of polymer formation and rain fall.

[8] Cellular automata were extremely popular in the 1990s as a way of simulating the emergent consequences of the repeated application of simple rules across each location in a spatial system. The most famous of these is Conway's *Game of Life*. Classic cellular automata have deterministic rules, but stochastic rules are sometimes also used. More recently, cellular automata have been used as models of cancer growth, ecology dynamics, vehicular traffic, and many other phenomena.

recursively through the connected component of trees that contains the cell where the fire starts. The "size" of the fire is simply the number of tree cells that burn in a fire. For example, see Fig. 1a. One question we can ask is, What is the steady-state distribution of forest fire sizes?

This toy model was famously studied by Malamud, Morein and Turcotte in a 1998 *Science* paper; early versions of the model go back to Henley in 1989. This model is also occasionally discussed in the forestry literature because the empirical distribution for the frequency of very large forest fires is heavy tailed and, to some, looks like a power-law distribution (see Fig. 1b). Furthermore, the distribution of fire sizes in the forest-fire model is a power-law distribution. The mathematics to show this rely on techniques derived from percolation theory in physics,[9] but it's fairly easy to explicitly simulate the whole process to derive the same result.[10] A simulation also allows us to easily pose and answer other questions, e.g., how do finite-size effects or fire breaks impact the distribution of fire sizes? The general result from the mathematics, however, is that no matter what state the system currently sits in, over time it will evolve to a *critical state*, where the dynamic distribution of cluster sizes follows a power law and thus the distribution of fire sizes follows the same form.

This model was criticized by Carlson and Doyle in a 1999 *Physical Review E* paper on the grounds that it leaves no room for the impact of natural selection or human design. Using the same basic model (a lattice, trees and fires), Carlson and Doyle showed that if a forest manager optimizes the "yield" of the entire land by inserting fire breaks (regions of land with no trees, which serve to prevent a fire from spreading from one patch of trees to another), the size distribution of fires stills follows a power law, but now for a completely different reason than in the original SOC version—that is, rather than self-organizing to the critical state, the system has been externally optimized to it. They called this model *highly-optimized tolerance* (HOT) and argued forcefully that engineered systems, including human-designed systems like the Internet, road networks, the economy, etc., as well as biologically evolved systems, should be modeled using a HOT framework rather than a SOC framework.[11]

The trouble with toy models like these is twofold. First, they're not meant to be models of real processes and thus it's not clear how much they really tell us about the real world. The main benefit is that they build our intuition, but perhaps building intuition about irrelevant processes is not helpful. Second, these models are difficult to compare with empirical data. In the past, "tests" of such models have been done by comparing the only thing they could, the model's prediction of and the empirical distribution for fire sizes. When done carefully, however, such comparisons

[9] See `http://guava.physics.uiuc.edu/~nigel/courses/563/Essays_2010/PDF/Funke.pdf` for instance.

[10] The main detail in doing the simulation is to do it in such a way that one can keep track of the size of a fire. For instance, the code in Section 3 does not support this; it's just a nice visualization.

[11] For more on this debate, see Newman, Girvan and Farmer, "Optimal Design, Robustness, and Risk Aversion" *Physical Review Letters* **89** 028301 (2002).

almost always suggests that the model is wrong or at least highly incomplete. (For instance, the p-value for the power-law tail in Fig. 1b is $p = 0.05$, indicating strong deviations from the power-law prediction.) Making them more realistic is basically the only way to test whether their predictions really hold.

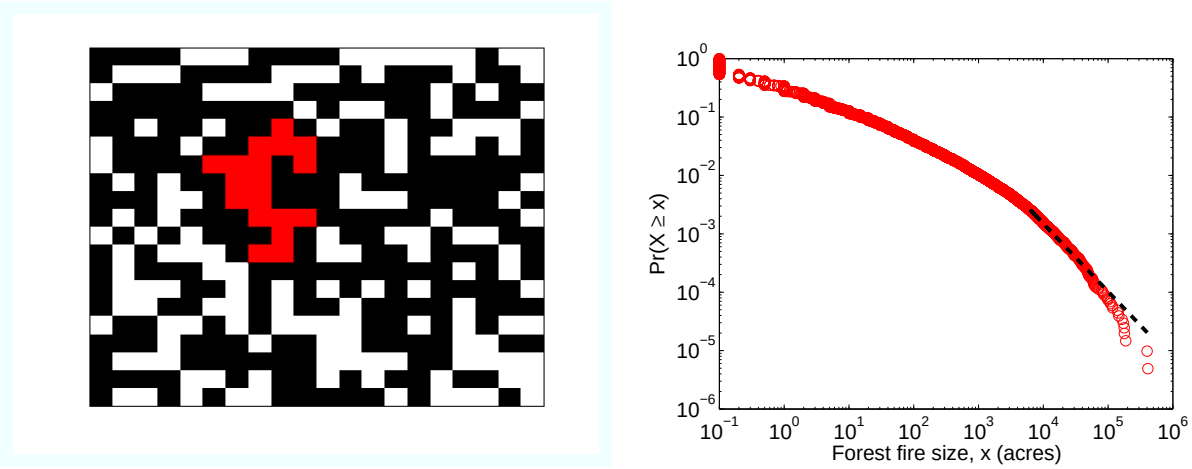

Figure 4: (a) An example simulation of the forest fire model (on a 20×20 grid), with a burning cluster of trees highlighted in red; in the next step, these cells convert to "empty" (black). (b) The empirical ccdf for the number of acres burned in forest fires on U.S. federal land, between 1986 and 1996 (data from the National Fire Occurrence Database, USDA Forest Service and Department of the Interior), along with a power-law fit to the upper tail.

2.4 Other mechanisms

There are now a very large number of mechanisms known that produce power-law distributions. For instance:

- optimization of information content in language (originally due to Mandelbrot)
- monkeys hitting random keys on a keyboard, even if they hit keys with unequal probabilities[12]
- certain kinds of duplication processes, including a model of duplication-mutation-and-complementarity put forward for gene networks and a model of duplication-with-modification put forward for the distribution of file sizes on your computer
- certain kinds of constrained multiplicative random walks, which are often used to model the wealth distribution but where debt (negative wealth) is disallowed.
- inverses of physical quantities
- the Yule process model of macroevolutionary speciation, which is closely related to de Solla Price's model of citation counts, which is also sometimes called "preferential attachment"
- phase transitions, percolation and critical phenomena in physics
- "coherent noise", a model of biological extinction due to Sneppen and Newman
- extremal processes and record breaking
- etc. etc. etc.

Two excellent, although now slightly out of date, surveys that describe these and several other mechanisms are

- M.E.J. Newman, "Power laws, Pareto distributions and Zipf's law." *Contemporary Physics* **46**(5), 323–351 (2005), and
- M. Mitzenmacher, "A Brief History of Generative Models for Power Law and Lognormal Distributions." *Internet Mathematics* **1**(2), 226-251 (2004).

[12] Amusing, some primatologist apparently tried this experiment literally (which, of course, did not draw on any of their NSF funding). They found that monkeys are not very good random number generators, and often defecate on the typewriter instead of diligently trying to write Shakespeare.

3 Matlab code for Forest Fire model

If you want to try out the forest fire model, here is Matlab code that simulates and visualizes it.

```
% -- forest fire model simulation
% -- adapted from http://courses.cit.cornell.edu/bionb441/ (Fall 2006)

% model parameters
n          = 100;      % grid size
Plightning = 0.000005; % p(tree  -> fire)
Pgrowth    = 0.01;     % p(empty -> tree)

% model data structures
z   = zeros(n,n);
o   = ones(n,n);
veg = z;
sum = z;

% visualize the simulation
figure(1);
imh = image(cat(3,z,veg*.02,z));
set(imh, 'erasemode', 'none')
axis equal
axis tight

% burning -> empty
% green   -> burning if one neigbor burning or with prob=f (lightning)
% empty   -> green with prob=p (growth)
% veg      = {empty=0 burning=1 green=2}
for i=1:3000
    %nearby fires?
    sum = (veg(1:n,[n 1:n-1])==1) + (veg(1:n,[2:n 1])==1) + ...
           (veg([n 1:n-1], 1:n)==1) + (veg([2:n 1],1:n)==1) ;

    veg = 2*(veg==2) - ((veg==2) & (sum>0 | (rand(n,n)<Plightning))) + ...
          2*((veg==0) & rand(n,n)<Pgrowth) ;

    set(imh, 'cdata', cat(3,(veg==1),(veg==2),z) )
    drawnow
end
```

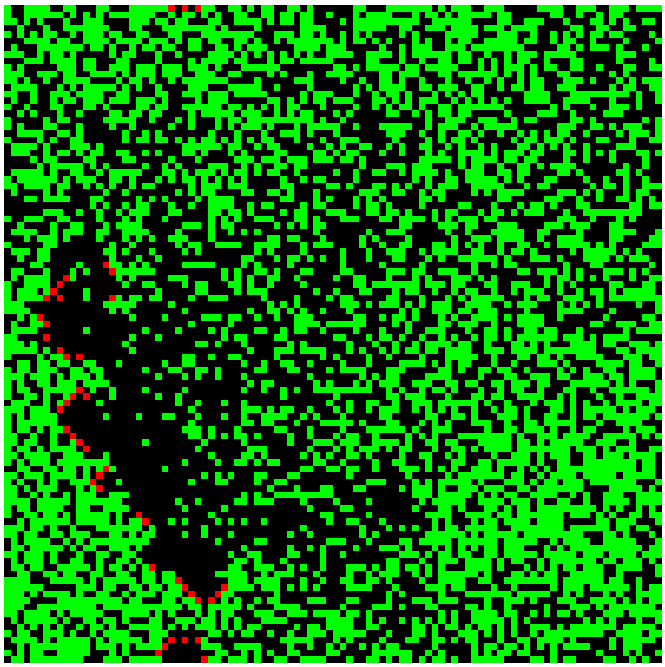

Figure 5: A snapshot from the simulation: black squares are "empty," green are "trees," and red are "burning."