Chaos, Complexity, and Inference (36-462)

Lecture 15: Estimating Heavy-Tailed Distributions

Cosma Shalizi

3 March 2009

Estimating Heavy-Tailed Distributions

Maximum likelihood The good way to get power law parameter estimates

Log-log regression The bad way to get power law parameter estimates

Non-parametric density estimation Do you care if you *have* a power law?

Further reading: Clauset *et al.* (2009)

Maximum Likelihood

Start with the Pareto (continuous) case probability density:

$$p(x; \alpha, x_{\min}) = \frac{\alpha - 1}{x_{\min}} \left(\frac{x}{x_{\min}} \right)^{-\alpha}$$

Assuming IID samples, log-likelihood is easy

$$\mathcal{L}(\alpha, x_{\min}) = n \log \frac{\alpha - 1}{x_{\min}} - \alpha \sum_{i=1}^{n} \log \frac{x_i}{x_{\min}}$$

Take derivative and set equal to zero at the MLE:

$$\begin{aligned}
\frac{\partial}{\partial \alpha}\mathcal{L} &= \frac{n}{\alpha - 1} - \sum_{i=1}^{n} \log x_i / x_{\min} \\
\hat{\alpha} &= 1 + \frac{n}{\sum_{i=1}^{n} \log x_i / x_{\min}}
\end{aligned}$$

What about $x_{\min}$? If we know that it's really a Pareto, then the MLE for $x_{\min}$ is $\min x_i$. Otherwise, see later.

Zipf or Zeta Distribution

Same story:

$$p(x; \alpha, x_{\min}) = \frac{x^{-\alpha}}{\zeta(\alpha, x_{\min})}$$

$$\mathcal{L}(\alpha, x_{\min}) = -n \log \zeta(\alpha, x_{\min}) - \alpha \sum_{i=1}^{n} \log x_i$$

$$\frac{\zeta'(\hat{\alpha}, x_{\min})}{\zeta(\hat{\alpha}, x_{\min})} = -\frac{1}{n} \sum_{i=1}^{n} \log x_i$$

In practice it's easier to just maximize $\mathcal{L}$ numerically than to solve that equation

When $x_{\min} > 6$ or so,

$$\hat{\alpha} \approx 1 + \frac{n}{\sum_{i=1}^{n} \log \frac{x_i}{x_{\min} - 0.5}}$$

Result due to M. E. J. Newman, see Clauset *et al.* (2009)

Properties of the MLE

1. Consistency: easiest to see for Pareto. By LLN,

$$\frac{1}{n}\sum_{i=1}^{n} \log x_i/x_{\min} \rightarrow \mathbf{E}\left[\log X/x_{\min}\right] = \frac{1}{\alpha - 1}$$

so $\hat{\alpha} \rightarrow \alpha$; similarly for Zipf

2. Standard error

$$\mathrm{Var}\left[\hat{\alpha}\right] = \frac{(\alpha-1)^2}{n} + O(n^{-2})$$

Can plug in $\hat{\alpha}$, or do jack-knife or bootstrap

Nonparametric Bootstrap in One Slide

Wanted: sampling distribution of some estimator $\hat{G}$ of a functional G of a distribution F (e.g., a parameter)
Given: data $x_1, x_2, \ldots x_n$, all assumed IID from F
Procedure: draw n samples, *with replacement*, from data, giving $b_1, b_2, \ldots b_n$
Calculate $\hat{G}(b_1, \ldots b_n) = \hat{G}_b$
Repeat many times
Empirical distribution of $\hat{G}_b$ is about the sampling distribution of $\hat{G}$
(some conditions apply)

Properties of the MLE (continued)

3. Asymptotically Gaussian and efficient:

$$\hat{\alpha} \rightsquigarrow \mathcal{N}(\alpha, \frac{(\alpha-1)^2}{n})$$

and this is the fastest rate of convergence
4. (Pareto) If $x_{\min}$ is known or fixed, $(\hat{\alpha}-1)/n$ has an inverse gamma distribution, which gives exact confidence intervals

Log-Log Regression

Recall that for a power law

$$\begin{aligned} F^{\uparrow}(x) = \Pr(X \geq x) & \propto & x^{-(\alpha-1)} \\ \log F^{\uparrow}(x) & \propto & C - (\alpha - 1)\log x \end{aligned}$$

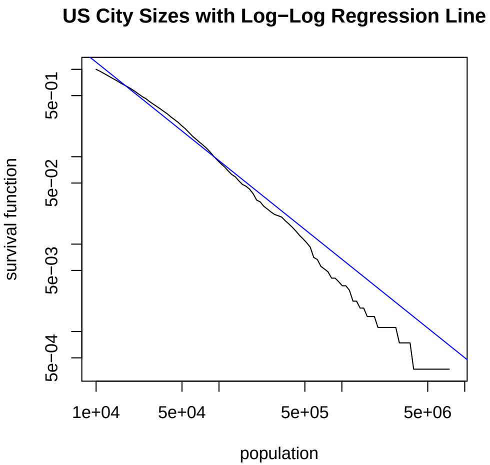
Empirical survival function:

$$\hat{F}_n^{\uparrow}(x) \equiv \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{[x,\infty)}(x_i)$$

As $n \to \infty$, $\hat{F}_n^{\uparrow}(x) \to F^{\uparrow}(x)$.
Estimate α by linearly regressing $\log \hat{F}_n^{\uparrow}(x)$ on $\log x$.

History

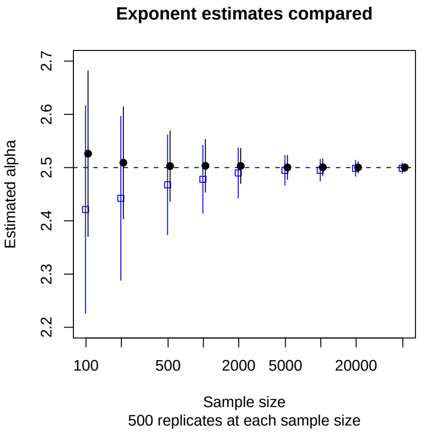
First real investigation of power law data came with Villfredo Pareto's work on economic inequality in 1890s
Used log-log regression
Taken up by Zipf in 1920s–1940s
Very widely used in physics, computer science, etc.


US City Sizes with Log−Log Regression Line
survival function
5e−01
5e−02
5e−03
5e−04
1e+04
5e+04
5e+05
5e+06
population

If the data really come from a power law, this is consistent:

$$\hat{\alpha}_{LLR} \rightarrow \alpha$$

but this doesn't say *how fast* it converges, and in fact the errors are large and persistent (compared to MLE)



Simulated from $\mathrm{Pareto}(2.5, 1)$; blue = regression, black = MLE (shifted a bit for clarity); mean $\hat{\alpha} \pm$ standard deviation

Why This Is Bad: Improperly Normalized

Notice that $F^{\uparrow}(x_{\min}) = 1$, so true log survival function crosses 0 at $x_{\min}$

But least-squares line does not do so in general! $\Rightarrow$ estimated function cannot be a probability distribution!

Could do constrained linear regression — but somehow you never see that

Why This Is Bad: Wrong Error Estimates

Usual formulas for standard errors in regression assume Gaussian noise

$$Y = \beta_0 + \beta_1 Z + \epsilon, \ \epsilon \sim \mathcal{N}(0, \sigma^2)$$

so using those formulas here means you're assuming *log-normal* noise for $\hat{F}_n^{\uparrow}$
and the central limit theorem says $\hat{F}_n^{\uparrow}$ has *Gaussian* noise
i.e., the usual formulas *do not apply* here
Can get error estimates (if you must) by bootstrap

Why This Is Bad: Lack of Power

People often point to a high R^2 for the regression as a sign that it must be right

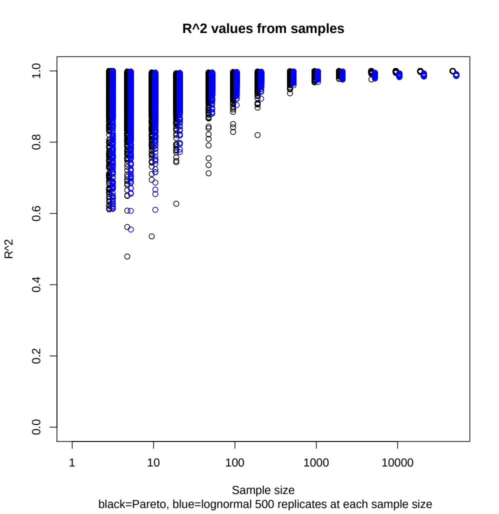
This is *always* foolish when it comes to regression

This is *especially* foolish here — distributions like log-normal have very high R^2 even with *infinite* samples

The R^2 test lacks **power** and **severity** against such alternatives

Example: Log-Log Regressions of Power Laws and Log-Normals

Simulated from $\mathrm{Pareto}(2.5, 5)$ and $\log\mathcal{N}(0.6626308, 0.65393343)$ — chosen to come close to the former. (Also, simulated values < 5 discarded.)
Did log-log regression for both, plot shows distribution of R^2 values from simulations.



Note: R^2 stabilizes at over 0.9 for the log-normal!

Log-Log Regression on the Histogram

Bad as log-log regression of the survival function is, it's still better than log-log regression of the histogram

1. Loss of information (unlike survival function)
2. Results depend on choice of bins for histogram
3. Even bigger normalization issues
4. Even worse errors — comparatively larger fluctuations, especially in the tail where they have the most leverage on the regression

There *may* be times when log-log regression on the survival function is reasonable (though I can't think of any); there are none when log-log regression on the histogram is

Conclusions about Log-Log Regression

1. Do not use it.
2. Do not believe papers using it.

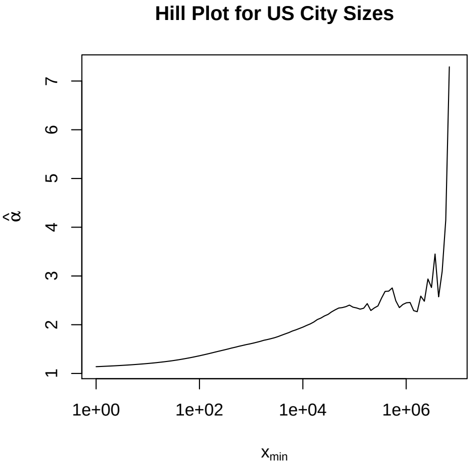
Estimating $x_{\min}$

Need to estimate $x_{\min}$
Simple for a pure power law: to maximize likelihood,
$\widehat{x}_{\min} = \min x_i$
Not useful when it is only the *tail* which follows a power law

Hill Plots

One approach: try various $x_{\min}$, plot $\hat{\alpha}$ vs. $x_{\min}$, look for stable region
Called "Hill plot" after Hill (1975)
Also gives an idea of fragility of results

```
> hill.estimator <- function(xmin,data)
  {pareto.fit(data,xmin)$exponent}
> hill.plotter <- function(xmin,data)
  {sapply(xmin,hill.estimator,data=data)}
> curve(hill.plotter(x,cities),from=min(cities),
  to=max(cities),log="x",
  main="Hill Plot for US City Sizes",
  xlab=expression(x_min),
  ylab=expression(alpha))
```



Note log scale of horizontal axis

Kolmogorov-Smirnov Distance

Kolmogorov-Smirnov statistic: measure of distance between one-dimensional cumulative distribution functions

$$D_{KS}(F, G) = \sup_x |F(x) - G(x)|$$

Here look at

$$D_{KS}(x_{\min}) = \sup_{x \geq x_{\min}} |\hat{F}_n^{\uparrow}(x) - P(x; \hat{\alpha}, x_{\min})|$$

where $P(x; \hat{\alpha}, x_{\min})$ is the Pareto survival function we get by assuming a given $x_{\min}$ and estimating

Estimate $x_{\min}$ by Minimizing D_{KS}

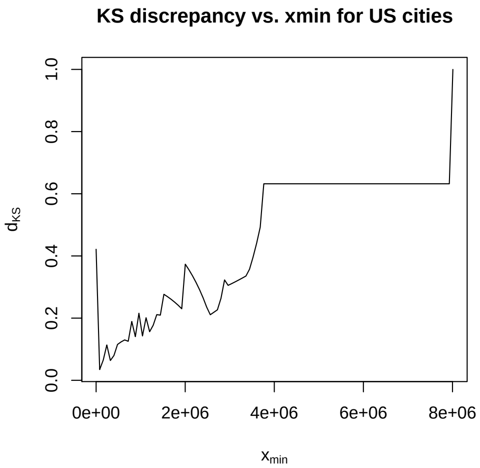
Pick the $x_{\min}$ where the distance between data and estimated distribution is smallest

$$\widehat{x}_{\min} = \operatorname*{argmin}_{x_{\min} \in x_i} D_{KS}(x_{\min})$$

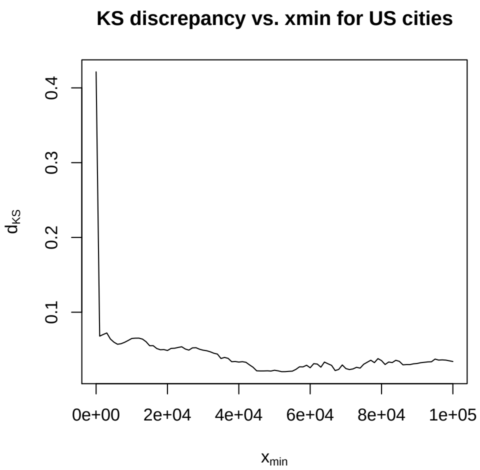
Only considering actual data values is faster and seems to not miss anything

Another principled approach: BIC (Handcock and Jones, 2004), *but* we find that works slightly less well than minimum KS (Clauset *et al.*, 2009)

```
> ks.test.for.pareto <- function(threshold,data) {
    model <- pareto.fit(data,threshold)
    d <- ks.test(data[data>=threshold],ppareto,
      threshold=threshold,exponent=model$exponent)
   return(as.vector(d$statistic)) }
> ks.test.for.pareto.vectorized <- function(threshold,data)
  { sapply(threshold,ks.test.for.pareto,data=data) }
> curve(ks.test.for.pareto.vectorized(x,cities),
    from=min(cities),to=max(cities),
    xlab=expression(x[min]),ylab=expression(d[KS]),
    main="KS discrepancy vs. xmin for US cities")
```



$\widehat{x}_{\min}$ is evidently small



$\widehat{x}_{\min} = 5.246 \times 10^4$

Properties

1. In simulations, when there *is* a power law tail, this is good at finding it
2. When there isn't a distinct tail but there is an asymptotic exponent, choses $x_{\min}$ such that $\hat{\alpha}$ becomes right
3. Error estimates: bootstrap

Nonparametric Density Estimation as an Alternative

All of this is *assuming* a power-law tail, i.e., parametric form

Often this is neither justified nor important, but estimating the distribution is

Can then use non-parametric density estimation

Kernel Density Estimation in One Slide

Data $x_1, x_2, \ldots x_n$

$$\hat{p}_n(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}K\left(\frac{x - x_i}{h_n}\right)$$

where **kernel** K has $K \geq 0$, $\int K(x)dx = 1$, $\int xK(x)dx = 0$,
$0 < \int x^2K(x)dx < \infty$
and **bandwidth** $h_n \to 0, nh_n \to \infty$ as $n \to \infty$
common choice of K: standard Gaussian density ϕ
Ideally, $h_n = O(n^{-1/3})$
Generally, pick h_n by cross-validation

basic `R` command: `density`

better: use the `np` package from CRAN (Hayfield and Racine, 2008)

Ordinary non-parametric estimation works poorly with heavy-tailed data, since it generally produces light tails
Special methods exist, e.g.:

- transform data so $[0, \infty) \mapsto [0, 1]$ monotonically
  e.g., $\frac{2}{\pi} \arctan x$
- do ordinary density estimation on transformed data
  being careful to keep $\Pr([0, 1]) = 1$
- apply reverse transformation to estimated density

See Markovitch and Krieger (2000) or Markovich (2007) (harder to read)

Next time: how to tell the difference between power laws and other distributions

Clauset, Aaron, Cosma Rohilla Shalizi and M. E. J. Newman (2009). "Power-law distributions in empirical data." *SIAM Review*, **forthcoming**. URL `http://arxiv.org/abs/0706.1062`.

Handcock, Mark S. and James Holland Jones (2004). "Likelihood-based inference for stochastic models of sexual network formation." *Theoretical Population Biology*, **65**: 413–422. URL `http://www.csss.washington.edu/Papers/wp29.pdf`.

Hayfield, Tristen and Jeffrey S. Racine (2008). "Nonparametric Econometrics: The `np` Package." *Journal of Statistical Software*, **27(5)**: 1–32. URL `http://www.jstatsoft.org/v27/i05`.

Hill, B. M. (1975). "A simple general approach to inference about the tail of a distribution." *Annals of Statistics*, **3**:

1163–1174. URL http://www.jstor.org/pss/2958370.

Markovich, Natalia (2007). *Nonparametric Analysis of Univariate Heavy-Tailed Data: Research and Practice*. New York: John Wiley.

Markovitch, Natalia M. and Udo R. Krieger (2000). "Nonparametric estimation of long-tailed density functions and its application to the analysis of World Wide Web traffic." *Performance Evaluation*, **42**: 205–222. doi:10.1016/S0166-5316(00)00031-6.