INFO 6010
Computional Methods
for
Information Science Research

Meeting 4: Power law vs Poisson

Paul Ginsparg

Cornell University, Ithaca, NY

15 Feb 2013

General “Big Data” Procedure

- Define a probabilistic model
  (i.e., use data to create language model, a probability distribution over all strings in the language, learned from corpus, and use model to determine probability of candidates)
- Enumerate candidates
  (e.g., segmentations, corrected spellings)
- Choose the most probable candidate:

$$\mathrm{best} = \mathrm{argmax}_c \in \mathrm{candidates}\, P(c)$$

Python: best = max(candidates, key=P)

Big Data = Simple Algorithm

Statistical Machine Translation

Google *n*-gram corpus created by researchers in the machine translation group (released 2006).

Translating from foreign language (f) into English (e) similar to correcting misspelled words.

The best English translation is modeled as:

$$\mathrm{best} = \mathrm{argmax}_e P(e|f) = \mathrm{argmax}_e P(f|e)P(e)$$

where $P(e)$ is the language model for English, which is estimated by the word *n*-gram data, and $P(f|e)$ is the translation model, learned from a bilingual corpus (where pairs of documents are marked as translations of each other). Although top systems make use of many linguistic features, including parts of speech and syntactic parses of the sentences, seems that majority of knowledge necessary for translation resides in the *n*-gram data.

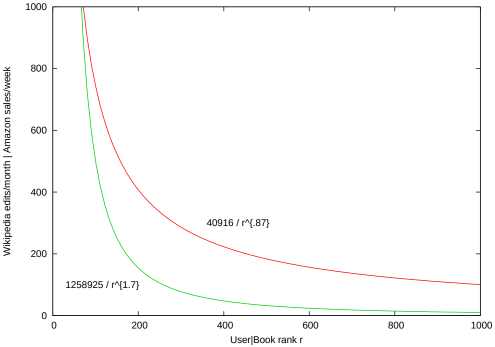
Further details in Brants,Popat,Xu,Och,Dean (2007)
"Large Language Models in Machine Translation",
http://acl.ldc.upenn.edu/D/D07/D07-1090.pdf

Power laws more generally

E.g., consider power law distributions of the form $c\, r^{-k}$, describing the number of book sales versus sales-rank r of a book, or the number of Wikipedia edits made by the r^{th} most frequent contributor to Wikipedia.

- Amazon book sales: $c\, r^{-k}$, $k \approx .87$
- number of Wikipedia edits: $c\, r^{-k}$, $k \approx 1.7$

(More on power laws and the long tail here:
Networks, Crowds, and Markets:
Reasoning About a Highly Connected World
by David Easley and Jon Kleinberg
Chpt 18: `http://www.cs.cornell.edu/home/kleinber/networks-book/networks-book-ch18.pdf`)

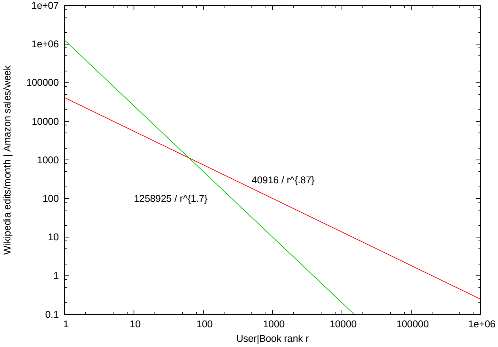


Normalization given by the roughly 1 sale/week for the 200,000th ranked Amazon title:

$$40916r^{-.87}$$

and by the 10 edits/month for the 1000th ranked Wikipedia editor:

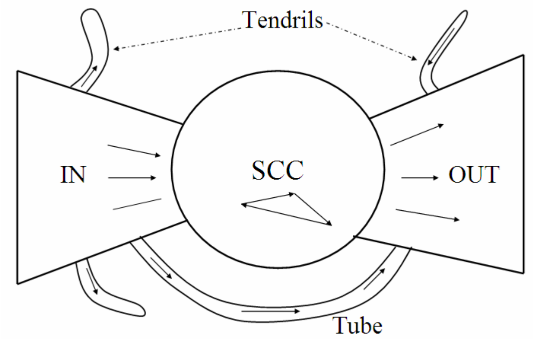
$$1258925r^{-1.7}$$



Long tail: about a quarter of Amazon book sales estimated to come from the long tail, i.e., those outside the top 100,000 bestselling titles

Bowtie structure of the web

A.Broder,R.Kumar,F.Maghoul,P.Raghavan,S.Rajagopalan,S. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. Computer Networks, 33:309–320, 2000.



- Strongly connected component (SCC) in the center
- Lots of pages that get linked to, but don't link (OUT)
- Lots of pages that link to other pages, but don't get linked to (IN)
- Tendrils, tubes, islands

# of in-links (in-degree) averages 8–15, not randomly distributed (Poissonian), instead a power law:

# pages with in-degree i is $\propto 1/i^{\alpha}$, $\alpha \approx 2.1$

Poisson Distribution

Bernoulli process with N trials, each probability p of success:

$$p(m) = \binom{N}{m} p^m (1-p)^{N-m} \ .$$

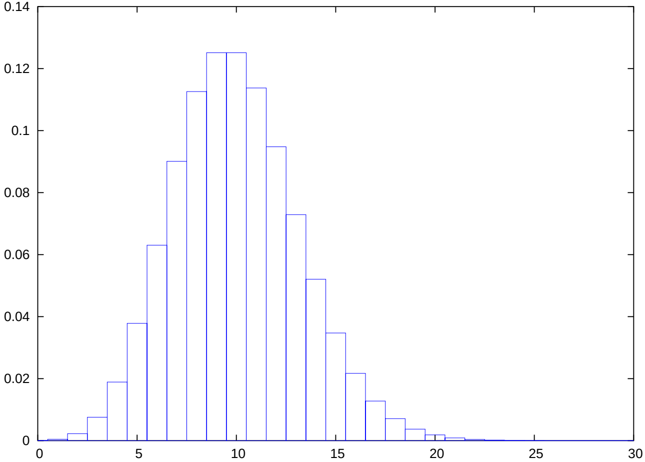
Probability $p(m)$ of m successes, in limit N very large and p small, parametrized by just $\mu = Np$ ($\mu =$ mean number of successes).
For $N \gg m$, we have $\frac{N!}{(N-m)!} = N(N-1)\cdots(N-m+1) \approx N^m$,
so $\binom{N}{m} \equiv \frac{N!}{m!(N-m)!} \approx \frac{N^m}{m!}$, and

$$p(m) \approx \frac{1}{m!} N^m \left(\frac{\mu}{N}\right)^m \left(1 - \frac{\mu}{N}\right)^{N-m} \approx \frac{\mu^m}{m!} \lim_{N\to\infty} \left(1 - \frac{\mu}{N}\right)^N = \mathrm{e}^{-\mu} \frac{\mu^m}{m!}$$

(ignore $(1-\mu/N)^{-m}$ since by assumption $N \gg \mu m$).
N dependence drops out for $N \to \infty$, with average μ fixed ($p \to 0$).
The form $p(m) = \mathrm{e}^{-\mu} \frac{\mu^m}{m!}$ is known as a Poisson distribution
(properly normalized: $\sum_{m=0}^{\infty} p(m) = \mathrm{e}^{-\mu} \sum_{m=0}^{\infty} \frac{\mu^m}{m!} = \mathrm{e}^{-\mu} \cdot \mathrm{e}^{\mu} = 1$).

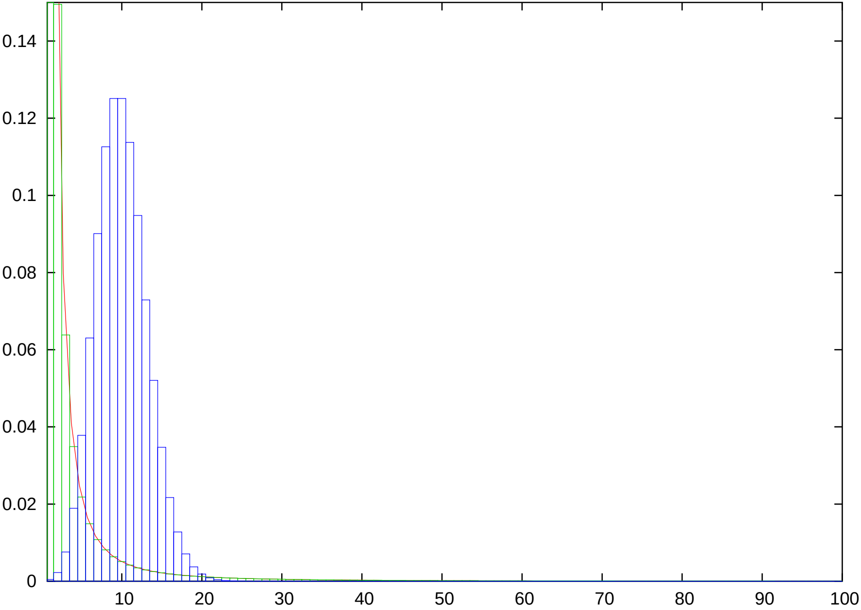
Poisson Distribution for $\mu = 10$

$$p(m) = \mathrm{e}^{-10} \frac{10^m}{m!}$$

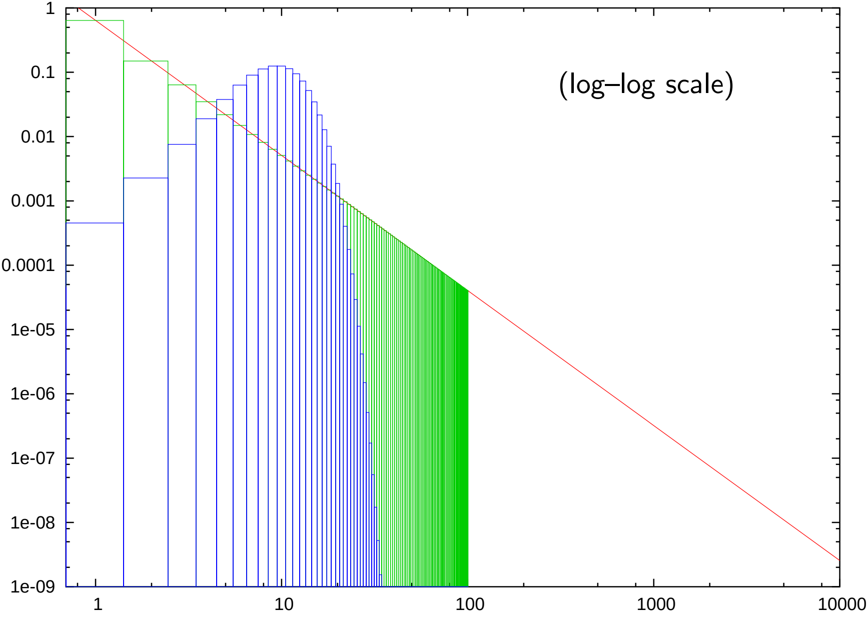


Compare to power law $p(m) \propto 1/m^{2.1}$

Power Law $p(m) \propto 1/m^{2.1}$ and Poisson $p(m) = \mathrm{e}^{-10}\frac{10^m}{m!}$



Power Law $p(m) \propto 1/m^{2.1}$ and Poisson $p(m) = \mathrm{e}^{-10}\frac{10^m}{m!}$



Why your friends . . .

Definitions:

Consider sampling N values X_i of some variable X.

Then the *expectation value* is the average: $\mathrm{E}[X] = \frac{1}{N}\sum_i X_i$.

The *variance* is defined as $\mathrm{Var}[X] = \frac{1}{N}\sum_i (X_i - \mathrm{E}[X])^2$,

and satisfies $\mathrm{Var}[X] = \mathrm{E}[X^2] - (\mathrm{E}[X])^2$.

The *standard deviation* is the square root: $\mathrm{std}[X] = \sqrt{\mathrm{Var}[X]}$

Feld 1991:

Node i has degree d_i, i.e., d_i friends.

$$total\_fof = \sum_{\text{nodes } i} \sum_{\text{friends f of i}} d_f = \sum_i d_i^2$$

(since each d_f occurs d_f times in the first double sum).

Average fof per person $= \frac{1}{N}\sum_i d_i^2 = \mathrm{E}[d^2] = \mathrm{Var}[d] + (\mathrm{E}[d])^2$

The average fof per friend $= \mathrm{E}[d^2]/\mathrm{E}[d] = \mathrm{E}[d] + \mathrm{Var}[d]/\mathrm{E}[d]$

The variance is positive, so the above is always greater than E[d].

Used: detecting flu, disease innoculation, administrative propaganda

Digression: “naive” Bayes

Spam classifier:
Imagine a training set of 2000 messages,
1000 classified as spam (S),
and 1000 classified as non-spam ($\overline{S}$).

180 of the S messages contain the word “offer”.
20 of the $\overline{S}$ messages contain the word “offer”.

Suppose you receive a message containing the word “offer”.
What is the probability it is S? Estimate:

$$\frac{180}{180+20} = \frac{9}{10}\ .$$

(Formally, assuming “flat prior” $p(S) = p(\overline{S})$:

$$p(S|\text{offer}) = \frac{p(\text{offer}|S)p(S)}{p(\text{offer}|S)p(S) + p(\text{offer}|\overline{S})p(\overline{S})} = \frac{\frac{180}{1000}}{\frac{180}{1000}+\frac{20}{1000}} = \frac{9}{10}\ .)$$

Basics of probability theory

- $A =$ event
- $0 \leq p(A) \leq 1$
- joint probability $p(A, B) = p(A \cap B)$
- conditional probability $p(A|B) = p(A, B)/p(B)$

Note $p(A, B) = p(A|B)p(B) = p(B|A)p(A)$, gives posterior probability of A after seeing the evidence B

$$\text{Bayes `Thm` :} \quad p(A|B) = \frac{p(B|A)p(A)}{p(B)}$$

In denominator, use
$p(B) = p(B, A) + p(B, \overline{A}) = p(B|A)p(A) + p(B|\overline{A})p(\overline{A})$

$$\text{Odds:} \quad O(A) = \frac{p(A)}{p(\overline{A})} = \frac{p(A)}{1 - p(A)}$$

“naive” Bayes, cont’d

Spam classifier:
Imagine a training set of 2000 messages,
1000 classified as spam (S),
and 1000 classified as non-spam ($\overline{S}$).

words $w_i = \{$“offer”, “FF0000”, “click”, “unix”, “job”, “enlarge”,$\ldots\}$
n_i of the S messages contain the word w_i.
m_i of the $\overline{S}$ messages contain the word w_i.

Suppose you receive a message containing the words
$w_1, w_4, w_5, \ldots$.
What are the odds it is S? Estimate:

$$p(S|w_1, w_4, w_5, \ldots) \propto p(w_1, w_4, w_5, \ldots|S)p(S)$$

$$p(\overline{S}|w_1, w_4, w_5, \ldots) \propto p(w_1, w_4, w_5, \ldots|\overline{S})p(\overline{S})$$

Odds are

$$\frac{p(S|w_1, w_4, w_5, \ldots)}{p(\overline{S}|w_1, w_4, w_5, \ldots)} = \frac{p(w_1, w_4, w_5, \ldots|S)p(S)}{p(w_1, w_4, w_5, \ldots|\overline{S})p(\overline{S})}$$

“naive” Bayes odds

Odds

$$\frac{p(S|w_1, w_4, w_5, \ldots)}{p(\overline{S}|w_1, w_4, w_5, \ldots)} = \frac{p(w_1, w_4, w_5, \ldots|S)p(S)}{p(w_1, w_4, w_5, \ldots|\overline{S})p(\overline{S})}$$

are approximated by

$$\approx \frac{p(w_1|S)p(w_4|S)p(w_5|S)\cdots p(w_\ell|S)p(S)}{p(w_1|\overline{S})p(w_4|\overline{S})p(w_5|\overline{S})\cdots p(w_\ell|\overline{S})p(\overline{S})}$$

$$\approx \frac{(n_1/1000)(n_4/1000)(n_5/1000)\cdots(n_\ell/1000)}{(m_1/1000)(m_4/1000)(m_5/1000)\cdots(m_\ell/1000)} = \frac{n_1 n_4 n_5 \cdots n_\ell}{m_1 m_4 m_5 \cdots m_\ell}$$

where we’ve assumed words are independent events
$p(w_1, w_4, w_5, \ldots|S) \approx p(w_1|S)p(w_4|S)p(w_5|S)\cdots p(w_\ell|S)$,
and $p(w_i|S) \approx n_i/|S|$, $p(w_i|\overline{S}) \approx m_i/|\overline{S}|$
(recall n_i and m_i, respectively, counted the number of spam S and non-spam $\overline{S}$ training messages containing the word w_i)

“naive” Bayes log odds

Log Odds

$$\log \frac{n_1 n_4 n_5 \cdots n_\ell}{m_1 m_4 m_5 \cdots m_\ell} = \log \frac{n_1}{m_1} + \log \frac{n_4}{m_4} + \log \frac{n_5}{m_5} + \cdots + \log \frac{n_\ell}{m_\ell}$$

So calculate the fixed weights $w_i = \log(n_i/m_i)$ once and for all.

If word i occurs t_i times in a test message, log odds of S is given by $\sum_i t_i w_i$