Where we are at

- We justified an approach to lexical statistics based on population models (e.g., Zipf-Mandelbrot)
- We discussed random samples and expected values
- We showed how to estimate model parameters by comparing observed / expected frequency spectrum
- We need an efficient way to calculate expected values
  - for random samples of arbitrary size $N$
  - given a model of the population type probabilities $\pi_k$

Expected $V_m$ for sample of size $N$

To calculate $E[V_m(N)]$ . . .

- Average $V_m$ over a large number ($n$) of samples, all of them having the same size $N$

\[
E[V_m(N)] \approx \frac{1}{n} \cdot (V_m^{(1)} + V_m^{(2)} + \cdots + V_m^{(n)})
\]

- Mathematically, $E[V_m(N)]$ is the limit of this expression for $n \to \infty$ (but you can just think of $n$ as very large)
Expected $V_m$ for sample of size $N$

- We know how to calculate the probability that in a sample of size $N$, a given type $w_k$ (with parameter $\pi_k$) occurs exactly $m$ times:

$$p_{k,m} := \binom{N}{m}(\pi_k)^m(1-\pi_k)^{N-m}$$

- Which means that it will be counted in class $V_m$ in approximately $n \cdot p_{k,m}$ out of $n$ samples
- if $n$ is large enough, this estimate is very accurate
- Taking the sum over all types $w_k$ and dividing by $n$:

$$E[V_m(N)] = \sum_k p_{k,m} = \sum_k \binom{N}{m}(\pi_k)^m(1-\pi_k)^{N-m}$$

Binomial sampling vs. Poisson sampling

Switch to Poisson sampling can be motivated in two ways:

- **Philosophical:**
  - Not as unreasonable as it seems: think of the frequency distribution of nouns in text sample of 1 million running words (such as the Brown corpus) → sample size $N$ (= number of noun tokens) will be different for each sample

- **Practical:**
  - When $N$ is large and $\pi$ small (as with word frequency distributions), Poisson probabilities are a very good approximation to binomial probabilities
  - In lexical statistics, word frequency distribution models almost always use Poisson expectations

Poisson expectations for $V_m$ and $V$

$$E[V_m(N)] = \sum_k \frac{(N\pi_k)^m}{m!} e^{-N\pi_k}$$

$$E[V(N)] = \sum_k (1 - e^{-N\pi_k})$$

- $E[V]$ sums over probabilities that $w_k$ occurs at least once

Now we need to plug in population model for $\pi_k$

(we will use the Zipf-Mandelbrot model, of course)
Plugging in the population model

**Zipf-Mandelbrot:**

\[
\pi_k = \frac{C}{(k + b)^a}
\]

\[
E[V_m(N)] = \sum_k \frac{(NC)^m}{(k + b)^a} \cdot e^{-\frac{NC}{k+b}}
\]

\[
E[V_m(N)] = \sum_k (1 - e^{-\frac{NC}{k+b}})
\]

This looks ugly even to a mathematician . . .

. . . and to a computer

The bad news

\[
E[V_m(N)] = \sum_k \frac{(NC)^m}{(k + b)^a} \cdot e^{-\frac{NC}{k+b}}
\]

- This looks ugly even to a mathematician
- Are we stuck?

An idea . . .

- Look back at the observed word frequency data
- Huge type frequency lists with many ties in the ranking and unstable ordering across different samples
- More robust view on the data by pooling types with the same frequency → frequency spectrum
- Perhaps we can use a similar approach for the probabilities of the population model?
Pooling type probabilities

- Different from frequency spectrum because ZM model stipulates different, unique probability $\pi_k$ for each type $k$
- Pool types with similar probabilities into cells
  - intuition: contribution to $E[V_m]$ should be similar
  - e.g. for $\pi_k = .02501$ vs. $\pi_j = .02504$
- histogram for the distribution of type probabilities

- $L = 1000$ cells
- cell $j$ represents types with $\pi_k \approx j/L$
- cell count $c_j = \text{area}$ of bar in histogram

Plugging in, 2nd attempt

- Shorter summation for small $L \rightarrow$ easier to calculate
- But then it is only a coarse approximation:
  - for $L = 1000$, we pool all types with $\pi_k < .001$ together
  - some occur once in a million words, some once in 100 million words, some only once in a billion words
- We can refine the histogram, i.e. increase number $L$ of cells, but then the summation becomes expensive again
- The real advantage: we have moved the population model equation from $\pi_k$ to $c_j$, and thus out of the exponential and power functions
  - this makes it much easier to plug in a population model

$$E[V_m(N)] = \left( \frac{N}{L} \right)^m \cdot \left( \sum_{j=1}^{L} \frac{j^m}{m!} e^{-\frac{N}{L} j} \cdot c_j \right)$$

Refining the histogram

- $L = 1000$ cells
- $L = 2000$ cells
- $L = 5000$ cells
### LNRE models

**Baroni & Evert**

### Computing expectations

- Expectation = sample average
- Poisson sampling
- Plugging in LNRE models

### Pooling types

- Type density
- LNRE models

### Zipf-Mandelbrot as LNRE model

The problem:

- Type distribution
- The ZM & fZM

### LNRE models

**Type density**

**Zipf-Mandelbrot** as LNRE model

The problem:

- Type distribution
- The ZM & fZM

### Wrapping up

---

#### Refining the histogram

- L = 1000 cells
- L = 2000 cells
- L = 5000 cells

- **Type density function** $g(\pi) \geq 0$

#### The type density function

- Number of types $w_k$ with $A \leq \pi_k \leq B$

  \[ = \text{area under curve } g(\pi) \text{ between } A \text{ and } B \]

  \[ = \int_A^B g(\pi) d\pi \]

#### The integral form of expectations

\[
E[V_m(N)] = \sum_{j=1}^L \left( \frac{\binom{Nj}{m}}{m!} \cdot e^{-Nj} \cdot cj \right)
\]

- Mathematically, for $L \to \infty$ this converges to an integral, with $j/L \leftrightarrow \pi$ and $cj \leftrightarrow g(\pi) d\pi$:

\[
E[V_m(N)] = \int_0^1 \left( \frac{(N\pi)^m}{m!} \cdot e^{-N\pi} \cdot g(\pi) \right) d\pi
\]

- Beautiful! :-)

#### Summary time

**What did we just do?**

- Initial formula was too complex
- Histogram approximation: simpler but coarse
- Get nuances back by increasing number of cells
- . . . but this time we end up with a convenient integral that we can compute efficiently!
We can plug in any function \( g \) defined on \([0, 1]\).

Population model expressed in terms of a type density function \( g \) is what we call a LNRE model (for Large Number of Rare Events, Baayen 2001).

You can't just use any old function, of course — \( g \) must satisfy the following conditions:

\[
\begin{align*}
& g \geq 0 \\
& \int_0^1 \pi \cdot g(\pi) \, d\pi = 1
\end{align*}
\]

Do they look familiar to you?

Moreover, we want to use a function that can be derived from a plausible population model, e.g. Zipf-Mandelbrot.

The Zipf-Mandelbrot law as a LNRE model

We need to reformulate the Zipf-Mandelbrot law in terms of a type density function (to calculate expectations).

ZM has 2 parameters (and fZM has 3 parameters)

- type density function will also have parameters
  - same number of parameters, but different interpretation
  - cannot use parameter values of the population model!

Goal is to find a function \( g(\pi) \) that corresponds to a very fine histogram of the ZM (or fZM) type population.
Zipf-Mandelbrot as a LNRE model

- Find a function $g(\pi)$ that matches a very fine histogram of the Zipf-Mandelbrot law (as a population model)
- This could be done directly by trial and error for every possible combination of ZM parameters $a$ and $b$: ugly
  - we don’t even know which family of functions to use
  - there must be a better way!
- Luckily, there is an analytical solution

Meet $G$, the type distribution

- There is a way to derive ZM’s $g$ analytically
  ... but it requires another detour
- We can easily calculate the number of types with $\pi \geq \rho$, which we call the **type distribution** $G(\rho)$
- According to the ZM law, for $\rho = \pi_k$ there are exactly $k$ types with $\pi \geq \rho$ (viz. the types $w_1, \ldots, w_k$), i.e.:
  $$G(\pi_k) = k$$
- From this equation we will be able to work out $G$
- With the help of $G$ we can then derive the LNRE formulation of ZM in terms of a type density function $g$
  - NB: upper case $G$ stands for the type distribution, lower case $g$ for the type density function (standard notation)

Summary of the next few steps ... for the less mathematically inclined among us

- Plug together $g(\pi)$ and the ZM law for $\pi_k$
- Math happens
- Out comes ZM formulated in terms of $g(\pi)$
- And now ... another detour (sorry!)

Sneak preview: from $G$ to $g$

- $G(\rho) = \int^1_\rho g(\pi) \, d\pi$
- $\int^B_A g(\pi) \, d\pi =$ number of types with $A \leq \pi_k \leq B$
- $G(\rho) =$ number of types with $\rho \leq \pi_k$
- there are no types with $\pi_k > 1$
- $G' = -g$, or equivalently $g = -G'$
- This is the second fundamental theorem of calculus
- Intuitively:
  - If you increase $\rho$, say from $\rho$ to $\rho + x$, $G$ decreases (fewer types → minus sign)
  - The **amount** by which it decreases (number of types between $\rho$ and $\rho + x$) is proportional to $g(\rho)$
Calculating \( G \) from the Zipf-Mandelbrot law

- According to the ZM law, for \( \rho = \pi_k \) there are exactly \( k \) types with \( \pi \geq \rho \) (viz. the types \( w_1, \ldots, w_k \)), i.e.:

\[
G(\pi_k) = k
\]

- Insert ZM formula for the type probabilities \( \pi_k \):

\[
G \left( \frac{C}{(k + b)^a} \right) = k
\]

\( \text{find a function } G \text{ that satisfies this equation} \)

\( \text{err} \ldots \)

From \( G \) to \( g \)

\[
g(\pi) = -G'(\pi) \quad \text{with} \quad G(\pi) = C^\frac{1}{a} \cdot \pi^{-\frac{1}{a} - 1} \]

\( \text{(trivial) math happens} \)

\[
g(\pi) = \left( \frac{C^\frac{1}{a}}{a} \right) \cdot \pi^{-\frac{1}{a} - 1}
\]

\( \text{simplify by renaming constants:} \)

\[
g(\pi) = C^* \cdot \pi^{-\alpha - 1}
\]

\( \alpha = \frac{1}{2} \) replaces ZM’s \( a \) as “slope” parameter \((0 < \alpha < 1)\)

\( C^* \) is normalizing constant determined from constraint

\[
\int_0^1 \pi \cdot g(\pi) \, d\pi = 1
\]

Calculating \( G \) from the Zipf-Mandelbrot law

\[
G \left( \frac{C}{(k + b)^a} \right) = k
\]

\( \text{ZM: } k \mapsto \pi_k = \frac{C}{(k + b)^a} \iff G: \pi_k \mapsto k \)

\( \text{to get back from } \pi_k \text{ to } k, \text{ all we have to do is to solve the Zipf-Mandelbrot equation for } k, \text{ obtaining:} \)

\[
k = C^\frac{1}{a} \cdot (\pi_k)^{-\frac{1}{a} - 1} - b
\]

\( \text{we can now define } G \text{ by} \)

\[
G(\rho) := C^\frac{1}{a} \cdot \rho^{-\frac{1}{a} - 1} - b
\]

and have found a function that satisfies \( G(\pi_k) = k \)

The cutoff parameter \( B \)

\( \text{we are not quite done yet: we lost one parameter } (b) \)

\[
g(\pi) = C^* \cdot \pi^{-\alpha - 1}
\]

\( \text{according to the Zipf-Mandelbrot law, there are no types with } \pi > \pi_1 \text{ (where typically } \pi_1 \ll 1) \text{, but } g(\pi = 1) > 0 \text{ no matter what value } \alpha \text{ takes} \)

\( \text{we need an “upper threshold” parameter} \)

\( \text{obvious choice: } \pi_1, \text{ but for mathematical reasons the threshold parameter } B \text{ close rather than equal to } \pi_1 \)

\( \text{surprise, surprise: } B = \frac{a-1}{b} \)

\( b \) is back!
The LNRE ZM model

\[ g(\pi) = \begin{cases} 
C \cdot \pi^{-\alpha - 1} & 0 \leq \pi \leq B \\
0 & \pi > B 
\end{cases} \]

- shape parameter \(0 < \alpha < 1\) ("slope")
- (upper) cutoff parameter \(0 < B \leq 1\)
- \(C = \frac{1 - \alpha}{B^{1-\alpha}}\)
- relation to Zipf-Mandelbrot law:
  \[ a = \frac{1}{\alpha} \quad S = \frac{1 - \alpha}{\alpha} \cdot \frac{A^{-\alpha} - B^{-\alpha}}{B^{1-\alpha} - A^{1-\alpha}} \]

\[ E[V_m(N)] = \int_0^1 \frac{(N\pi)^m}{m!} e^{-N\pi} g(\pi) d\pi \]

- \(E[V_m(N)] = \frac{C}{m!} \cdot \int_0^B (N\pi)^m e^{-N\pi \pi^{-\alpha - 1}} d\pi \)
- \(= \ldots = \frac{C}{m!} \cdot N^a \cdot \gamma(m - \alpha, NB)\)

- The (lower) incomplete Gamma function \(\gamma\) is a so-called special function well-understood by mathematicians
- \(\gamma\) and \(m! = \Gamma(m + 1)\) can be computed efficiently
- This and several similar properties make the LNRE formulations of ZM and fZM convenient and robust

The LNRE fZM model

\[ g(\pi) = \begin{cases} 
C \cdot \pi^{-\alpha - 1} & A \leq \pi \leq B \\
0 & \text{otherwise} 
\end{cases} \]

- shape parameter \(0 < \alpha < 1\) ("slope")
- cutoff parameters \(0 < A < B \leq 1\)
  - fZM with \(A = 0\) \(\rightarrow\) ZM model
- \(C = \frac{1 - \alpha}{B^{1-\alpha} - A^{1-\alpha}}\)
- relation to Zipf-Mandelbrot law:
  \[ a = \frac{1}{\alpha} \quad S = \frac{1 - \alpha}{\alpha} \cdot \frac{A^{-\alpha} - B^{-\alpha}}{B^{1-\alpha} - A^{1-\alpha}} \]

\[ b = \frac{C}{B^\alpha \cdot \alpha} \]

Outline

- Computing expectations from the population model
- The type density function and LNRE modeling
- Zipf-Mandelbrot as LNRE model
- Wrapping up
Wrapping up

- Wake up! Math is done
- In principle, you can forget about all this and use LNRE models as black boxes (says Marco)
- However...

Things it would be good for you to remember

- In order to apply LNRE model to real-life data you need a way to estimate model parameters (typically by matching expected and observed frequency spectrum)
- Aspects you might actively intervene in:
  - choose a LNRE model
  - details of parameter estimation (cost function etc.)

Performing a LNRE analysis in zipR

- `spc <- read.spc("Brown100k.spc")`
  - load observed frequency spectrum from file
- `model <- lnre("zm", spc)`
  - pick ZM model and estimate parameters from spectrum
- `summary(model)`
  - displays model parameters & goodness-of-fit
- `EV(model, 1e+6)`
  - expected V at 1 million word sample size
- `spc.exp <- lnre.spc(model, 1e+6)`
  - expected spectrum at 1 million word sample size
- `plot(spc.exp)`
  - plot expected spectrum

LNRE models
Baroni & Evert
Computing expectations
Expectation \( \equiv \) sample average
Poisson sampling
Plugging in ZM
LNRE models
Pooling types
Type density
LNRE models
Zipf-Mandelbrot as LNRE model
The problem
Type distribution
Zipf-Mandelbrot
The ZM & GZM LNRE models
Wrapping up

LNRE models
Baroni & Evert
Computing expectations
Expectation \( \equiv \) sample average
Poisson sampling
Plugging in ZM
LNRE models
Pooling types
Type density
LNRE models
Zipf-Mandelbrot as LNRE model
The problem
Type distribution
Zipf-Mandelbrot
The ZM & GZM LNRE models
Wrapping up