Probability Generating Functions

Joel C. Miller
Institute for Disease Modeling

April 16, 2018

Context

Consider the spread of a disease starting from a single infected individual in a large population:

- A new disease emerges from some animal reservoir.
- A disease is introduced from some other population.

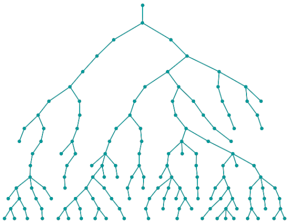
Focus on early spread, before saturation effects matter.

Context

Consider the spread of a disease starting from a single infected individual in a large population:

- A new disease emerges from some animal reservoir.
- A disease is introduced from some other population.

Focus on early spread, before saturation effects matter.



http://images.math.cnrs.fr/La-probabilite-d-extinction-d-une.html

Outline

- Quick taste of PGF properties
- Overview of things we can calculate with PGFs (and inference implications)
- A Python implementation

Probability Generating Functions

Joel Miller @joel_c_miller · Mar 14

Everything you ever wanted to know about using #ProbabilityGeneratingFunctions to study #InfectiousDisease : arxiv.org/abs/1803.05136 (I am contemplating turning this into a textbook on PGFs if I find interested collaborators who use PGFs in other biological systems)

5 28 54

Joel Miller @joel_c_miller · Mar 14

Apparently I'm the only person who thinks Probability Generating Functions deserve a hashtag!! 😀

2 5

Christophe Fraser
@ChristoPhraser

Following

Replying to @joel_c_miller

Definitely happy to follow #ProbabilityGeneratingFunctions ! They rock, though I still haven't quite got past the 'this is just weird magic' stage

Demagickifying PGFs — Alea iacta est

- Consider one infected individual who rolls a normal 6-sided die to figure out how many to infect.

Demagickifying PGFs — Alea iacta est

- Consider one infected individual who rolls a normal 6-sided die to figure out how many to infect.

	⚀	⚁	⚂	⚃	⚄	⚅
Number infected	1	2	3	4	5	6

Demagickifying PGFs — Alea iacta est

- Consider one infected individual who rolls a normal 6-sided die to figure out how many to infect.

	⚀	⚁	⚂	⚃	⚄	⚅
Number infected	1	2	3	4	5	6

- Then

$$\mu(x) = \sum P(i)x^i = \frac{1}{6}(x + x^2 + x^3 + x^4 + x^5 + x^6)$$

encodes the distribution of the infections caused.

PGF for multiple infectors

- Now consider two such individuals. The PGF of the combined number of infections caused

	⚀	⚁	⚂	⚃	⚄	⚅
⚀	2	3	4	5	6	7
⚁	3	4	5	6	7	8
⚂	4	5	6	7	8	9
⚃	5	6	7	8	9	10
⚄	6	7	8	9	10	11
⚅	7	8	9	10	11	12

Coefficient of x^5 is $4/36$

PGF for multiple infectors

- Now consider two such individuals. The PGF of the combined number of infections caused is the product $[\mu(x)]^2$.

	⚀	⚁	⚂	⚃	⚄	⚅
⚀	2	3	4	5	6	7
⚁	3	4	5	6	7	8
⚂	4	5	6	7	8	9
⚃	5	6	7	8	9	10
⚄	6	7	8	9	10	11
⚅	7	8	9	10	11	12

Coefficient of x^5 is $4/36$

	$x^1/6$	$x^2/6$	$x^3/6$	$x^4/6$	$x^5/6$	$x^6/6$
$x^1/6$	$x^2/36$	$x^3/36$	$x^4/36$	$x^5/36$	$x^6/36$	$x^7/36$
$x^2/6$	$x^3/36$	$x^4/36$	$x^5/36$	$x^6/36$	$x^7/36$	$x^8/36$
$x^3/6$	$x^4/36$	$x^5/36$	$x^6/36$	$x^7/36$	$x^8/36$	$x^9/36$
$x^4/6$	$x^5/36$	$x^6/36$	$x^7/36$	$x^8/36$	$x^9/36$	$x^{10}/36$
$x^5/6$	$x^6/36$	$x^7/36$	$x^8/36$	$x^9/36$	$x^{10}/36$	$x^{11}/36$
$x^6/6$	$x^7/36$	$x^8/36$	$x^9/36$	$x^{10}/36$	$x^{11}/36$	$x^{12}/36$

Coefficient of x^5 is $4/36$

PGF for multiple infectors

- Now consider two such individuals. The PGF of the combined number of infections caused is the product $[\mu(x)]^2$.

	⚀	⚁	⚂	⚃	⚄	⚅
⚀	2	3	4	5	6	7
⚁	3	4	5	6	7	8
⚂	4	5	6	7	8	9
⚃	5	6	7	8	9	10
⚄	6	7	8	9	10	11
⚅	7	8	9	10	11	12

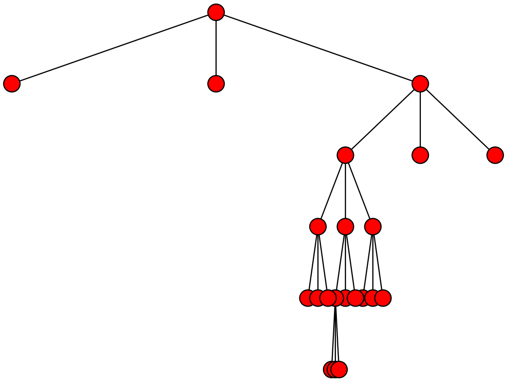
Coefficient of x^5 is $4/36$

	$x^1/6$	$x^2/6$	$x^3/6$	$x^4/6$	$x^5/6$	$x^6/6$
$x^1/6$	$x^2/36$	$x^3/36$	$x^4/36$	$x^5/36$	$x^6/36$	$x^7/36$
$x^2/6$	$x^3/36$	$x^4/36$	$x^5/36$	$x^6/36$	$x^7/36$	$x^8/36$
$x^3/6$	$x^4/36$	$x^5/36$	$x^6/36$	$x^7/36$	$x^8/36$	$x^9/36$
$x^4/6$	$x^5/36$	$x^6/36$	$x^7/36$	$x^8/36$	$x^9/36$	$x^{10}/36$
$x^5/6$	$x^6/36$	$x^7/36$	$x^8/36$	$x^9/36$	$x^{10}/36$	$x^{11}/36$
$x^6/6$	$x^7/36$	$x^8/36$	$x^9/36$	$x^{10}/36$	$x^{11}/36$	$x^{12}/36$

Coefficient of x^5 is $4/36$

- The PGF of the number of infections caused by n individuals is given by $[\mu(x)]^n$.

PGF for multiple generations



PGF for multiple generations

- Starting with one infected individual, what do we expect after 2 generations of transmission?

PGF for multiple generations

- Starting with one infected individual, what do we expect after 2 generations of transmission?
- Assuming first person rolls a particular value, we write

$$\text{⚀}(x) = \mu(x),$$
$$\text{⚁}(x) = [\mu(x)]^2,$$
$$\vdots$$
$$\text{⚅}(x) = [\mu(x)]^6$$

PGF for multiple generations

- Starting with one infected individual, what do we expect after 2 generations of transmission?
- Assuming first person rolls a particular value, we write

$$\begin{aligned} \epsdice{1}(x) &= \mu(x), \\ \epsdice{2}(x) &= [\mu(x)]^2, \\ &\vdots \\ \epsdice{6}(x) &= [\mu(x)]^6 \end{aligned}$$

- For the second generation of transmission we have each with equal probability

$$\frac{1}{6}\epsdice{1}(x) + \frac{1}{6}\epsdice{2}(x) + \cdots + \frac{1}{6}\epsdice{6}(x) = \sum \frac{1}{6}[\mu(x)]^i$$

PGF for multiple generations

- Starting with one infected individual, what do we expect after 2 generations of transmission?
- Assuming first person rolls a particular value, we write

$$
\begin{aligned}
\text{⚀}(x) &= \mu(x), \\
\text{⚁}(x) &= [\mu(x)]^2, \\
&\vdots \\
\text{⚅}(x) &= [\mu(x)]^6
\end{aligned}
$$

- For the second generation of transmission we have each with equal probability

$$
\begin{aligned}
\frac{1}{6}\text{⚀}(x) + \frac{1}{6}\text{⚁}(x) + \cdots + \frac{1}{6}\text{⚅}(x) &= \sum \frac{1}{6}[\mu(x)]^i \\
&= \mu(\mu(x))
\end{aligned}
$$

PGF for multiple generations

- Starting with one infected individual, what do we expect after 2 generations of transmission?
- Assuming first person rolls a particular value, we write

$$
\begin{aligned}
\epsdice{1}(x) &= \mu(x), \\
\epsdice{2}(x) &= [\mu(x)]^2, \\
&\vdots \\
\epsdice{6}(x) &= [\mu(x)]^6
\end{aligned}
$$

- For the second generation of transmission we have each with equal probability

$$
\begin{aligned}
\frac{1}{6}\epsdice{1}(x) + \frac{1}{6}\epsdice{2}(x) + \cdots + \frac{1}{6}\epsdice{6}(x) &= \sum \frac{1}{6}[\mu(x)]^i \\
&= \mu(\mu(x))
\end{aligned}
$$

- The big takeaway is that composition of PGFs shows up naturally in getting from one generation to the next.

Other distributions

- Assume each individual causes a Poisson-distributed number of infections with rate parameter $\mathcal{R}_0$.

$$\mu(x) = \sum_{i=0}^{\infty} P(i) x^i = \sum_{i=0}^{\infty} \frac{\mathcal{R}_0{}^i e^{-\mathcal{R}_0}}{i!} x^i$$

Other distributions

- Assume each individual causes a Poisson-distributed number of infections with rate parameter $\mathcal{R}_0$.

$$\mu(x) = \sum_{i=0}^{\infty} P(i)x^i = \sum_{i=0}^{\infty} \frac{{\mathcal{R}_0}^i e^{-\mathcal{R}_0}}{i!} x^i$$

$$= e^{-\mathcal{R}_0} \sum_{i=0}^{\infty} \frac{(\mathcal{R}_0 x)^i}{i!} = e^{-\mathcal{R}_0} e^{\mathcal{R}_0 x}$$

Other distributions

- Assume each individual causes a Poisson-distributed number of infections with rate parameter $\mathcal{R}_0$.

$$\begin{aligned}
\mu(x) &= \sum_{i=0}^{\infty} P(i)x^i = \sum_{i=0}^{\infty} \frac{{\mathcal{R}_0}^i e^{-\mathcal{R}_0}}{i!} x^i \\
&= e^{-\mathcal{R}_0} \sum_{i=0}^{\infty} \frac{(\mathcal{R}_0 x)^i}{i!} = e^{-\mathcal{R}_0} e^{\mathcal{R}_0 x} \\
&= e^{\mathcal{R}_0 (x-1)}
\end{aligned}$$

Other distributions

- Assume each individual causes a Poisson-distributed number of infections with rate parameter $\mathcal{R}_0$.

$$\begin{aligned}\mu(x) &= \sum_{i=0}^{\infty} P(i)x^i = \sum_{i=0}^{\infty} \frac{{\mathcal{R}_0}^i e^{-\mathcal{R}_0}}{i!} x^i \\ &= e^{-\mathcal{R}_0} \sum_{i=0}^{\infty} \frac{(\mathcal{R}_0 x)^i}{i!} = e^{-\mathcal{R}_0} e^{\mathcal{R}_0 x} \\ &= e^{\mathcal{R}_0 (x-1)}\end{aligned}$$

- Many commonly-used distributions have compact, closed-form PGFs.

Making predictions from the PGF

- The PGF of the number infected in generation g is $\mu^{[g]}(x)$ where the superscript denotes function composition $\mu(\mu(\mu(\cdots(x)))))$.

Making predictions from the PGF

- The PGF of the number infected in generation g is $\mu^{[g]}(x)$ where the superscript denotes function composition $\mu(\mu(\mu(\cdots(x)))))$.
- The extinction probability at generation g can be found by setting $x = 0$.

$$\mu^{[g]}(0) = P(0|g) + P(1|g)0 + P(2|g)0^2 + \cdots = P(0|g)$$

Making predictions from the PGF

- The PGF of the number infected in generation g is $\mu^{[g]}(x)$ where the superscript denotes function composition $\mu(\mu(\mu(\cdots(x)))))$.
- The extinction probability at generation g can be found by setting $x = 0$.

$$\mu^{[g]}(0) = P(0|g) + P(1|g)0 + P(2|g)0^2 + \cdots = P(0|g)$$

- For $\mu(x) = (x + \cdots + x^6)/6$, this gives 0.

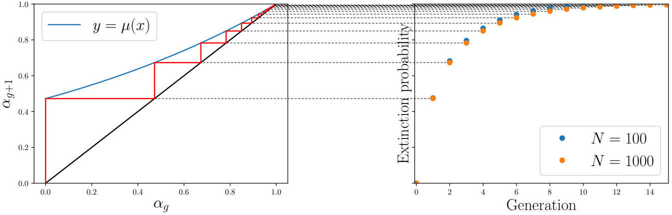
Making predictions from the PGF

- The PGF of the number infected in generation g is $\mu^{[g]}(x)$ where the superscript denotes function composition $\mu(\mu(\mu(\cdots(x)))))$.
- The extinction probability at generation g can be found by setting $x = 0$.

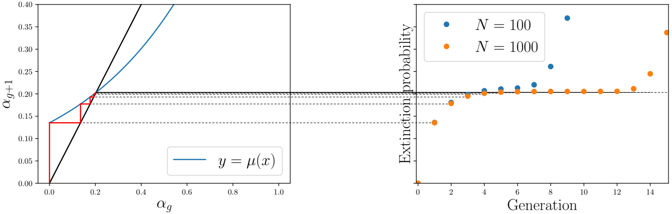
$$\mu^{[g]}(0) = P(0|g) + P(1|g)0 + P(2|g)0^2 + \cdots = P(0|g)$$

- For $\mu(x) = (x + \cdots + x^6)/6$, this gives 0.
- But for the Poisson distribution we get something different...

Poisson $\mathcal{R}_0 = 0.75$



Poisson $\mathcal{R}_0 = 2$



Black magic

Probability of 0 can be found by iterating from $x = 0$

Black magic

Probability of 0 can be found by iterating from $x = 0$

- If only there were an efficient way to calculate the coefficient of arbitrary x^i for $\mu^{[g]}(x)$...

Black magic

Probability of 0 can be found by iterating from $x = 0$

- If only there were an efficient way to calculate the coefficient of arbitrary x^i for $\mu^{[g]}(x)$...
- Given a PGF $f(x)$, we can quickly approximate the coefficient of x^n in its Taylor Series by

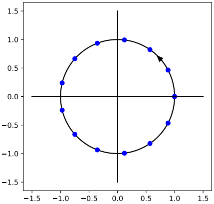
$$r_n \approx \frac{1}{M} \sum_{m=0}^{M} \frac{f(e^{2\pi i m/M})}{\left(e^{2\pi i m/M}\right)^n}$$

Black magic

Probability of 0 can be found by iterating from $x = 0$

- If only there were an efficient way to calculate the coefficient of arbitrary x^i for $\mu^{[g]}(x)$...
- Given a PGF $f(x)$, we can quickly approximate the coefficient of x^n in its Taylor Series by

$$r_n \approx \frac{1}{M} \sum_{m=0}^{M} \frac{f(e^{2\pi i m/M})}{\left(e^{2\pi i m/M}\right)^n}$$

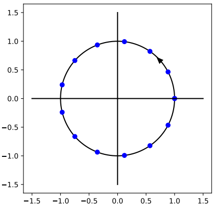


Black magic

Probability of 0 can be found by iterating from $x = 0$

- If only there were an efficient way to calculate the coefficient of arbitrary x^i for $\mu^{[g]}(x)$...
- Given a PGF $f(x)$, we can quickly approximate the coefficient of x^n in its Taylor Series by

$$r_n \approx \frac{1}{M} \sum_{m=0}^{M} \frac{f(e^{2\pi i m/M})}{(e^{2\pi i m/M})^n}$$



- This is really "just" a result from Complex Analysis about calculating residues through contour integration.

Going further

- We can also calculate the number of active and completed infections at generation g.

Going further

- We can also calculate the number of active and completed infections at generation g.
- Let $\Pi_g(y,z) = \sum p_{ir}(g) y^i z^r$ be the PGF for having i active infections and r completed infections at generation g.

Going further

- We can also calculate the number of active and completed infections at generation g.
- Let $\Pi_g(y,z) = \sum p_{ir}(g) y^i z^r$ be the PGF for having i active infections and r completed infections at generation g.
- Then $\Pi_g(y,z)$ satisfies

$$\Pi_g(y,z) = \begin{cases} y & g = 0 \\ z\mu(\Pi_{g-1}(y,z)) & g > 0 \end{cases}$$

Going further

- We can also calculate the number of active and completed infections at generation g.
- Let $\Pi_g(y,z) = \sum p_{ir}(g) y^i z^r$ be the PGF for having i active infections and r completed infections at generation g.
- Then $\Pi_g(y,z)$ satisfies

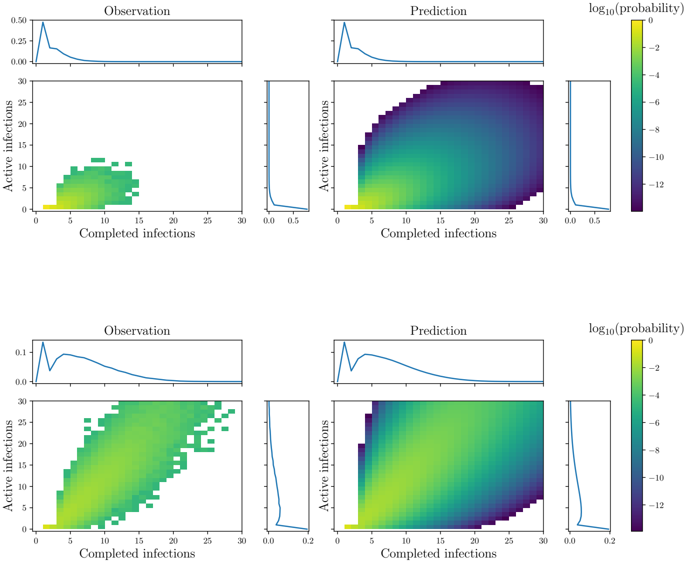
$$\Pi_g(y,z) = \begin{cases} y & g = 0 \\ z\mu(\Pi_{g-1}(y,z)) & g > 0 \end{cases}$$

- Similar contour integrations works.

Comparison with simulation (Poisson, third generation)

$\mathcal{R}_0 = 0.75$

$\mathcal{R}_0 = 2$



Final sizes

It can be shown (the proof is beautiful) that $P(\text{Final size} = j)$ equals $1/j$ times the coefficient of x^{j-1} in $[\mu(x)]^j$.

Final sizes

It can be shown (the proof is beautiful) that $P(\text{Final size} = j)$ equals $1/j$ times the coefficient of x^{j-1} in $[\mu(x)]^j$.

Distribution	**PGF**	**Probability of j infections**
Poisson	$e^{\mathcal{R}_0(x-1)}$	$\frac{(j\mathcal{R}_0)^{j-1}}{j!}e^{-j\mathcal{R}_0}$
Uniform	$x^{\mathcal{R}_0}$	$\begin{cases} 1 & j=1,\ \mathcal{R}_0=0 \\ 0 & \text{otherwise} \end{cases}$
Binomial	$(q+px)^n$	$\frac{1}{j}\binom{nj}{j-1}p^{j-1}q^{nj-j+1}$
Geometric	$p/(1-qx)$	$\frac{1}{j}\binom{2j-2}{j-1}p^j q^{j-1}$
Negative Binomial	$\left(\frac{q}{1-px}\right)^r$	$\frac{1}{j}\binom{rj+j-2}{j-1}q^{rj}p^{j-1}$

Inference

- Given an assumed offspring distribution type (often Negative Binomial), a prior on the parameters of the distribution, and observed outbreak sizes:

Inference

- Given an assumed offspring distribution type (often Negative Binomial), a prior on the parameters of the distribution, and observed outbreak sizes:
- Bayesian inference allows us to infer improved parameter values.

Inference

- Given an assumed offspring distribution type (often Negative Binomial), a prior on the parameters of the distribution, and observed outbreak sizes:
- Bayesian inference allows us to infer improved parameter values.
- See:
  - Downgrading disease transmission risk estimates using terminal importations **bioRxiv**
  - Inference of R0 and Transmission Heterogeneity from the Size Distribution of Stuttering Chains. **PLoS One**
  - Characterizing the Transmission Potential of Zoonotic Infections from Minor Outbreaks. **PLoS Comp Bio**

Python Implementation

I have created a Python package that does most of this:
https://github.com/joelmiller/Invasion_PGF

(currently not compatible with python 2.x — just need to test and push new version to github)

Example

```
>>> import Invasion_PGF as pgf
>>> def mu(x):
...     return (1. + x + x**2 + x**3)/4
...
>>> pgf.R0(mu)
1.5000001241105565
>>> #probabilities of extinction up to generation 3
>>> pgf.extinction_prob(mu, 3, intermediate_values = True)
array([ 0.        ,  0.25      ,  0.33203125,  0.36972018])
>>> #possible states in generation 3
>>> pgf.active_infections(mu, 3, 5)
array([ 0.36972018,  0.05259718,  0.07178445,  0.09609134,  0.07393309])
>>> pgf.completed_infections(mu, 3, 5)
array([ -2.04281037e-17,   2.50000000e-01,   6.25000000e-02,
         7.81250000e-02,   9.76562500e-02])
>>> pgf.active_and_completed(mu, 3, 5, 5)
array([[ 0.        ,  0.25      ,  0.0625    ,  0.03125   ,  0.015625  ],
       [ 0.        ,  0.        ,  0.        ,  0.015625  ,  0.015625  ],
       [ 0.        ,  0.        ,  0.        ,  0.015625  ,  0.01953125],
       [ 0.        ,  0.        ,  0.        ,  0.015625  ,  0.0234375 ],
       [ 0.        ,  0.        ,  0.        ,  0.        ,  0.01171875]])
```

Summary

- PGFs allow us to magically calculate a lot of properties of small outbreaks.
- We've implemented many of the relevant functions in python.
- Detailed tutorial (effectively a short textbook) available at https://arxiv.org/abs/1803.05136