Statistical Network Analysis

Prof. Dr. Ingo Scholtes

Chair of Machine Learning for Complex Networks
Center for Artificial Intelligence and Data Science (CAIDAS)
Julius-Maximilians-Universität Würzburg
Würzburg, Germany

ingo.scholtes@uni-wuerzburg.de

Lecture 07
Generating Functions

November 30, 2022

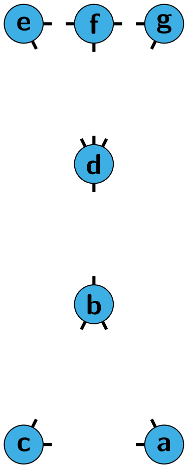
Notes:

- **Lecture L07:** Generating Functions 30.11.2022
- **Educational objective:** We introduce probability generating functions and explain how we can use them to study expected properties of random networks. As a first application, we use generating functions to analytically explain the friendship paradox that we discovered last week.
  - Introducing Generating Functions
  - Generating Functions in Networks
  - Explaining the friendship paradox

- **Exercise 05:** Generating functions and Molloy Reed model due 07.12.2022

Motivation

- **Molloy-Reed model**: ensemble of random networks with fixed
  - degree **sequence**
  - degree **distribution**
- generalizes ensemble of $G(n, p)$ random graphs (with Binomial degree distribution)
- **null model** for topology of networked systems (i.e. who is linked to *whom*)
- we introduced the **friendship paradox**, i.e. random neighbours of a randomly chosen node v having – on average – higher degree than v



$$S = (2, 3, 2, 4, 2, 3, 2)$$

Notes:

- In the last lecture, we introduced the **Molloy-Reed or configuration model**, which defines a statistical ensemble of microstates with a fixed degree sequence of degree distribution (which define the macrostate of the ensemble). To generate **random networks with given degree sequence**, we randomly connect pairs of "link stubs" generated from the desired degree sequence. We can generalise this model to generate **random networks with fixed degree distribution**. For this we first draw a degree sequence of length n from the degree distribution, check whether the sequence is graphical, and then apply the Molloy-Reed model.
- The Molloy-Reed model is widely used to study which aspects of a system's network topology (i.e., who is linked to *whom*) can be explained based on a system's degree sequence (i.e., who is linked to *how many*) or degree distribution (i.e., *how many* are linked to *how many*). Since the **topology** of the generated networks is random, the expected properties of microstates are solely determined by the degree sequence (or distribution).
- Today, we will show hpw we can use **generating functions** to make statements about expected properties of microstates in the resulting ensemble, which partly explains the popularity of the underlying model. In the last lecture we have also mentioned that we can consider the Molloy-Reed ensemble as a generalization of the simpler random graph models, if we fix the degree distribution to a Binomial (Poisson/Normal) distribution. If we learn how to analytically derive expected properties for this more general ensemble, we can also apply those skills to random networks.

Generating functions

- we introduce an **analytical framework to study expected properties of random networks** with given degree distributions

- we start with primer on **probability generating functions**

- we invest three weeks because …
  1. generating functions are a crucial tool in statistical network analysis
  2. you should know **how** popular statements in network science were derived
  3. you should **understand the limitations** of ensemble studies
  4. skipping derivations would leave you with an uneasy feeling

image credit: Stephen Edmonds, Wikimedia Commons, CC-BY-SA

Notes:

- The analytical framework that we will use to address this problem is based on **probability generating functions**. It is mostly used outside of the network context, where it has a number of applications in probability theory and combinatorics.
- It has first been used in the way presented in this course in the paper → MEJ Newman, SH Strogatz, DJ Watts, 2001 . In our coverage of the topic we will – for the most part – follow the presentation (and notation) used in this article. We will introduce this framework in detail, because …
    1. it is one of the most important methods in the (analytical) statistical analysis of complex networks,
    2. it explains why degree distributions are an important and frequently studied macrostate,
    3. it is needed to understand how widely known results from network science were derived, and
    4. it will help us to understand the limitations of the ensemble perspective.
- Before we can show how we can apply generating functions to networks, we first need to introduce a few basics.

Probability generating functions

probability generating function

Let $P(X = k)$ be a probability mass function of a discrete random variable X assuming values in $\mathbb{N}_0$. We call the absolutely convergent power series

$$G_0(x) := \sum_{k=0}^{\infty} P(k)x^k$$

the **(probability) generating function of $P(k)$**.

application of generating functions

- "A generating function is a **clothesline on which we hang up a sequence of numbers** for display. [...] Generating functions can give stunningly **quick derivations of various probabilistic aspects** of the problem that is represented by your unknown sequence" → Herbert S. Wilf, 1994
- we can use **generating functions of degree distributions** to compute component size distributions, average shortest path lengths, robustness, etc. of networks

Pierre-Simon Laplace
1749 – 1827

Herbert S. Wilf
1931 - 2012

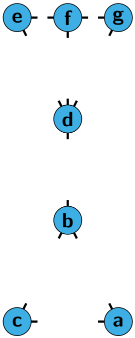
Notes:

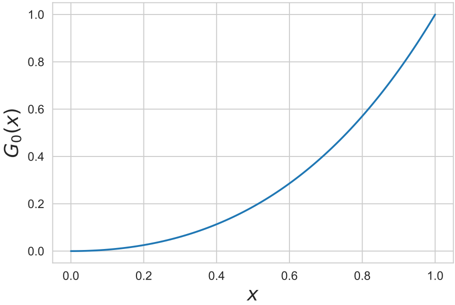
- Let us consider a discrete probability mass function $P(X = k)$ for a discrete random variable assuming values in $\mathbb{N}_0$. We can encode this (potentially infinite) sequence of probabilities into an absolutely convergent power series. We obtain a function $G_0(x)$ with a continuous parameter x in the domain $[0, 1]$. Note that $G_0(x)$ is absolutely convergent because $\sum_{k=0}^{\infty} |P(k)| = 1$.
- If P is a probability mass function we call G_0 a probability generating function, a term that can be traced back to a work by Laplace from 1779. We can also use generating functions to encode other sequences or infinite series of numbers, i.e. they are not limited to probability mass functions.
- The key idea to understand is that a generating function is nothing more than a fancy way to encode a sequence of numbers into a continuous function. If these numbers are probabilities, we can use them to make surprisingly simple statements about expected values, variances, etc. In the network context, these statements relate to statistical quantities of interest like, e.g., component size distribution, diameter, robustness, etc.
- For a general introduction to generating functions, I refer you to the textbook → HS Wilf, 1995 , for which the second edition is available as a free PDF at http://www.math.upenn.edu/~wilf/DownldGF.html.

Generating functions of degree distributions

- consider the function $G_0(x)$ generating $P(k) := P(X = k)$

$$G_0(x) := \sum_{k=0}^{\infty} P(k) \cdot x^k = P(0) + P(1)x + P(2)x^2 + P(3)x^3 + \cdots$$





$$P(2) = \frac{4}{7}, P(3) = \frac{2}{7}, P(4) = \frac{1}{7}$$

$$G_0(x) = \frac{4x^2}{7} + \frac{2x^3}{7} + \frac{x^4}{7}$$

Notes:

- We are interested networks, so let us consider an example of a function that generates the degree distribution of our toy network. The functional form is easy to understand based on the definition of G_0 and the resulting function is plotted below.
- We note that for $x = 1$ we obtain

$$G_0(1) = \sum_{k=0}^{\infty} P(k) \cdot 1 = 1$$

- while for $x = 0$ we have:

$$G_0(0) = \sum_{k=0}^{\infty} P(k) \cdot 0^k = P(0)$$

- In the analysis that we will do in the following lectures we often assume $G_0(0) = P(X = 0) = 0$, in which case the generating function assumes a minimum value of zero for $x = 0$. In the case of a network, we can ensure $P(X = 0) = 0$ by only considering nodes that have at least one link, i.e. we ignore nodes that have no links.

Generating probabilities?

- we say $G_0(x)$ **generates** $P(k)$ because

$$\left[\frac{1}{k!}\frac{d^k}{dx^k}G_0\right]_{x=0} = P(k)$$

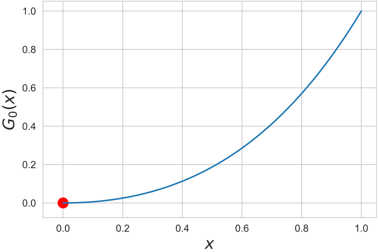
example network

$$G_0(x) = \frac{4x^2}{7} + \frac{2x^3}{7} + \frac{x^4}{7}$$

$$\left[\frac{1}{2!}\frac{\mathrm{d}^2}{\mathrm{d}x^2}G_0\right]_{x=0} = \frac{1}{2}\cdot\frac{8}{7} = P(2)$$

$$\left[\frac{1}{3!}\frac{\mathrm{d}^3}{\mathrm{d}x^3}G_0\right]_{x=0} = \frac{1}{6}\cdot\frac{12}{7} = P(3)$$

$$\left[\frac{1}{4!}\frac{\mathrm{d}^4}{\mathrm{d}x^4}G_0\right]_{x=0} = \frac{1}{24}\cdot\frac{24}{7} = P(4)$$



Notes:

- $G_0(x)$ is called **probability generating function function** because it generates the probability mass function $P(k)$. This is due to the equation above. Remember the following:
  - $k!$ denotes the factorial, i.e. $k! = k \cdot (k-1) \cdot (k-2) \cdot \ldots \cdot 1$ while $\frac{\mathrm{d}^k}{\mathrm{d}x^k} f$ denotes the k-th derivative of f by x
  - $\frac{\mathrm{d}^k}{\mathrm{d}x^k} f$ is the Leibniz notation of the k-th derivative, an alternative notation for the Lagrange notation $f^{(k)}$. We do not use Lagrange's notation to avoid confusion with the k-th power of a function f^k (which we will use later). However, for the sake of brevity we use Lagrange's notation f' for the first derivative $\frac{d}{dx} f$.
- We can validate the fact that the probability mass function is recovered by the k-th derivatives at $x = 0$ for our toy example network. Calculating the derivatives for a power series is particularly easy since for $f(x) = c \cdot x^k$, $f'(x) = \frac{d}{dx} f = k \cdot c \cdot x^{k-1}$.
- This has a remarkable implication: the **whole degree distribution $P(k)$ of the network is encoded in the slope of the (continuous) function G_0 at the point $x = 0$**.

Generating functions: first derivative

- **first derivative of $G_0(x)$ at $x = 1$** yields expected value

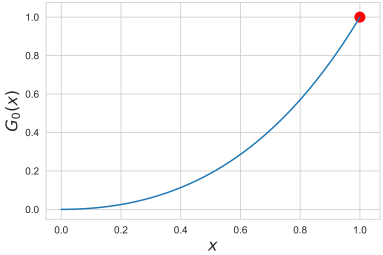
$$G_0'(1) = \left[\frac{\mathrm{d}}{\mathrm{d}x} G_0\right]_{x=1} = \left[\sum_{k=0}^{\infty} kP(k)x^{k-1}\right]_{x=1} = \sum_{k=0}^{\infty} kP(k) = \langle k \rangle$$

example network

$$G_0(x) = \frac{4x^2}{7} + \frac{2x^3}{7} + \frac{x^4}{7}$$

$$G_0'(x) = \frac{8x}{7} + \frac{6x^2}{7} + \frac{4x^3}{7}$$

$$G_0'(1) = \frac{18}{7} = \langle k \rangle \approx 2.57$$



Notes:

- Apart from this, probability generating functions have additional properties that make them interesting for statistical analysis. We first observe that the first derivative at the point $x = 1$ is the mean of the underlying distribution. This simply follows from the definition of the generating function and the definition of a derivative of the power series. By taking the first derivative of the generating function, we simply produce a factor k for $P(k)$ in the sum, while reducing the exponent by one. If we evaluate the first derivative at $x = 1$ we get the mean (i.e. the expected value) of the underlying probability mass function.
- For our toy example, it is easy to verify that the first derivative at $x = 1$ yields the the mean degree of 2.57 of our example network.
- This has the interesting implication that the **expected value of the probability mass function is encoded in the slope of the generating function at the point $x = 1$**.

Generating functions: higher derivatives

- **m-th derivative of $G_0(x)$ at $x = 1$** yields **m-th raw moment** of $P(k)$

$$\boxed{\left[\left(x\frac{\mathrm{d}}{\mathrm{d}x}\right)^m G_0\right]_{x=1} = \sum_{k=0}^{\infty} k^m P(k) = \langle k^m \rangle}$$

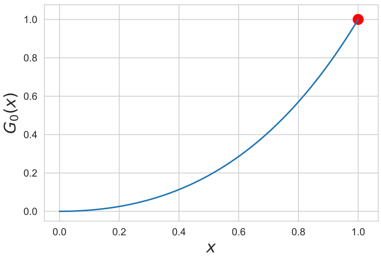
example network

$$G_0(x) = \frac{4x^2}{7} + \frac{2x^3}{7} + \frac{x^4}{7}$$

$$\left(x\frac{\mathrm{d}}{\mathrm{d}x}\right) G_0 = \frac{8x^2}{7} + \frac{6x^3}{7} + \frac{4x^4}{7}$$

$$\left(x\frac{\mathrm{d}}{\mathrm{d}x}\right)^2 G_0 = \frac{16x^2}{7} + \frac{18x^3}{7} + \frac{16x^4}{7}$$

$$\left[\left(x\frac{\mathrm{d}}{\mathrm{d}x}\right)^2 G_0\right]_{x=1} = \frac{50}{7} = \langle k^2 \rangle \approx 7.143$$



Notes:

- We can generalise this to the **m-th raw moment $\langle k^m \rangle$ of a probability mass function** (see L04 slide 4). The m^{th} raw moment is the m-th moment about zero, or in other words, the expected value of the m^{th} power of k (thus the notation). We can obtain it from the generating function by the formula above. Here, the operator $\left(x\frac{d}{dx}\right)^m$ means: take the derivative and multiply with x, do this m times.

- The variance is the second "central" moment, i.e. the moment over the mean of the distribution. In L04 we have seen that we can compute the variance from the second raw moment and the mean as follows:

$$Var(X) = E(X^2) - E(X)^2 = \langle k^2 \rangle - \langle k \rangle^2$$

$$\langle k^2 \rangle = \mathrm{Var}(X) + \langle k \rangle^2$$

- For $m = 1$, we have $\left[\left(x\frac{\mathrm{d}}{\mathrm{d}x}\right)^m G_0\right]_{x=1} = [x \cdot G_0']_{x=1} = G_0'(1)$, i.e. we confirm that the m-th raw moment is just the mean

- For $m = 2$ we have

$$\left[\frac{d}{dx} x G_0'(x)\right]_{x=1} = \left[\sum_{k=0}^{\infty} k^2 P(k) x^{k-1}\right]_{x=1} = \langle k^2 \rangle$$

- Again, we find something remarkable: **all raw moments of a probability mass function $P(k)$ are encoded in the slope of the generating function at the point $x = 1$**.

Powers of generating functions

- consider second power $[G_0]^2$ of generating function G_0, i.e.

$$[G_0(x)]^2 = \left[\sum_{k=0}^{\infty} P(k)x^k\right]^2 = [P(0) + P(1)x + P(2)x^2 + \ldots]^2$$

- we have

$$\begin{aligned}[G_0(x)]^2 &= P(0)^2 + x(P(0)P(1) + P(1)P(0)) \\ &\quad + x^2(P(0)P(2) + P(1)P(1) + P(2)P(0)) + \ldots \\ &= \sum_{i+j=0}^{\infty} P(i)P(j)x^{i+j}\end{aligned}$$

interpretation

m-th power $[G_0]^m$ generates pmf P of **sum of m independent realisations** X_i of random variable X, i.e.

$$P\left(\sum_{i=1}^{m} X_i\right)$$

Notes:

- What other operations could we apply to a generating function? Let us consider the m-th power, i.e. we **multiply a generating function m times with itself**. For $m = 2$ we observe that the factors associated with x^k give exactly the probabilities of those pairs of (independent) realizations that sum up to k. Example: the probability that two realizations of the random variable X exactly sum to two is $P(X = 0)P(X = 2) + P(X = 1)P(X = 1) + P(X = 2)P(X = 0)$ etc. We can rewrite this sum as given above, where the summation goes over all ordered pairs of two integer numbers $j, k \geq 0$ whose sum is $0, 1, \ldots$. Using the multinomial theorem we obtain a similar formula for general $m > 1$:

$$\left[\sum_{k=0}^{l} P(k)x^k\right]^m = \sum_{k_0+\ldots+k_l=m} \frac{m!}{k_0!\ldots k_l!} \left(P(k_0)x^{k_0}\right)^{k_0} \left(P(k_1)x^{k_1}\right)^{k_1} \ldots \left(P(k_l)x^{k_l}\right)^{k_l}$$

- In the equation above, the summation goes over all sequences $k_1, k_2, \ldots, k_l$ of l integer numbers which sum to m, i.e. we can rewrite as:

$$\sum_{k_1+k_2+\ldots+k_m=0}^{\infty} P(k_1)P(k_2)\ldots P(k_m)x^{k_1+k_2+\ldots+k_m}$$

- This leads us to the interpretation that the **m-th power of G_0 generates the probability mass function for the sum of m independent realizations** of the underlying random variable.

Composing generating functions

- consider **two generating functions** G_0 **and** G_1 where
  - G_0 generates $P(X = k)$
  - G_1 generates $Q(Y = k)$
- for **composition** $G_0 \circ G_1$ we get

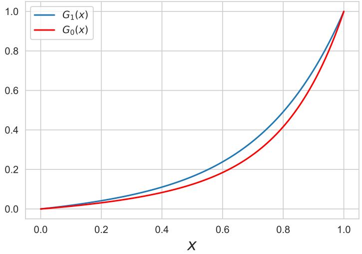
$$G_0(G_1(x)) = \sum_{k=0}^{\infty} P(k)[G_1(x)]^k$$

- $[G_1(x)]^k$ generates pmf of sum of k independent realizations of Y

interpretation

$G_0 \circ G_1$ generates pmf P of **sum of X independent realizations Y_i of Y**, i.e.

$$P\left(\sum_{i=1}^{X} Y_i\right)$$



example

- G_0 generates pmf P of face value X of fair dice with eight sides
- G_1 generates pmf Q of face value Y of fair dice with six sides
- $G_0 \circ G_1$ generates pmf of $\sum_{i=1}^{X} Y_i$ where Y_i are rolls of dice Y

→ see Exercise Sheet 05

Notes:

- Apart from taking derivatives and powers, we can also **compose functions** by using the output of one function as the input of another function. Consider an example with two random variables X and Y whose probability mass functions P and Q are generated by two generating functions G_0 and G_1. This works, because the domain of a probability generating function is the interval $[0, 1]$, and the function assumes values in the same interval.
- To make this concrete, assume that the two random variables are the output of two dice experiments with six and eight sides respectively. Consider the following combined random experiment consisting of two steps:

    1. roll the dice with eight sides and remember the result X (e.g. $X = 7$)
    2. roll the dice with six sides X times, and sum all of the face-values Y. I.e. if the outcome of the first dice is $X = 7$, sum all the face values Y obtained in seven rolls of the six-faced dice.

- Denote this sum by S, which is another random variable that can take values in the range $[1, \ldots, 48]$. The minimum value $S = 1$ implies that you rolled a one with the first dice, and another 1 with the second dice. The maximum value $S = 48$ implies that you rolled an eight with the first dice, and eight times a six with the second dice.
- Assuming that G_0 generates the probabilities of the outcomes X and that G_1 generates the probabilities of the outcomes Y, the probabilities for the different outcomes of S are generated by $G_0(G_1(x))$. The derivatives at $x = 0$ provide the probabilities for particular values of S, while derivatives at $x = 1$ allow us to calculate the mean of S and its higher moments.

Multiplication and division with x

- for **product of probability generation function $G_0(x)$ with x** we get

$$xG_0(x) := \sum_{k=0}^{\infty} xP(k)x^k = \sum_{k=0}^{\infty} P(k)x^{k+1} = P(0)x + P(1)x^2 + P(2)x^3 + \dots$$

interpretation

xG_0 generates $P(X = k-1) = P(X+1 = k)$

- for **division of probability generation function $G_0(x)$ with x** we get

$$\frac{1}{x}G_0(x) := \sum_{k=0}^{\infty} \frac{1}{x}P(k)x^k = \sum_{k=0}^{\infty} P(k)x^{k-1} = \frac{1}{x}P(0) + P(1) + P(2)x + \dots$$

interpretation

$\frac{1}{x}G_0$ generates $P(X = k+1) = P(X-1 = k)$

Notes:

- We close this introduction of generating functions by considering the functions obtained by multiplying or dividing a generating function $G_0(x)$ with its functional argument x. It is easy to calculate the resulting power series (see above).
- Note that these definitions require us to set $P(X=0)=0$ for the underlying probability mass function (which in a network we can ensure by ignoring zero-degree nodes): For $P(X=0)>0$, the function $\frac{1}{x}G_0(x)$ is not well-defined as the first term is $\frac{P(X)}{0}$. With $P(X)=0$ we can define $\frac{0}{0}=0$, which ensures that the resulting function is well-defined.
- What quantities are generated by these (new) generating functions?
- For xG_0 the k-th power of x is now associated with $P(k-1)$, i.e.

$$\left[\frac{1}{k!}\frac{d^k}{x^k}xG_0\right]_{x=0} = P(X=k-1) = P(X+1=k)$$

- For $\frac{1}{x}G_0$ the k-th power of x is now associated with the factor $P(k+1)$, i.e.

$$\left[\frac{1}{k!}\frac{d^k}{x^k}\frac{1}{x}G_0\right]_{x=0} = P(X=k+1) = P(X-1=k)$$

Application to statistical network ensembles

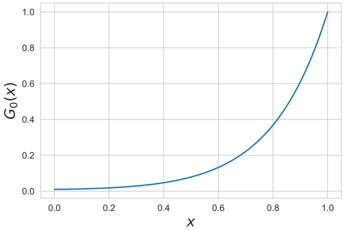
- we assume that G_0 **generates degree distribution** $P(X = k)$ of a random microstate

generative model perspective

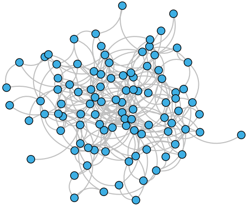
1. for given degree distribution generate random network using configuration model → L06
2. pick node v uniformly at random
3. for degree k of v pmf $P(k)$ is generated by G_0

- **generating function** encodes statistical ensemble of networks with fixed degree distribution $P(X = k)$

→ MEJ Newman, SH Strogatz, DJ Watts, 2001



generating function of Poisson distribution



random microstate from Molloy-Reed ensemble with Poisson distribution

Notes:

- We finally explain how exactly we can apply generating functions to statistical ensembles of networks. We assume that the (non-zero) part of a degree distribution $P(X = k)$ of random microstates is generated by a function G_0. We further assume that the topology of the network is random, i.e. the network is generated by the Molloy-Reed/configuration model introduced in L05.
- In other words: we assume that we know the probability that a randomly chosen node v has degree k, while all other features of the network are assumed to be random. This implies that **degrees of nodes are uncorrelated**, i.e. the independence assumption made in the calculation of some of the properties based on generating functions (cf. powers/composition of generating functions) holds.
- The figure above shows a random microstate with a Poisson distribution, which we generated using the Molloy-Reed model. The figure on the top right shows the corresponding generating function that generates a Poisson degree distribution. We thus use a continuous object (a function) to encode a random discrete object (a network or, more precisely, its degree distribution).
- We now show how can we use this perspective in practice, e.g. to study the friendship paradox observed in last week's lecture. For this, we will adopt the approach of using generating functions to statistical ensembles of networks with fixed degree distributions as presented in → MEJ Newman, SH Strogatz, DJ Watts, 2001

Practice session

- we plot probability generating functions for statistical ensembles with different degree distributions
- we compare the generating function for the degree distribution and the **excess degree distribution** in random networks

practice session

see notebook 07-01 and 07-02 in `gitlab` repository at

→ https://gitlab.informatik.uni-wuerzburg.de/ml4nets_notebooks/2022_wise_sna_notebooks

Notes:

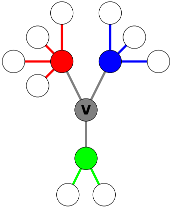
- In the first pracrice session, we take a more practical perspective on generating functions. We will calculate and plot generating functions for the degree distributions of randomly generated and synthetic networks.
- We further consider the so-called excess degree distribution, which we already considered in exercise session 4. We will compare the generating function of the degree distribution and the excess degree distribution, which we will analytically study in the following slides.

Excess degree distribution

- distribution of the **degrees of neighbours** w without (v, w)

$$G_1(x) := \frac{\sum_{k=0}^{\infty} \frac{k}{\langle k \rangle} P(k) x^k}{x} = \frac{\sum_{k=0}^{\infty} k P(k) x^{k-1}}{\langle k \rangle} = \frac{G_0'(x)}{G_0'(1)}$$

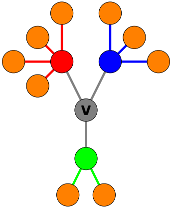
(often called **excess degree distribution**)



$\{4, 3, 2\}$

- distribution of **number of second-nearest neighbours** of v

$$\sum_{k=0}^{\infty} P(k) [G_1(x)]^k = G_0(G_1(x))$$



$4 + 3 + 2 = 9$

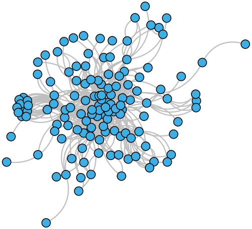
Notes:

- Consider a node v chosen uniformly at random in a Molloy-Reed microstates. What other quantities can we calculate? For the distribution of the **degrees of neighbours w of a randomly chosen v**, the following holds:
  - The probability that a randomly chosen node w has degree k is $P(k)$. However, we must additionally account for the fact that w is also a neighbour of v (i.e. w was **not** chosen uniformly at random).
  - A node with degree k has k chances to be randomly chosen as neighbour of v, so the probability that w has degree k is proportional to $kP(k)$
  - We need to normalize this to obtain a probability, i.e. we divide each probability $P(k)$ by $\sum_k kP(k) = \langle k \rangle$.
  - In addition, we must discount for link (v, w), which decrements the resulting degree by one. We can achieve by dividing the generating function by x.
- We obtain a **new generating function $G_1(x)$**, which generates the probability that a random neighbour of a randomly chosen node v has degree k (without (v, w)). This is often called the **excess degree distribution** (cf. EX 04). Using this function we can calculate the **distribution of the number of nodes at distance two** to a randomly chosen node v as follows:
  - We sum all degrees of neighbors w of v (without considering (v, w)).
  - We thus sum k realizations of neighbour degrees (which are generated by G_1) where k is generated by G_0. Hence the distribution that we are looking for is generated by the composition of G_0 and G_1!
- This holds if there is **zero clustering**, i.e. if we can ignore the case that a neighbour of w is also a neighbour of v (which would lead to a closed triad).

The friendship paradox

- consider a random network generated using the Molloy-Reed model
- **friendship paradox** states that random neighbor w of randomly chosen node v has – on average – more neighbors than v
- for **mean degree** $\langle k \rangle$ and **mean neighbour degree** $\langle k_n \rangle$ this translates to

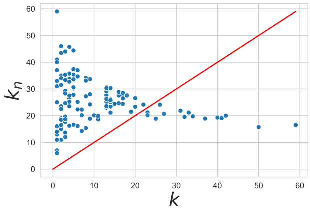
$$\langle k_n \rangle > \langle k \rangle$$



example: Lord of the Rings character co-occurrence

for random microstate from Molloy-Reed ensemble with degree sequence of Lord of the Rings character network we find

$$\langle k_n \rangle = 23.34 > 10.4 = \langle k \rangle$$



Notes:

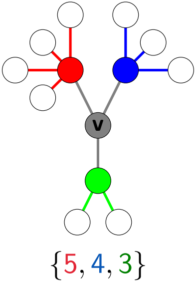
- Last week, we motivated the friendship paradox, which describes the – seemingly unintuitive – finding that your friends in a social network have – on average – more friends than you. We have studied this in the empirical example of the Lord of the Rings character co-occurrence network. For this we have compared the empirical mean degree to the empirical mean neighbor degree. If $\langle k_n \rangle > \langle k \rangle$, then the friendship paradox holds on average.
- The figure on the right shows the degrees and mean neighbor degrees of all characters in the Lord of the Rings books. We see that the average degree (average over the x-axis) is 10.4, while the average of the mean neighbor degree (average over the y-axis) is 23.34, i.e. the friendship paradox holds.
- We can use generating function G_1 (which encodes the distribution of neighbor degrees of without (v, w)) to analytically explain this phenomenon. We can further clarify under which conditions it occurs. This analysis utilizes the fact that we can calculate generating function G_1 based on the function G_0 that generates the node degree distribution.

Explanation with generating functions

- **neighbor degree distribution** of random node v is generated by

$$x \cdot G_1(x)$$

i.e. **excess degree distribution plus one**



- **expected neighbor degree** $\langle k_n \rangle$ is given as

$$\langle k_n \rangle = \left[\frac{\mathrm{d}}{\mathrm{d}x} x \cdot G_1(x) \right]_{x=1} = \left[1 \cdot G_1(x) + x \cdot G_1'(x) \right]_{x=1}$$

$$\Rightarrow \langle \mathbf{k_n} \rangle = \mathbf{1 + G_1'(1)}$$

Notes:

- Using generating functions, it is easy to analytically explain the friendship paradox. We simply need to compute the expected neighbor degree and compare it to the expected degree (which we can calculate based on G_0). For this we need a generating function that generates the degree distribution of a random neighbor w of a random node v (now including the degree that is due to link (v, w)). We know that G_1 generates the distribution of the degrees of the neighbors of a randomly chosen node v, discounting for (v, w). So we simply add one to the random variable, which we can do by multiplying function G_1 with x.
- Hence, $x \cdot G_1(x)$ generates the distribution of degrees of the neighbors w of a randomly chosen node v. We now compute the expected neighbor degree by calculating the value of the first derivative of $xG_1(x)$ for $x = 1$. For this we apply the product rule for derivatives, i.e. $(f \cdot g)' = f' \cdot g + f \cdot g'$.
- For any probability generating function, we have $G_1(1) = 1$ and thus $\langle k_n \rangle = 1 \cdot 1 + 1 \cdot G_1'(1)$.
- We thus find that the mean neighbor degree in a random network is just the expected value of the distribution generated by G_1 plus one. This is hardly surprising, since G_1 generates the degrees of w minus the one link (v, w), which we have added back by multiplying G_1 with x.

The expected neighbour degree

- to obtain $\langle k_n \rangle$ we first calculate $G_1'(x)$ as

$$G_1'(x) = \frac{d}{dx} \frac{G_0'(x)}{G_0'(1)} = \frac{d}{dx} \frac{\sum_{k=0}^{\infty} kP(k)x^{k-1}}{\sum_{k=0}^{\infty} kP(k)} = \frac{\sum_{k=0}^{\infty} k(k-1)P(k)x^{k-2}}{\sum_{k=0}^{\infty} kP(k)}$$

- for $x = 1$ we obtain

$$G_1'(1) = \frac{\sum_{k=0}^{\infty} k(k-1)P(k)}{\sum_{k=0}^{\infty} kP(k)} = \frac{\sum_{k=0}^{\infty} k^2 P(k)}{\sum_{k=0}^{\infty} kP(k)} - 1 = \frac{\langle k^2 \rangle}{\langle k \rangle} - 1$$

- for the **expected neighbour degree** $\langle k_n \rangle$ of randomly chosen node v we get

$$\langle k_n \rangle = 1 + G_1'(1) = \frac{\langle k^2 \rangle}{\langle k \rangle}$$

Notes:

- To calculate the **expected neighbor degree**, we must calculate $G_1'(1)$. For this we first replace $G_1(x)$ by its definition. We then replace both $G_0'(x)$ and $G_0'(1)$ by the corresponding power series and take the derivative by x
- We then evaluate the resulting function at the point $x = 1$ and obtain an expression for $G_1'(1)$ that consists of the ratio of the first two raw moments of the degree distribution minus one. We finally substitute $G_1'(1)$ in our expression $\langle k_n \rangle = 1 + G_1'(1)$ for the expected neighbor degree.
- This yields the interesting finding that the expected neighbor degree in a random microstate with arbitrary degree distribution is the **ratio of the first two raw moments of the degree distribution**. This simple formula yields important insights: It is not only the mean of the degree distribution that determines how connected our neighbors are, but also its heterogeneity. This expression has important implications for the expected properties of networks, e.g. for the diameter, the connectedness or the robustness.

The friendship paradox: role of variance

- variance $\mathrm{Var}(X)$ can be calculated as → "MOSSOM" mnemonic, L05, slide 6

$$\mathrm{Var}(X) = \langle k^2 \rangle - \langle k \rangle^2 \geq 0$$

- we can have two cases:

1. $\mathrm{Var}(X) = 0 \Rightarrow \langle \mathbf{k^2} \rangle = \langle \mathbf{k} \rangle^2$

$$\langle k_n \rangle = \frac{\langle k^2 \rangle}{\langle k \rangle} = \frac{\langle k \rangle^2}{\langle k \rangle} = \langle k \rangle$$

2. $\mathrm{Var}(X) > 0 \Rightarrow \langle \mathbf{k^2} \rangle > \langle \mathbf{k} \rangle^2$

$$\langle k_n \rangle = \frac{\langle k^2 \rangle}{\langle k \rangle} > \frac{\langle k \rangle^2}{\langle k \rangle} = \langle k \rangle$$

- friendship paradox holds in random networks with **any degree distribution with non-zero variance**

Notes:

- In lecture L05, we mentioned that we can calculate the **variance** of a distribution based on the first and second raw moment of a distribution. Think about the MOSSOM mnemonic: variance = Mean Of Square - Square Of Mean.
- Since the variance is necessarily non-negative, we can distinguish between two cases:
  1. $Var(X) = 0$: in this case, the second raw moment is necessarily equal to the square of the first raw moment. Substituting $\langle k^2 \rangle$ in the expression for the mean neighbor degree we find that in this case, the expected degree and the expected neighbor degree are actually the same. In this case, there is no friendship paradox!
  2. $Var(X) > 0$: in this case, the second raw moment is necessarily larger than the squared first raw moment. In this case the mean neighbor degree is larger than the mean degree.
- This simple analysis proves that a non-zero variance in the degree distribution is sufficient to explain the friendship paradox in random microstates of the Molloy-Reed model. Hence, **we expect the friendship paradox to hold whenever we have positive variance in the degree distribution**.
- Zero variance in the degree distribution actually implies that all degrees are the same. Such networks are called k-regular random networks (i.e. we have degree sequence $\{k, k, \ldots, k\}$). In $k - regular$ networks, the friendship paradox trivially does not hold, while it holds in all other networks (with varying “strength” that can be calculated from the formula above).

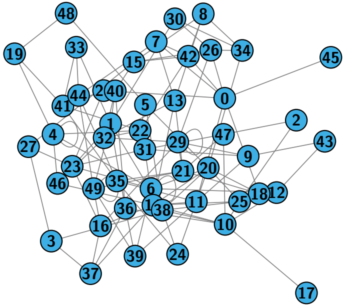
Friendship paradox in Erdős-Rényi networks

- friendship paradox can be interpreted as **sampling bias** occurring when sampling nodes by randomly following a link

- consider a sparse, random **Erdös-Rényi network**

- for randomly chosen node v compare two strategies to discover nodes
  1. collect d_v neighbors of v
  2. for random neighbor w of v collect $d_w - 1$ other neighbors of w

- how many **newly discovered nodes** do we expect?



Notes:

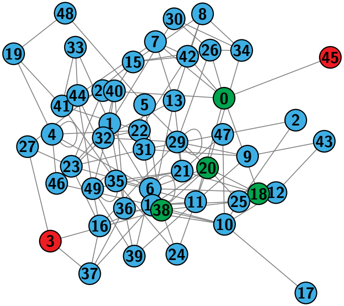
- As a related application, we can consider possible biases introduced by sampling strategies in networks.
- For this, consider a sparse Erdös-Rényi networks, i.e. a network randomly generated using the $G(n, p)$ or $G(n, m)$ model. Since the network is sparse we know that the degree distribution converges to a Poisson distribution in the limit of large n.
- In such a network, does it matter whether we (i) pick a node uniformly at random, or (ii) follow a random edge from a randomly chosen node?
- From our results on the friendship paradox, we already know that the expected degree of the random neighbor of a randomly chosen node is larger than for the node itself. This implies that these two sampling strategies are indeed not equivalent.
- A discussion of the bias introduced by sampling links based on so-called `traceroute` paths which makes use of generating functions can be found in

  $\longrightarrow$ D Achlioptas, A Clauset, D Kempe, C Moore, 2009
- If we consider the number of new nodes that we discover in the two sampling strategies starting from a given node v, we find that we expect to discover $\langle k_n \rangle - 1$ new nodes if we pick a random neighbor w of v while we expect to discover $\langle k \rangle$ nodes if we pick the random node v.
- This question might remind you about the analysis of the diameter that we have done in lecture L05.

Reminder: diameter of random graphs

- in lecture L05, we calculated the **expected diameter** of random networks as follows …

- consider randomly chosen node v
  - v has expected number of $\langle k \rangle$ neighbors
  - each neighbour of v has **expected number of $\langle k_n \rangle - 1$ new neighbors**
  - how many of those are neighbors of v?
  - for sufficiently **large sparse networks**: none since $C \to 0 \; (n \to \infty)$
  - we expect $\langle k \rangle^l$ nodes at distance l of node v



- from $D \Leftrightarrow \langle k \rangle^D \approx n$ we concluded

$$D \approx \frac{\log n}{\log \langle k \rangle}$$

- since $\langle k_n \rangle > \langle k \rangle$, **why is this not wrong?**

Notes:

- In lecture L05, we have argued that each of the neighbors of a random node v has again $\langle k \rangle$ neighbors.
- Doesn't the friendship paradox invalidate this claim? In the exercises we have empirically validated the resulting scaling relation and we have found that it holds very accurately. How can we explain this?
- Let us critically revisit two statements, which we used in the calculation in lecture 04:
    1. a (randomly chosen) node v has $\langle k \rangle$ neighbors
    2. each of these neighbors has again $\langle k \rangle$ neighbors
- In the light of what we have seen for the friendship paradox, the second statement sounds odd, because $\langle k_n \rangle > \langle k \rangle$ (because the Poisson distribution has positive variance).
- However, by assuming that each of the neighbors of v has an expected number of $\langle k \rangle$ neighbors, we have ignored the fact that one of those $\langle k \rangle$ neighbors is v. We (purposefully) ignored this in L05. We can now explain why the simple scaling relation that we have found is still valid.

Solution with generating functions

- $G_1(x)$ generates distribution of **degrees of neighbours** w of v without considering (v, w)

- for Poisson distribution we have $\rightarrow$ Exercise

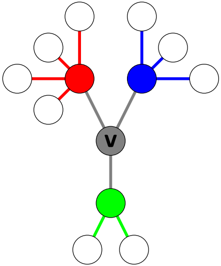
$$\left[\frac{1}{k!}\frac{\mathrm{d}^k}{\mathrm{d}x^k}G_0\right]_{x=0} = P(k) = \frac{\langle k\rangle^k}{k!}e^{-\langle k\rangle}$$

and thus $G_0(x) = e^{\langle k\rangle(x-1)}$

- for this **special case** we find

$$G_1(x) = \frac{\langle k\rangle e^{\langle k\rangle(x-1)}}{\langle k\rangle} = G_0(x)$$

- we generally have $G_0 \neq G_1$ for other degree distributions



$$G_1(x) := \frac{G_0'(x)}{G_0'(1)}$$

observation

for sparse random network with Poisson distribution:

$\langle k_n\rangle = G_1'(1) + 1 = G_0'(1) + 1 = \langle k\rangle + 1$

for **number of newly discovered neighbors** in Erdös-Rényi network, it does not matter whether we pick a random node or follow a random link

Notes:

- If we ignore the link back to v, the expected number of "new" nodes that we discover in each step is exactly $\langle k \rangle$ (as stated in L05). We can explain this analytically as follows: for sufficiently large sparse random networks we can approximate $P(k)$ with the Poisson distribution (see L05).

- In the exercise sheet we have seen that the generating function of the Poisson distribution is $G_0(x) = e^{\lambda(x-1)}$. This can easily be proven using the following two facts:

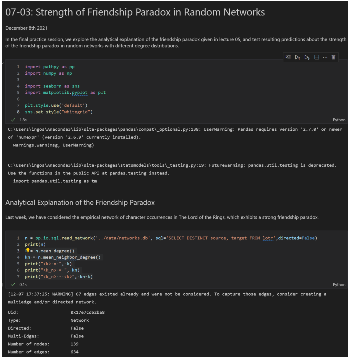
  1. the first derivative of $e^{c(x-1)}$ is $ce^{c(x-1)}$
  2. the k-th derivative over the factorial of k at $x = 0$ generates $P(k)$

- With $G_0'(x) = \langle k \rangle e^{\langle k \rangle (x-1)}$ and $G_0'(1) = \langle k \rangle$ we find that $G_1(x) = G_0(x)$. Since $G_1(x)$ generates the degree of neighbors j including the one link to node i, this implies $\langle k_n \rangle = \langle k \rangle + 1$.

- In other words: **For the special case of sparse random networks with a Poisson distribution G_0 and G_1 are actually identical**. This translates to the fact that the expected neighbor degree is exactly the expected degree plus one. Hence, the generating function G_0 contains **all** information about the (random) topology. This is the reason why random networks are particularly easy to study and it also explains why they are a special case in the larger set of ensembles of networks with arbitrary degree distributions. Do not forget that this special property follows from the Poisson degree distribution of sparse (and sufficiently large) Erdös-Rényi networks. It does **not** hold in general, i.e. we generally have $G_0 \neq G_1$.

Practice session

- we study the strength of the **friendship paradox** in random networks
- we validate our **analytical predictions** for different degree distributions



practice session

see notebook 07-03 in `gitlab` repository at

→ https://gitlab.informatik.uni-wuerzburg.de/ml4nets_notebooks/2022_wise_sna_notebooks

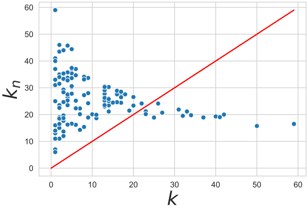
Notes:

- In the last practice session, we explore the friendship paradox and test our analytical predictions in simulations of random networks.

In summary

- we introduced **generating functions**, which we can use to encode **degree distributions of networks**

- we applied **generating functions** to analytically explain the friendship paradox



friendship paradox

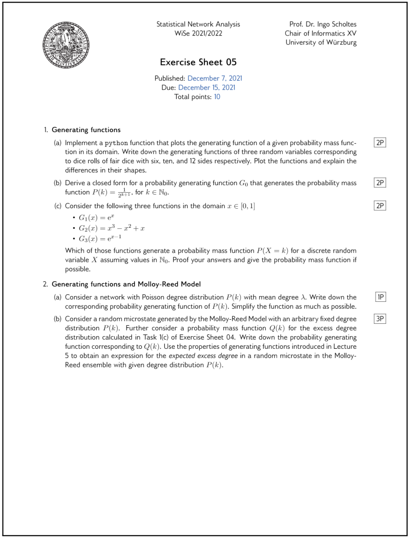
- **not a sociological finding**
- **sampling bias** that depends on variance of degree distribution
- intuition that neighbor w of random node v is similar to v is wrong

Notes:

- In summary, we introduced the framework of generating functions, which we can use to encode degree distributions of networks. We can use those functions to obtain surprisingly simple derivations for the expected properties of random networks with a given degree distribution.
- As a first application, we used generating functions to analytically explain the friendship paradox based on the variance of the degree distribution, highlighting that the phenomenon is a sampling bias rather than a sociological finding. It rectifies our wrong intuition that a random neighbor w of a random node v should be similar to v.
- A possible application of the friendship paradox are vaccination strategies. For a limited amount of vaccination doses, how can we maximize the number of saved lives (under the unrealistic assumption that there is no difference of vaccinating younger vs. older/healthy vs. non-healthy individuals). A simple strategy would be to randomly chose a fraction of the population, but the friendship paradox can provide us with a better strategy: Instead of sampling random individuals, we ask them to name a random friend and then offer this friend to receive a vaccination. This has a larger impact than the vaccination of a random sample of the population.
- Another application is related to the recent Cambridge Analytica scandal, which harvested data by making 270,000 users install a Facebook app. A flaw in Facebook's data access policy enabled this app to harvest data from the friends of those users. (Unrealistically) assuming zero clustering, we could use generating functions to calculate the expected number of users affected by the scandal.

Exercise sheet 05

- **fifth exercise sheet** available on WueCampus
  - explore properties of generating functions
  - apply generating functions in Molloy-Reed networks
- solutions are due **December 7th** (via WueCampus)
- present your solution to earn bonus points


Statistical Network Analysis
WiSe 2021/2022

Prof. Dr. Ingo Scholtes
Chair of Informatics XV
University of Würzburg

Exercise Sheet 05

Published: December 7, 2021
Due: December 15, 2021
Total points: 10

1. Generating functions
   (a) Implement a python function that plots the generating function of a given probability mass function in its domain. Write down the generating functions of three random variables corresponding to dice rolls of fair dice with six, ten, and 12 sides respectively. Plot the functions and explain the differences in their shapes. 2P
   (b) Derive a closed form for a probability generating function G_0 that generates the probability mass function $P(k) = \frac{1}{2^{k+1}}$, for $k \in \mathbb{N}_0$. 2P
   (c) Consider the following three functions in the domain $x \in [0, 1]$ 2P
   - $G_1(x) = e^x$
   - $G_2(x) = x^3 - x^2 + x$
   - $G_3(x) = e^{x-1}$

   Which of those functions generate a probability mass function $P(X = k)$ for a discrete random variable X assuming values in $\mathbb{N}_0$. Proof your answers and give the probability mass function if possible.
2. Generating functions and Molloy-Reed Model
   (a) Consider a network with Poisson degree distribution $P(k)$ with mean degree λ. Write down the corresponding probability generating function of $P(k)$. Simplify the function as much as possible. 1P
   (b) Consider a random microstate generated by the Molloy-Reed Model with an arbitrary fixed degree distribution $P(k)$. Further consider a probability mass function $Q(k)$ for the excess degree distribution calculated in Task 1(c) of Exercise Sheet 04. Write down the probability generating function corresponding to $Q(k)$. Use the properties of generating functions introduced in Lecture 5 to obtain an expression for the *expected excess degree* in a random microstate in the Molloy-Reed ensemble with given degree distribution $P(k)$. 3P

Notes:

Self-study questions

1. Write down the definition of a probability generating function for a random variable X assuming values from one to six with a uniform distribution.
2. How does the shape of a function generating a uniform distribution $P(X) = \frac{1}{n}$ change as we increase n.
3. What is the probability mass function generated by the m-th power of a function $G_0(x)$?
4. How can we calculate the variance of a distribution based on probability generating functions?
5. Given an example for a probability mass function generated by the composition $G_1(G_0(x))$ of two generating functions G_0 and G_1.
6. What is special about the generating functions G_0 and G_1 in sparse Erdös-Rényi networks
7. Use generating functions to explain under which conditions the friendship paradox holds.
8. In which case can we calculate the number of nodes at distance l of a randomly chosen node i as $\langle k \rangle^l$?
9. Explain why the distribution of neighbour degrees for a randomly chosen node is generated by $xG_1(x)$?

Notes:

References

reading list

- HS Wilf: **generatingfunctionology**, Academic Press, 1994
- DE Knuth: **The Art of Computer Programming: Vol 1: Fundamental Algorithms**, Addison Wesley, 1997 → Chapter 1.2.9
- MEJ Newman, SH Strogatz, DJ Watts: **Random graphs with arbitrary degree distributions and their applications**, Physical Review E, 2001
- SF Feld: **Why Your Friends Have More Friends Than You Do**, American Journal of Sociology 96, 1991
- NA Christakis, JH Fowler: **Social Network Sensors for Early Detection of Contagious Outbreaks**, PLOS ONE. 5, 2010

generatingfunctionology

Herbert S. Wilf
Department of Mathematics
University of Pennsylvania
Philadelphia, Pennsylvania

Notes: