# Network Project

March 9, 2017

**Abstract**

# 1 Implementation of the BA Model

## 1.1 The Initial Conditions

The BA model is a randomly generated model, which usees a mdethod called preferential attachement to favour which nodes to connect to. This means that nodes with a high degree are more likely to be attached to be new nodes. The algorithm I used works as follows: 1. Set of an initial network a time $\mathcal{G}_\prime$.

2.Increment time t → t+1

3.Add one new vertex. 4. Add m edges as follows.. ....
...
..
There are a few points of ambiguity in this model. The first of which is with respect to $\mathcal{G}_0$. There is no explicit guidance on how to choose $\mathcal{G}_0$, however the choice of starting graph does have an affect. When deriving a solving the master equation for the system, we will use the approximation that $E(t) = mN(t) forlarget$. However we can make this approximation exact by choosing an $\mathcal{G}_0$ such that $E(0) = mN(0)$.
In finding this, one assumption I would like to make is that ever node in $\mathcal{G}_\prime$ has the same degree. This make an easily programmably starting graph.This implies that $deg(n) = mforn \in \mathcal{G}_\prime$
There are many graphs with this property, however I would like to minimise the number of nodes in my starting graph(So our starting graph does not change our statistic) which implies we want a complete graph. THe algebra is as follow:
In a complete graph $E = \sum_{n=1}^{N} n - 1 = \frac{N(N-1)}{2}$
And so $E(0) = mN(0) \Rightarrow \frac{N(0)(N(0)-1)}{2} = mN(0)$

$\Rightarrow N(0)^2 - (2m - 1)N = 0$

$\Rightarrow N = 0(trivial) andN = 2m + 1$

Therefore choosing $\mathcal{G}_0$ to be a complete graph with $2m+1$ nodes is sufficient for the condition $E(0) = mN(0)$. Figure 1.1 shows the initial networks.

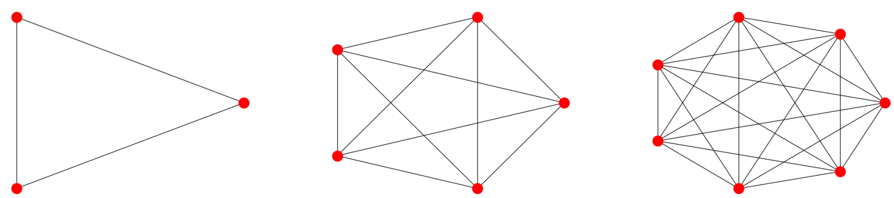

Figure 1: $\mathcal{G}_0$ *for m=1,2,3 respectively.*

## 1.2 Double Edges

Another point of ambiguity is with regards to multiple egdes. In the model, we have preferential attachement, which implies as we attach more edges to a node, it will be preferred even more when adding the node edge randomly. This "Rich get richer" attitude means that we are likely to get double edges when $m > 1$. For instance, if a new node $k$ is added and attached to node $n < k$, then the probability of that happening again rises, implying we are more likely to see a double edge. This is especially true for small networks. Figure 1 shows a graph of 10 without addressing this issue and one where we do. This phenomena does

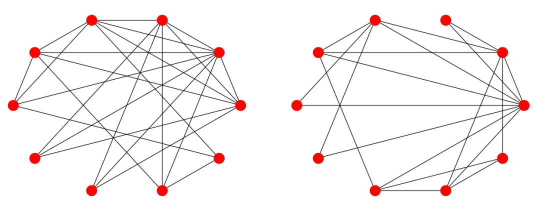

Figure 2: Left: *Example graph of 10 nodes where we allows double edges(m=3). Not that there are nodes with degree less than m.*
Right: *Exampleof graph of 10 nodes. Note that all nodes have degree $> m$. Note that in both cases, I have not used $\mathcal{G}_0$, and instead have used a small initial graph to emphasise the difference in the cases.*

not make sense in the circumstances for which this model is implemented, such as modeling the relationships between websites. Therefore I have decided to use the latter case. Also for large systems, theoretically there is no difference, since the probability of a node being chosen twice $\to 0$.

### 1.3 Udpating Probabilities

### 1.4 Testing

## 2 Theoretical Derivation of Degree

There are a few ways of approximated the degree distribution $p(k)$, all three of which use the master equation:

$$n(k,t+1) = n(k,t) + m\Pi(k-1,t)n(k-1,t) - m\Pi(k,t)n(k,t) + \delta_{k,m} \qquad (1)$$

Where $\Pi(k,t)$ is the probaility of an edge being attached to a node of degree $k$. Since we are taking $\Pi(k,t) \propto k$, and that the probabilities are normalised, way get that:

$$\Pi(k,t) = \frac{k}{\sum_{k=1}^{\infty} kn(k,t)} \qquad (2)$$

Where $kn(k,t)$ is the number of degrees of the nodes of degree k. Also, each edge is reponsible for 2 degrees, and so:

$$\Pi(k,t) = \frac{k}{2E(t)} \qquad (3)$$

I have already discussed that $E(t) = mN(t)$ using the initial conditions chosen, and so $\Rightarrow \Pi(k,t) = \frac{k}{2mN(t)}$. Applying this to (1) the master equation becomes:

$$n(k,t+1) = n(k,t) + \frac{(k-1)n(k-1,t)}{2N(t)} - \frac{kn(k,t)}{2N(t)} + \delta_{k,m} \qquad (4)$$

Now we define the probability of choosing a degree randomly with degree $k$ at time $t$:

$$p(k,t) = \frac{n(k,t)}{N(t)} \qquad (5)$$

So the master equation:

$$N(t+1)p(k,t+1) - N(t)p(k,t) = -\frac{k}{2}p(k-1,t) - \frac{k}{2}p(k,t) + \delta_k, m \qquad (6)$$

NOT SURE HERE

In order to go further, we assume that $p(k)$ has nice ergodic properties. This means that $p_\infty = lim_{t\to\infty}p(k,t)$

, i.e. the limit converges. Applying this to (6) the final form of our master equation becomes:

$$p_\infty(k) = -\frac{1}{2}((k-1)p_\infty(k-1) - kp_\infty(k)) + \delta_{k,m} \qquad (7)$$

## 2.1 Continuous Approximation

Equation (7) can be used to find the degree distribution of the model. An approximation of this distribution can be found using a limiting case, e.i. instead of have descrete degrees, we look at the continuous case $k+1 \to k + \Delta k$. (7) becomes:

$$p(k) \approx \lim_{\Delta k \to 0} \frac{-\frac{1}{2}((k-\Delta k)p_\infty(k-\Delta k) - kp_\infty(k)) + \delta_{k,m}}{\Delta k} \tag{8}$$

$$\Rightarrow p(k) \approx \frac{\partial k p_\infty(k)}{\partial k} \tag{9}$$

By inspection (Looking for a solution of the type $k^{-\gamma}$), we find that $p(k) \propto k^{-3}$ is a solution. This solution is very approximal. However once case we would expect to see such a distribution is for $m \to \infty$. As $m$ grows large, the difference between $k-1$ and $k$ grows small proportional to $k$, and so the limiting case becomes a reality.

## 2.2 Difference Derivation

It is poosible however to derive a solution from the difference equation. First we look at $k > m$ and rearrange (7):

$$\frac{p_\infty(k)}{p_\infty(k-1)} = -\frac{k-1}{2(k+1)} \tag{10}$$

This may no look particularly helpful, however there is an identity of the Gamma function. The equation:

$$\frac{f(z)}{f(z-1)} = \frac{z+a}{z+b} \tag{11}$$

Has the solution

$$f(z) = A\frac{\Gamma(z+1+a)}{\Gamma(z+1+b)} \tag{12}$$

Therefore our difference equation has solution

$$p_\infty(k) = A\frac{\Gamma(k)}{\Gamma(k+2)} \tag{13}$$

Using the identity $\Gamma(n) = (n-1)!$ for $n \in N_0$, the solution becomes:

$$p_\infty(k) = \frac{A}{k(k+1)(k+2)} \tag{14}$$

The constant $A$ can be found by looking at the boundary case, $k = m$, (7) becomes

$$p_\infty(m) = -\frac{m}{2}p_\infty(m) + 1 \tag{15}$$

$$\Rightarrow p_\infty(m) = \frac{1}{m+2} \tag{16}$$

This boundary conditon implies that

$$A = 2m(m+1) \tag{17}$$

Thus we derive the solution to the difference equation as:

$$p_{\infty}(k) = 2m(m+1)/k(k+1)(k+2) \tag{18}$$

I expect this distribution to be more accurate than that procured by the continuous approximation, as I have made less assumptions and approximations whilst deriving it.

# 3 Comparison with Real Data

Now I wish to compare these theoretical plots with the actual data captured by my model.
I shall run my programme for m=1,2,3 and for graphs of 10,000 nodes. I believe this is large enough to allow the ergodic properties of the probabilites, e.i. $p_{infty}(k)$ to arise.
A key characteristic of the model is that as one increases the number of nodes in the graph ($N$), the maximum dregee $k_1$ observed also increase, which means no matter big the graph, the statistics towards the larger degrees will always be sparse. To combat this I ran the same experiment 100 times in order to build up a enough observatinos for large degree $k$, improving our statistic. Figure 3.1 shows the outcome. Visually, one can see from that for small values of $k$, the probability fits our theoretical distribution perfectly. This is because there are a lot more nodes with degree small $k$, and so a lot more data is available, thus the distirbution is prominent. However, for large $k$ we have fewer and fewar nodes per degree, as predicted. This creates the 'fat tail' affect present on all three figures.
Notice as well that as m increase, our theoretical, and practical data moves closer to the line $p(k) = k^{-3}$. This verfifies the effect I would expect to see for large m.

## 3.1 Statistical Approach

I wish to analyse the model statistically. However, the 'fat tail' in the data will surely dominated any statistical test we wish to run. Thus a way of mimising this 'fat tail' affect, while keeping the necessary characteristics of the probability distribution is necessary. How I approached this we by log binning.
When creating anaylsing probability distributions from sampples, data is put into bins, and the frequency recorded. In most cases we have a bin length $b_{n+1} - b_n =: \Delta$, which is constant. However in a log bin proccess, the bins have a relation $\frac{b_{n+1}}{b_n} = \Delta$. This means that the bins increase logarithmically as the data grows larger(Hence the name).
This means for small $k$, where the data are plentiful, the bins are small to

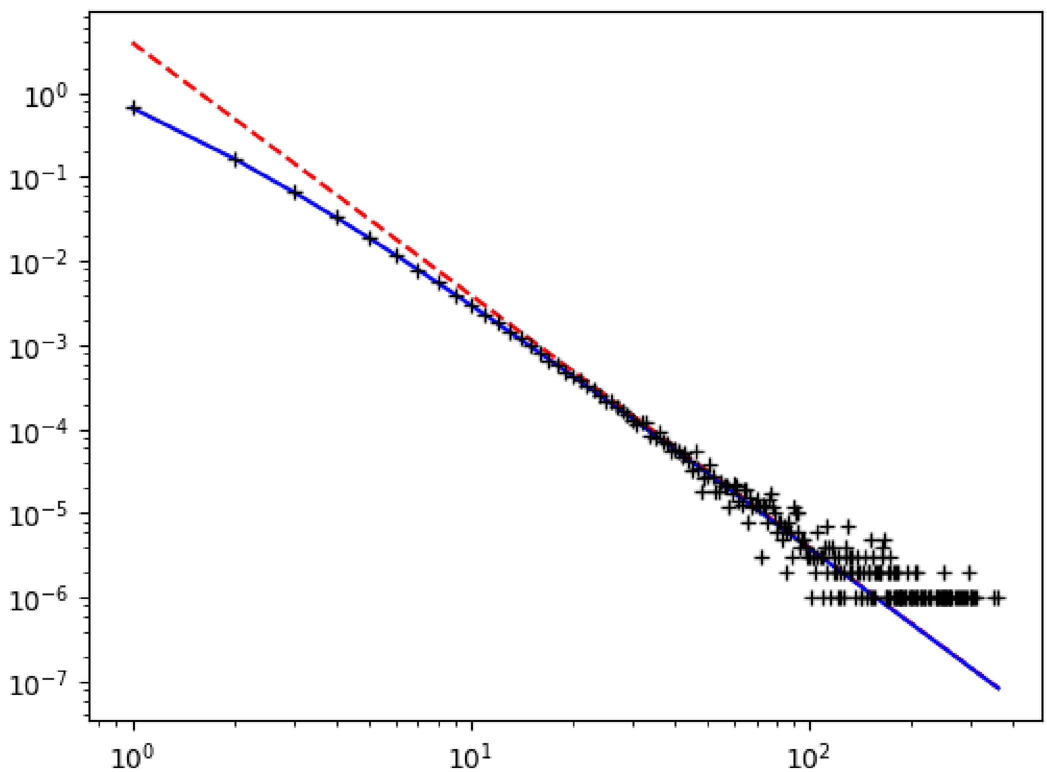


Figure 3:

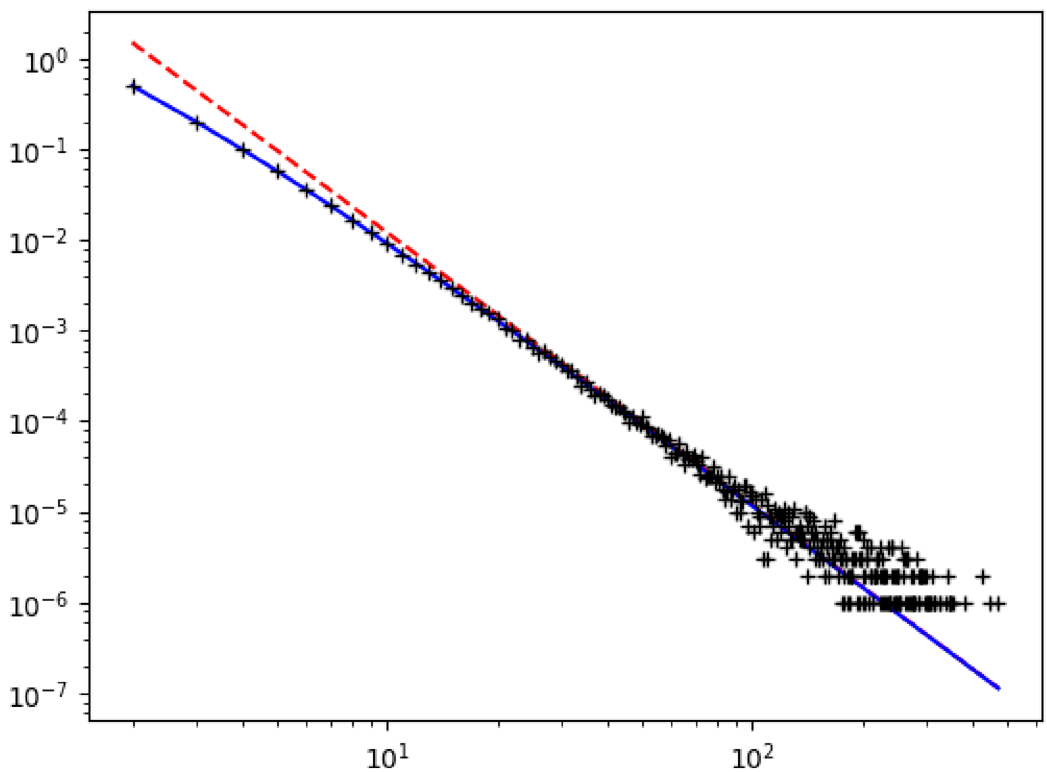


Figure 4: Left:

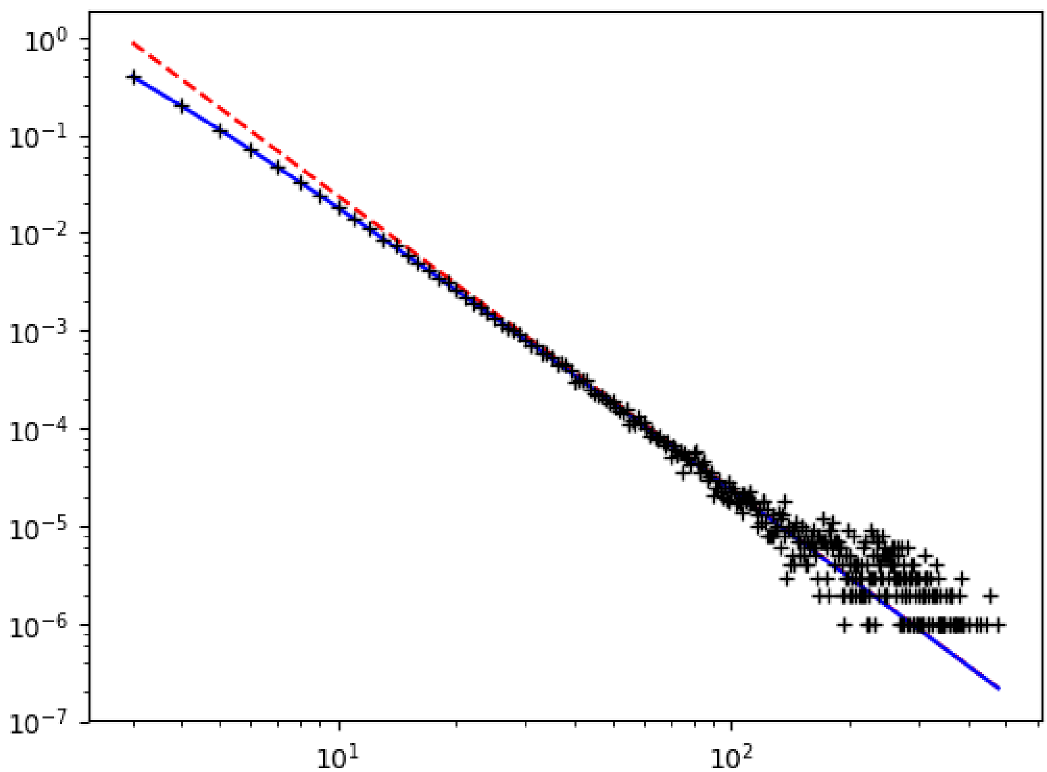


Figure 5: Left:

capture as much information as possible. However for large $k$, where our data is sparse, the bins are large, meaning the a lot of data is grouped together to help gain insight into the behviour. The geometric mean of each bin is then plotted. Figure 4,5,6 show this:

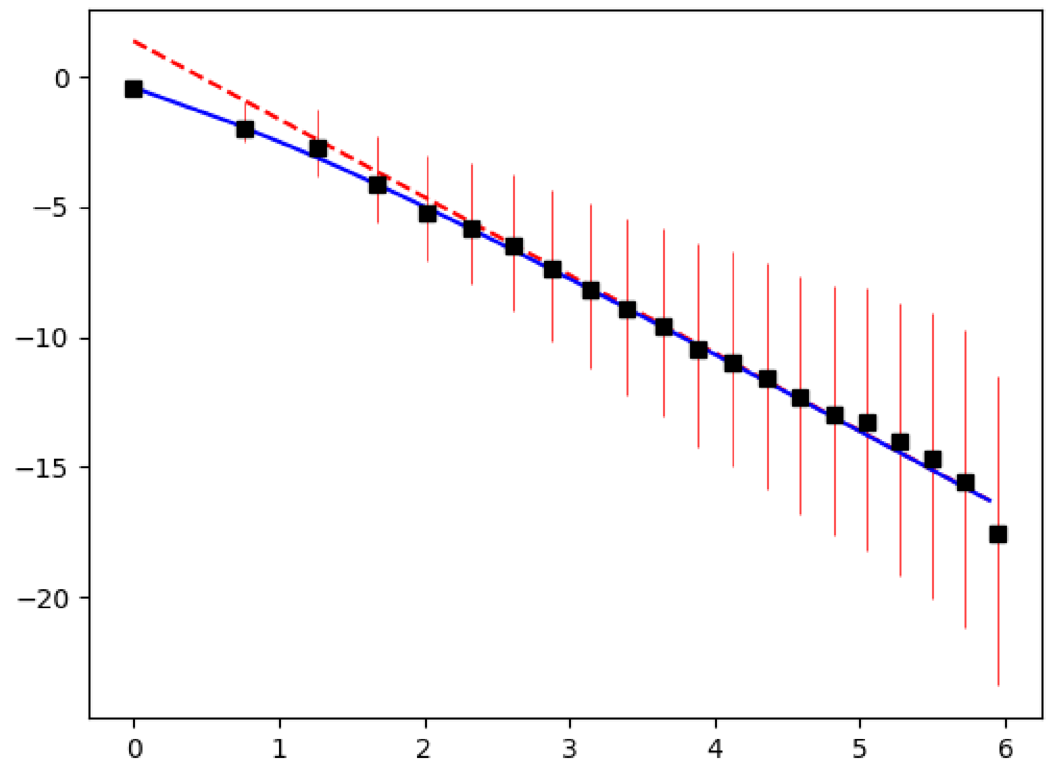


Figure 6: Left:

# 4 Largest K

-How does it depend on N? Theoretical 4 -Real data -Estimate uncertainties/errors

-data collapse?

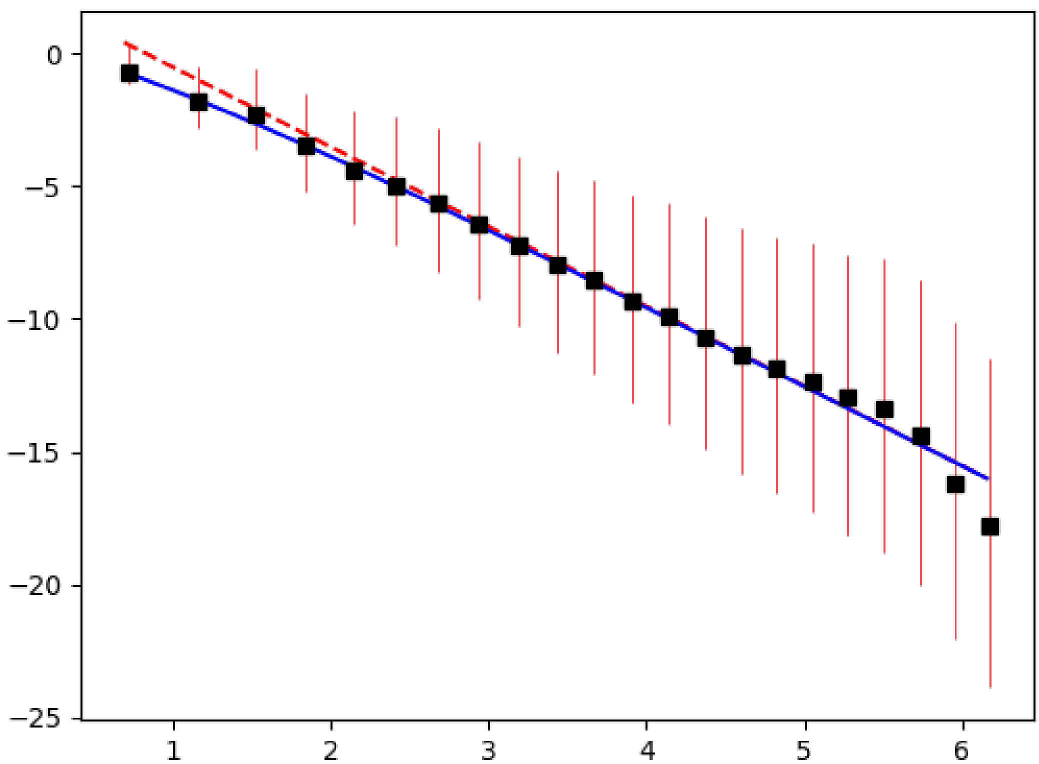


Figure 7: Left:

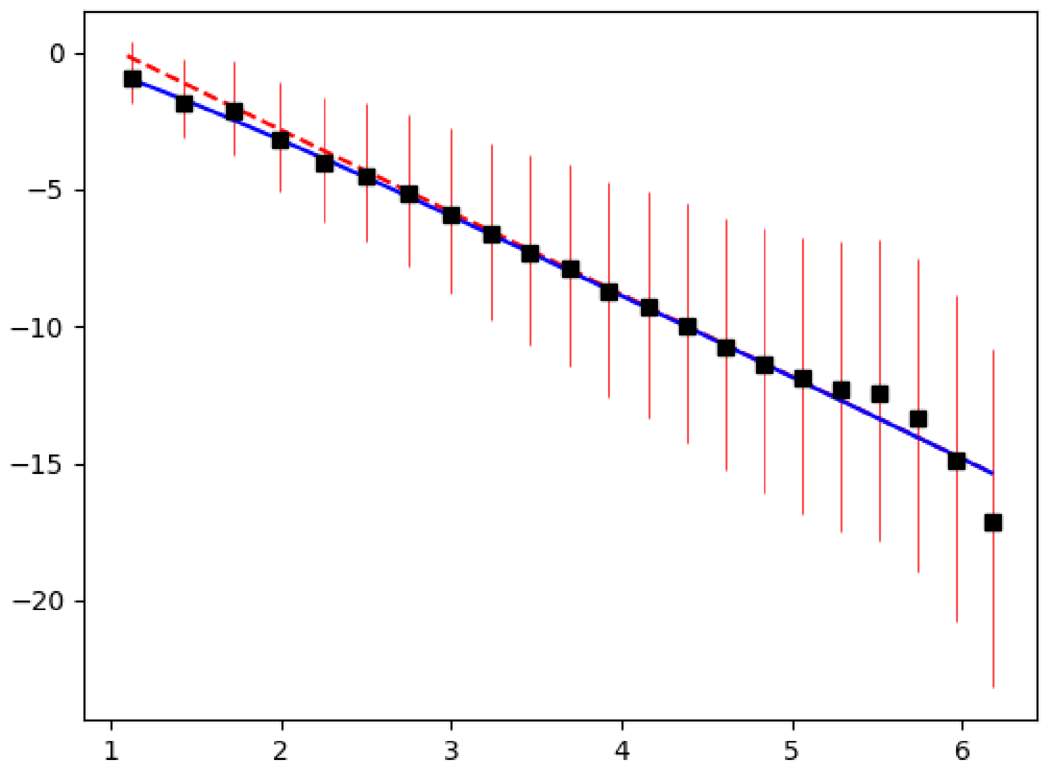


Figure 8: Left: