

# Preferential Attachment Coarse Projective Integration

Alexander Holiday

09/01/2013

## Introduction

The edge reconnecting model, detailed in [?], was implemented in C++. The system naturally converges to a limiting “graphon” (see [?]), and in order to increase the rate of convergence, coarse projective integration (CPI) was implemented. This brief outlines the model dynamics and reviews current simulation progress.

## Edge Reconnecting Model

The initial state of the edge reconnecting model is an Erdős-Rényi random graph with  $n$  vertices and  $m \sim O(n^2)$  edges. The system evolves as a discrete-time Markov chain, in which the following actions are performed during each step:

1. An edge  $e_{old}$  is chosen uniformly at random from the set of all possible edges,  $E(G)$ .
2. A vertex end of  $e_{old}$ ,  $v_1$ , is chosen uniformly from the two ends.
3.  $e_{old}$  is removed from the graph, and a vertex  $v_2$  is chosen from  $V(G)$  with a probability determined based on linear preferential attachment:

$$P(v_2 = v_i) = \frac{\deg(v_i) + \kappa}{2m + n\kappa}$$

where  $\kappa$  is a model parameter and  $\deg(v_i)$  denotes the degree of vertex  $i$  in  $V$ .

4. An edge  $e_{new}$  is added between  $v_1$  is connected to  $v_2$ .

Two distinct timescales arise from these dynamics,  $T \asymp n^2$  and  $T \asymp n^3$  where  $T$  is the number of steps. On the faster,  $O(n^2)$  scale, the degrees of the vertices may be considered constant, while the number of parallel edges between vertices changes. On the slower  $O(n^3)$  scale, the degree distribution evolves to a steady state value. While this separation of timescales has been proven on in [?], identifying them through numerical simulations is complicated by a couple of factors. First, the exact timescales themselves are difficult to discern. Both scales are really  $O(\rho_1 n^2)$  and  $O(\rho_2 n^3)$ , where the  $\rho_i$  constants are evaluated at the limit of infinite-sized graphs ( $n \rightarrow \infty$ ). This hints at the second, larger, problem: many of the results on the existence of these timescales in the first place are only valid in this large- $n$  limit. Simulation time then becomes problematic. and illustrate attempts to visualize these separate scales of evolution.

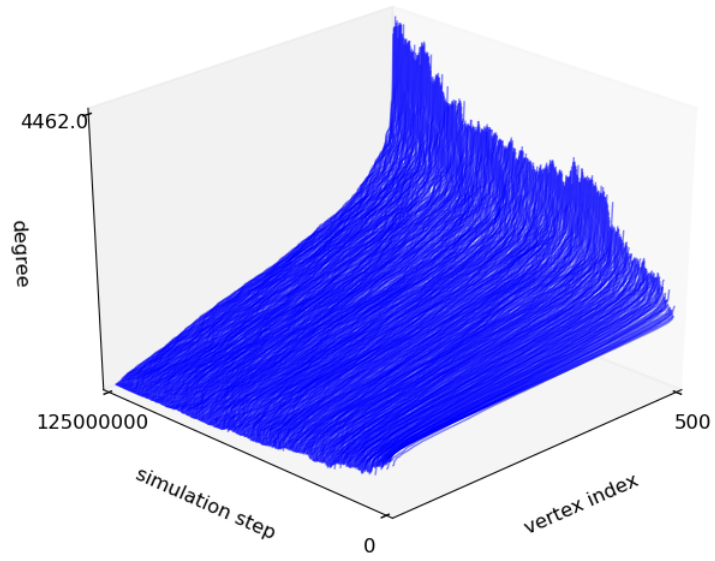
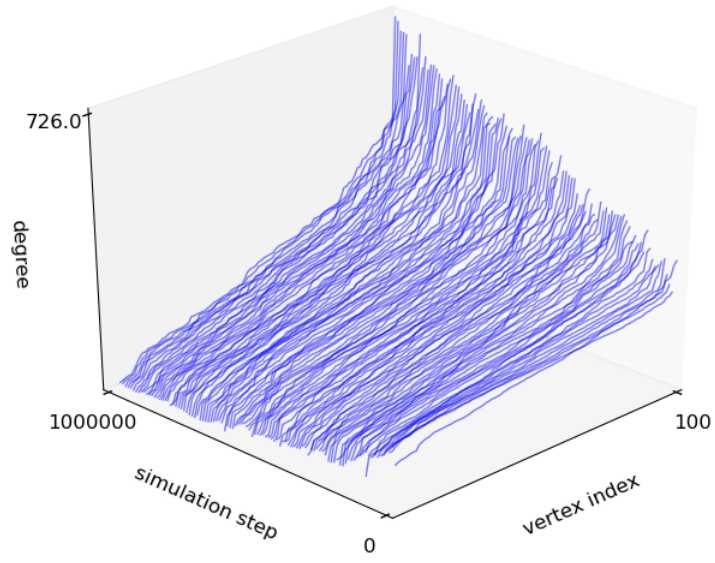


Figure 1: Evolution of degree distribution for  $n^3$  steps

In spite of the difficulties in accurately locating these different regions, progress has been made towards performing CPI on the system. Current work to this end is outlined below.

## CPI

Our approach in coarse-graining system dynamics is based on the existence of a gap in the spectrum of the adjacency matrix, and the subsequent ability to approximate  $A \approx \lambda_1 v^{(1)} v^{(1)\dagger}$  where  $A$  is the adjacency matrix of the system,  $\lambda_1$  is the leading eigenvalue of  $A$ , and  $v^{(1)}$  the corresponding eigenvector. 2 shows that clearly  $\lambda_1 \gg \lambda_i, i = 2, 3, \dots, n$ .

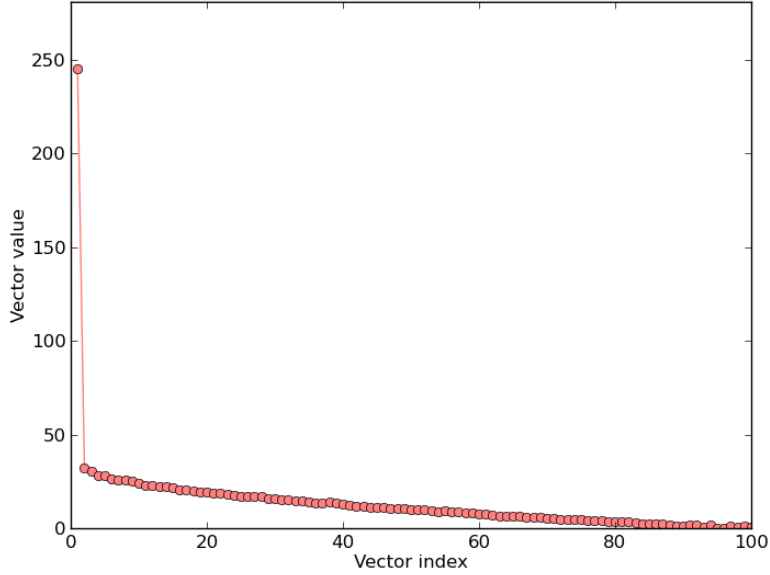


Figure 2: Eigenvalues of  $A$ ,  $n = 100$

3 illustrates the reconstruction of the adjacency matrix as  $A_{i,j} = \lambda_1 v_{i,j}^{(1)} v_{i,j}^{(1)\dagger}$  (after multiplication, each entry  $A_{i,j}$  was rounded to the nearest integer if  $i \neq j$  or to the nearest even integer if  $i = j$ ). Visually, the two correspond very well.

As  $\lambda_1 v^{(1)} v^{(1)\dagger}$  appears a good approximation of  $A$ , it was reasoned that we could use this eigenvalue/eigenvector combination as a coarse description of  $A$ . In order to further reduce the dimensionality of our coarse system, the eigenvector was fitted with a fifth-degree polynomial, as shown in ???. The six coefficients of this function were then used as a smaller set of coarse variables, leading to a final seven-dimensional representation of the system (coefficients plus eigenvalue). The following outlines the CPI framework:

1. Simulate the full edge reconnecting model dynamics for some number of steps until the fast variables are sufficiently slaved to the slow
2. Record the adjacency matrix as the system evolves on the slow manifold (in fact, it is more efficient to immediately compute the leading eigenvector, fit it with a polynomial, and store only these coefficients with the leading eigenvalue as time progresses)
3. Project forward the coarse variables (coefficients and eigenvalue)
4. Reconstruct a new adjacency matrix from the new, projected coefficients and eigenvalue

- (a) Compute a new eigenvector as  $v_i = \sum_{k=0}^{k=6} i^k c_k$  (where  $c_k$  represents the coefficients of the polynomial) and round to the nearest integer

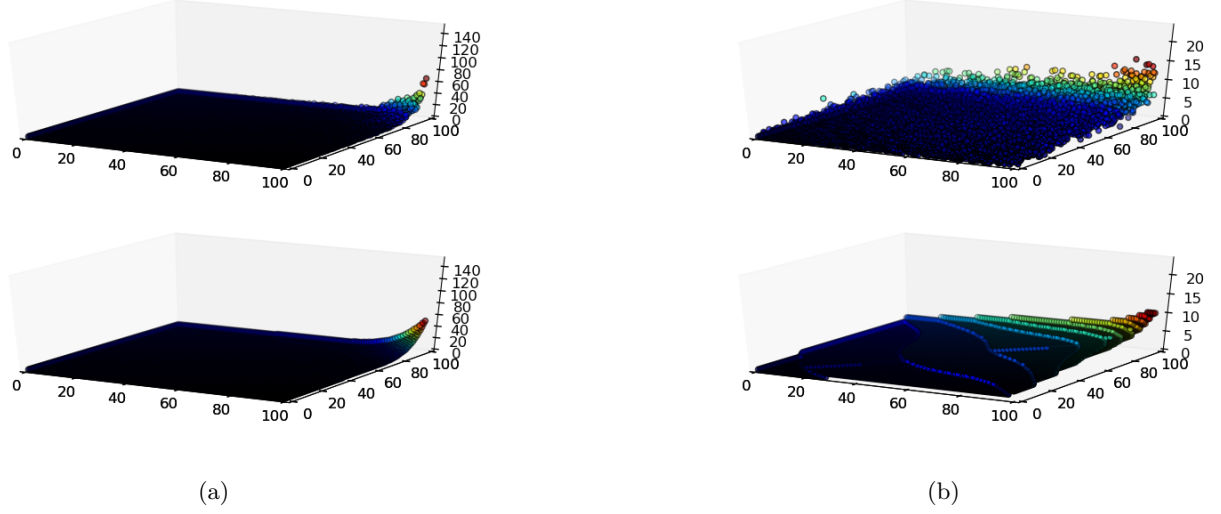


Figure 3: Original adjacency matrix taken from simulation (top) and its reconstruction from the leading eigenvalue/eigenvector pair (bottom),  $n = 100$

(b) Compute the new adjacency matrix as  $A_{i,j} = \lambda_1 v_{i,j}^{(1)} v_{i,j}^{(1)\dagger}$  and round as discussed above

5. Repeat step one until system reaches steady state

Preliminary results of this method are shown in ??, in which the evolution of the degree distribution is shown for both the full simulation and the a simulation in which CPI has been employed.

## Future work

The largest obstacle to further progress is the accurate identification (through simulation) of the disparate timescales present in this system's dynamics. Once these are found, a more efficient and accurate CPI method should readily follow.