Strategic Positioning in Network Centrality

Daniel Nachajon
dn2387@columbia.edu

Leo Touroul
leo.touroul@columbia.edu

Joshua Safyan
jds2258@columbia.edu

Erwann Campenon
emc2234@columbia.edu

April 26, 2017

Abstract

In the 15th century, the Medici family quickly rose to prominence among families in Florence and established a considerable infuence in Europe that lasted for hundreds of years. The centrality of the Medici within the network of Florentine families may have played a role in their ascent to power. Specifically, the betweenness centrality of the Medici far exceeded that of other leading families in Florence, suggesting that their strategic position within the network as an intermediary may have fueled their influence. In this work, inspired by the Medici, we show a simple framework of agent behavior and explore whether it leads agents to a disproportionate Betweeness Coefficient. We approach this task from several perspectives: 1) Theoretical Analysis, 2) Simulations, and 3) Learning Algorithms.

Keywords: Betweenness Centrality; Network Formation; Graph Theory; Reinforcement Learning.

1 Introduction

In the first half of the 15th century Cosimo de'Medici began a process that centralized the position and power of his family within the Republic of Florence. Though they never assumed the monarchy and were considered only citizens, the family produced 3 Popes of the Catholic Church and 2 Regent Queens of France. Furthermore they used the family textile business to found the Medici Bank. This bank was to become the largest bank in Europe at the time. While many factors likely contributed to the rise and acquisition of political power of the Medici family, one distinguishing feature that has been explored is the series of marriage contracts that Cosimo cemented. Among the prominent Florentine families, the Medici family played a central role when observed through the marriage networks. Their Betweeness Coefficient was more than twice as large as the next largest coefficient. Researchers have suggested that this disparity in centrality likely allowed the Medici and advantageous position within the economic and political movements of that period.

We have no real way of assessing what the legendary Patriach was thinking as he positioned his family, or if there was a method to his approach. However, if we believe that the Betweeness Coefficient explains a great part of the family's success, then one might ask what mechanisms lead to larger coefficients? In this work, we created a simple framework of agent behavior and explore whether this leads agents to a disproportionate Betweeness Coefficient. We approach this task from a few perspectives: 1) Theoretical Analysis, 2) Simulations, and 3) Learning Algorithms.

2 Discussion on the form of the utility function

As in any Network Game, the utility function can be written in two parts with a benefit and a cost, based on metrics of the graph. Thus, it can be written as

$$u_i(g) = b_i(g) - \sum_{j \in N(i)} c(j, g)$$

As explained in the introduction, we want to consider the betweenness centrality measure as the benefit as we believe this captures well the idea of being powerful or influential on a network. Furthermore, it is easy to see that the clustering coefficient would most probably bring either to the complete network, or to a network with several components in which all nodes are linked with one another. Therefore, this measure is not interesting if used on its own. However, even though we did not dive into this, it could be interesting to see if only some players had this measure of utility, what would happen? Another choice that has to be made is the dependency of the cost. For the analytical study, we will assume that the cost is a function of the degree of the node j to which node i is connected. This is an arbitrary choice that reflects the fact that it is more time-consuming to keep a good relation with someone who knows a lot of people and therefore doesn't necessarily need you, versus having a relation with someone who knows only you.

So the utility function we will use now will be of the form:

$$u_i(g) = BC_i(g) - \sum_{j \in N(i)} c(d_j)$$

2.1 The case of one active player and all the others are passive

As the player is the only active player, he will be the only one forming edges between himself and the other players so the network will evolve toward a star network. We can simply write the betweenness of the active player as:

$$BC_i(g) = \frac{d_i(d_i - 1)}{(n-1)(n-2)} c(1) d_i$$

Having this, we can think of two cases:

- If c(1) = 0, then creating a new edge does only increase the betweenness so the network will end up being a star network. This result holds even if the other players all create a random edge with probability less than 1/(n-1) each round. In that case, the star network will eventually emerge and the active player will destroy edges that would be created between two other nodes than himself at the round after they are created.
- If c(1)>0, then we must see how this influences the choice to create edges or not.

We can first notice that the utility of player i is a polynomial of degree 2 that is equal to 0 when the degree of i is 0 or 1 so player i has no incentive to create the first edge, even more if the cost is high. Then, this utility is 0 again when

$$d_i = 1 + c(1)(n-1)(n-2)$$

The derivative of this utility regarding the degree of node i is

$$\frac{\partial u_i}{\partial d_i} = \frac{2d_i - 1 - c(1)(n-1)(n-2)}{(n-1)(n-2)}$$

So the biggest the network and the highest the cost, the less likely it is that the active player would try to create any edge, because as long as his degree is less than

$$d_i = \frac{1 + c(1).(n-1).(n-2)}{2}$$

he will rather destroy edges than create any. This concludes the basic case of one active player. Moving on, we will see some interesting cases with two active players and the others passive.

2.2 Some results with two active players

A first natural case to consider is a *double-star network* in which the two active players are each connected to all passive players. In that case, if there are n nodes in the network:

$$u_{a_1}(g) = u_{a_2}(g) = \frac{n-3}{2(n-1)} - (n-2)c(2)$$

This utility is positive only if

$$c(2) < \frac{n-3}{2(n-1)(n-2)}$$

So we see that the cost has to be very low, overall if n is large. Another interesting result of this simple case is that it doesn't matter if the two active players form an edge between themselves or not, it will not change their betweenness, Nonetheless this edge has a cost so they would rather not form it.

Another case we considered is the case of the network with two active players, reaching a state in which each active player is at the center of a star network. Let's assume that the active players are a1 and a2 and that a1's component has n1 nodes and a2's component has n2 nodes. Then:

$$u_1(g) = \frac{(n_1-1)(n_1-2)}{(n_1+n_2-1)(n_1+n_2-2)} - n_1 c(1)$$

$$u_2(g) = \frac{(n_2-1)(n_2-2)}{(n_1+n_2-1)(n_1+n_2-2)} - n_2 c(1)$$

A first interesting thing to notice is that if either a1 or a2 wants to connect to the other component, whoever it connects to, its utility is going to be the same. So he would rather connect with a passive player (myopic strategy) to have the least cost. But let's look at the pairwise stability: This is only achieved if

$$c(2) \geq \frac{max(n_1,n_2)(2min(n_1,n_2)+1)}{(n_1+n_2-1)(n_1+n_2-2)}$$

2.3 Going toward all active players

First, we can say easily that the complete network cannot be an equilibrium under our model because then the betweenness of each player would be 0 while his cost would be $(n-1) * c(n-1)$. Therefore, there is no pairwise stability as each player can destroy an incidental edge without hurting his betweenness and only limiting its cost, unless the cost is 0 in which case we have pairwise stability: everyone's betweenness is 0 and destroying an incidental edge does not change it. However, if we authorize destroying edges between two other nodes, then the complete network won't emerge either.

The other interesting case we've studied is the case of a Watts Strogatz Network. If the rewiring probability is zero to start with, we have a ring network. Then, the betweenness of each node is approximately, considering that n is large:

$$BC(g) = \frac{2}{(n-1)(n-2)}\frac{1}{2}.2.\sum_{j\in[1..\frac{n-1}{2}]} j \approx \frac{n-3}{4(n-2)}$$

which goes to 0.25 as n goes to infinity.

Let's analyse the case in which an edge is removed. Then, it is easy to see that for any node at a distance i that is less than n/2 of this removed edge, the betweenness is:

$$BC_i(ringnetwork - 1) = i(n-i-1)\frac{2}{(n-1)(n-2)}$$

We see that this is an increasing function of i on [0,n/2] So from a greedy perspective, all players would destroy the edge further from them and we would restart from the empty network.

We trivially notice that adding an edge incidental to him is less interesting for a node as it increases his cost, and does not increase his betweenness as much as the removing strategy.

3 Betweeness Coefficient

We take the standard definition of Betweeness Coefficient (BC) in this work

$$BC(v) = \frac{2}{(n-1)(n-2)}\sum_{s\neq v\neq t}\frac{\sigma_{st}(v)}{\sigma_{st}}$$

That is, for all pairs of nodes not v, we care about the fraction of shortest paths between then that contain v. This summation is normalized by the number of edges possible in a graph of size n. The Medici were able to dominate this metric.

An alternative to using the betweenness centrality is to use the normalized betweenness centrality, which would be equal to:

$$NBC_i(g) = \frac{BC_i(g) - \min_j BC_j(g)}{\max_j BC_j - \min_j BC_j(g)}$$

The advantage of this measure is that the objective of every active player is to get to 1.

4 Agent Based Framework

Among the questions we seek to answer is, given a simple set of actions and agent preferences, can we induce superior betweeness? Throughout our models we allow agents to perform one of 3 actions in a turn:

1. Create an edge in the network
2. Delete an edge in the network
3. Do nothing

While these are the actions we permit active players in our network to play, these actions only make sense within some context of utility or preferences. In other sections of this report, the choice is deterministic and so we assess the utility of greater BC against some notion of cost in creating links. In this section, we really only focus on actions 1 and 3 by turning the decision making process probabilistic. Based on pre-allocated agent preferences, an agent encounters a target node to form a link and either decides to form or not form the edge. Admitedly, these will be subjective choices but the objective here is to garner intuition. Loosely speaking, agents in the network are classified as either *active* or *passive*. Passive agents follow a very simple behavior and have no desire to really play the game. As a starting point, passive nodes operate in an Erdos-Renyi fashion in each turn of the game. On the other hand, *active* players are allowed to express preferences through two broad avenues: degree distribution and clustering coefficient.

In our simulations, each active node samples the network according to the degree distribution to select a target node for potential attachment. We let active nodes explore the network in three ways:

1. Preferential - Nodes with higher degree are more likely to be selected
2. Uniform - All nodes are equally likely, irrespective of degree
3. Anti-Preferential - Nodes with lower degree are more likely to be selected.

It is easy to see that a single active player wishes to be the center of a star network and leave it at that to maximize his BC. If he had his way, every other node would have degree 1. This is the rationale that underpins the choice to sample in an Anti-Prefential way. However, if we suppose that a player see two disjoint star networks, under certain conditions he might wish to bridge the two together, thus motivating a Preferential attachement mode.

Once an *active* node has sampled a target node, he must decide whether or to not to link to the target. This decision is made as a function of the clustering coefficient of the target node. As the clustering coefficient goes from 0 to 1, the target node transitions from being isolated/star-like to a clique. We adopt a relatively simple set of basis functions that turn clustering coefficient into a link probability. This probability is then used to determine if the edge should be formed.

1. UNIFORM - No bearing, always wants to form the edge
2. LTRIANGLE - Higher clustering is more favorable in a linear fashion
3. RTRIANGLE - Lower clustering is more favorable in a linear fashion
4. MODAL - Some target CC is the most desired and the probability falls off linearly for greater and lower clustering. Unless otherwise specified, we center this profile around 50% clustering

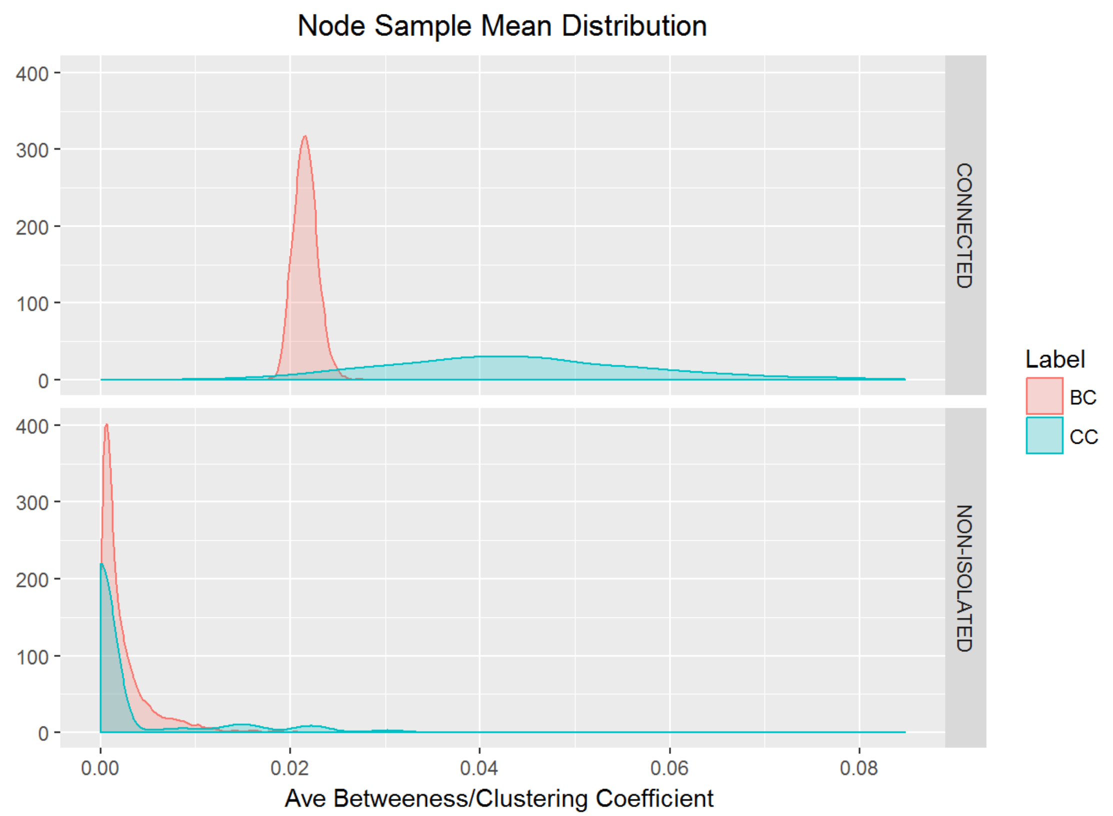
In running these simulations, we explore how expression of these are linked with BC.

5 Approaching Betweeness via Simulated Preferences

In all of the simulations we ran, we fixed the network size at 100 nodes. This is to make the network large enough but also tractable. As we proceed, we make the interactions more complex and crowded. First, we begin by describing the *passive* nodes who operated in an Erdos-Renyi way. Our initial explorations have them linking to any node random way with link probability $p = 0.01$ which can be shown to be a transition probability for each node having degree at least 1. Later on, we explore a mode connected model for the passive players. Next, we explore how a single *active* node, faced with 99 *passive* nodes fares under the simulation framework. Next, we explore a more crowded game, where 40% of the nodes are active. Finally we explore whether or not we can apply a schedule to the preferences, that is, make them time-dependent, in order to exceed the BC.

5.1 Benchmark: Erdos-Renyi

Before attempting to understand the effects of our model, we generated 1000 samples from a Erdos Renyi random graph with $n = 100$ and $p_1 = 0.01$ and $p_2 = log(100)/100$. For each sample, we compute the average BC and average Clustring Coefficient (CC) over each of the 100 nodes. The empirical densities and quantiles are shown below.

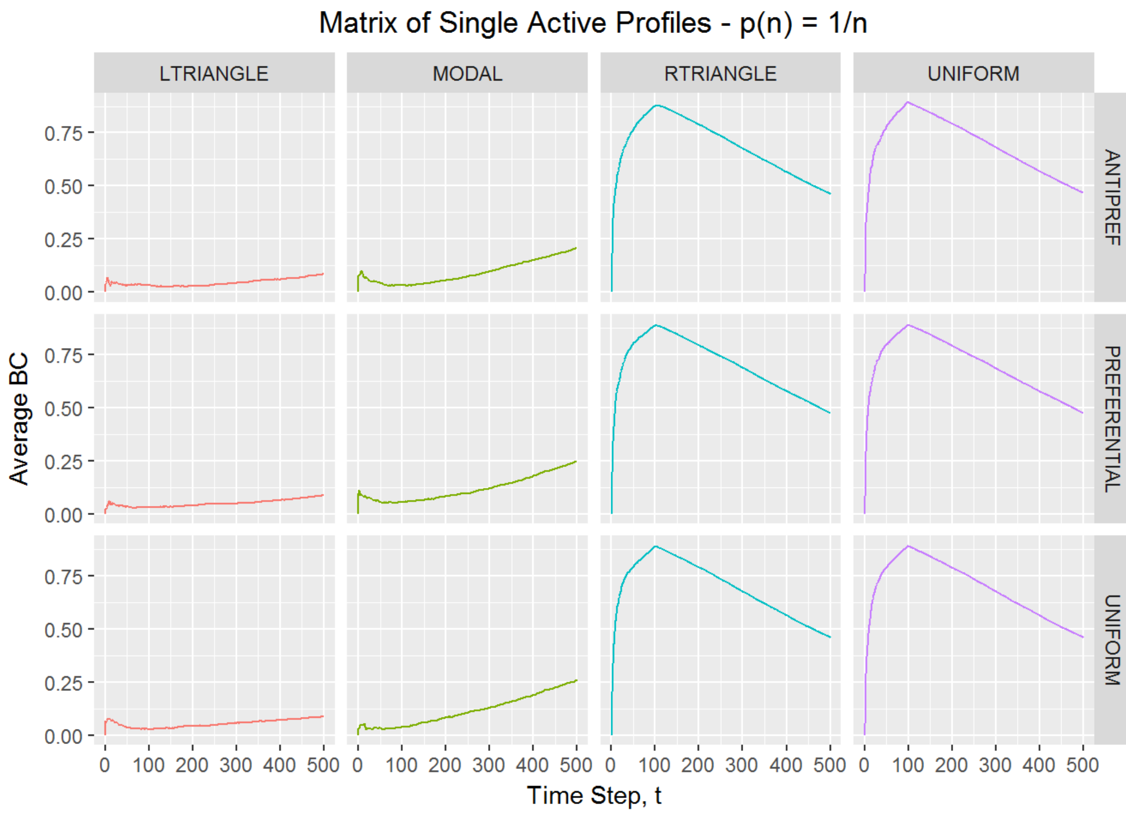


Clearly, left to their own devices, *passive* nodes are not likely to find themselves in very central positions. While it is true that the behavior under the two transition functions is vastly different, neither produces extraordinarily high BC or CC. These two do offer some insight into how *active* player preferences will fare against a random backdrop.

5.2 Single Active Player Performance

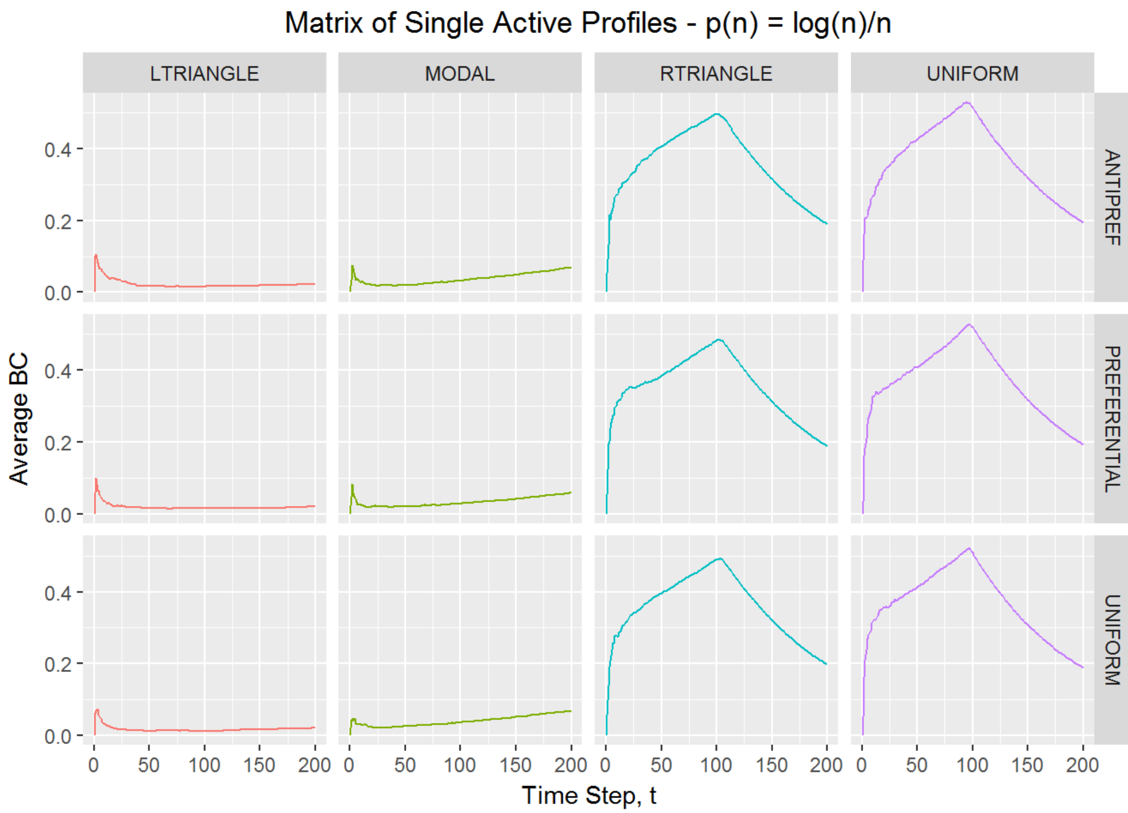
Before comparing these preferences in conjunction, it is instructive to analyze what happens in the case of a single active player expressing their preferences in the context of the Erdos-Renyi random graph described previously. To explore these behaviors, we ran 20 simulations over 500 timesteps of the game for each of the 12 degree - clustering preference combinations. The behavior of the 99 passive nodes in this network will also have a great deal of bearing on the ability of the active node to achieve prominent centrality. Therefore, we consider two case. In the first case, the passive nodes form edges with probability $p(n) = \frac{1}{n}$. This can be shown to be a transition function for each node to have degree at least 1 with almost surety. In the second case, we use the link probability $p(n) = \frac{\log n}{n}$, the transition function for the entire graph to be connected. The matrix plots below shows the average BC of the active player at each point in time.

5.3 Passive Probability, $1/n$



There are few key observations to make from these results. Recall that at $t = 0$ the graph is just the empty graph over 100 nodes. At the onset, the defualt behavior is to make a random edge to get things going. We note that *RTRIANGLE* and *UNIFORM* preferences over clustering coefficients behave very similiarly, rising steadily and rapidly until approximately $t = 100$, at which point the active player begins conceeding centrality in an almost linear fashion. This is expected; most nodes have few edges and hence small clustering coefficient. Active players with a penchant for smaller coefficients will want to link with almost any node with high probabiliy. *UNIFORM* clustering preference means that the player will choose to link with probabiliy close to 1 which explains why the two modes operate similiarly. On the otherhand, the *LTRIANGLE* mode makes it almost impossible for active players with this preference to want to link with almost any node at the onset of the game. Finally, the *MODAL* clustering expression (with a target CC of 50%) takes some time to build a critical mass of nodes with large enough clustering coefficients. At which, the player starts offering higher link probabilities and so its Betweeness Centrality begins to rise. Clearly, more than 500 time iterations are needed to maximize the BC of a *MODAL* player in this context. Lastly, we note that in the single player game, the 3 different attachment modes don't offer much difference. This is probably due to the fact that there is not enough of a degree distribution to distinguish between low and high degree nodes.

5.4 Passive Probability: $log(n)/n$



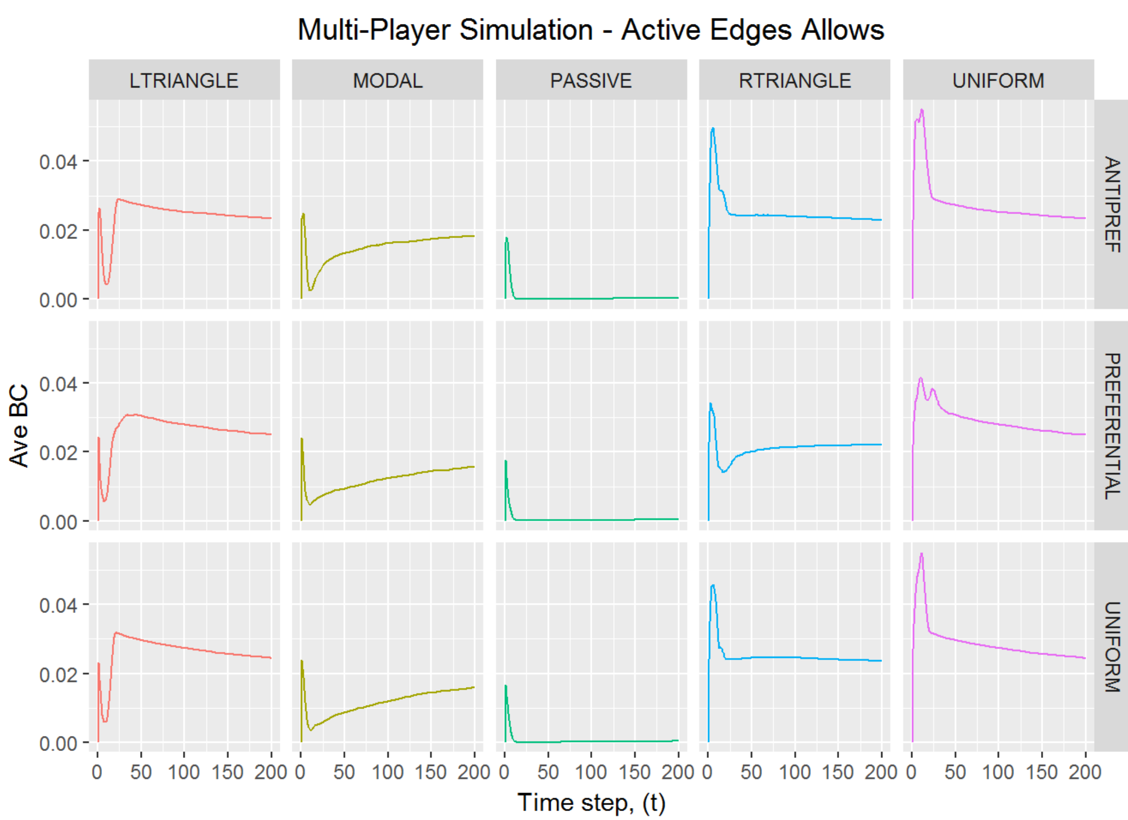
The behavior of the actives against a more connected tapestry of nodes has a marked effect. Though we again see that clustering preferences of *RTRIANGLE* and *UNIFORM* peak at around the $t = 100$ mark, we note that the magnitude of the BC drops from the roughly 88% they were able to acheive to approximately 50% and 53% respectively. The decline thereafter is also more accelerated at the onset and the qualitative shape of the curves is less similar. It appears that the peaked nature of the passive BC makes *RTRIANGLE* less likely to connect than in the non-isolated case as well as versus the *UNIFORM* player. We are also able to tease out more insight into what is inhibiting the other two clustering preferences. Though these are on average more likely to form an edge, this effect is not strong enough to overcome the fact that in the connected case it is simply harder to be the central node.

5.5 Multiple Active Agents

Having seen the effect of these preferences in isolation, we next consider a game of 100 players, 60 *passive* and 40 *active*. We play the game thrice, once for each of the different degree distribution preferences and in each of the games we include 10 active players for each of the clustering coefficient preferences. Each sample was run for 200 time-steps. Each game was simulated 20 times. In contrast to the previous simulation,this game is far more crowded as 40% of the network can be viewed as trying to maximize its BC.

When the network is so crowded we run the risk of active players deciding it is reasonable to form an edge with another active player. In deterministic, strategic play, this is quite possible as one player may deem it worthwhile to let one player connect with his component if the resulting BC is advantegeous. To benchmark this effect, we run the same framework twice. First allowing active player connections and the second forbidding them. This will make fewer edges overall, so we need to take those result in context. In both of these cases, the passive nodes form edges with probability $p = 1/n$.

5.6 Crowded Network - Permitting Edges Between Active Nodes



We see clear evidence of the crowding and clashing having an impact on the BC of the players. For comparison, we include the trail of the *passive* nodes to see the results in comparison. A few things jump out from these results. Perhaps the most intriguing is that the all the active profiles appear to be converging to an average BC of about 2%. This is far lower than what was acheived with a single active player but this time, we have that active play is at least convergent. Again, we see that players who favor clustering coefficients in a *RTRIANGLE* and *UNIFORM* fashion rise quickly. But this time they peak and very quickly fall off. Those players with an *LTRIANGLE* profile this time experience and greater ability to form edges, but they too find a hard ceiling in the largest BC they can enjoy. The *MODAL* players are the only ones who seem to experience a rising BC in the 200 time steps.

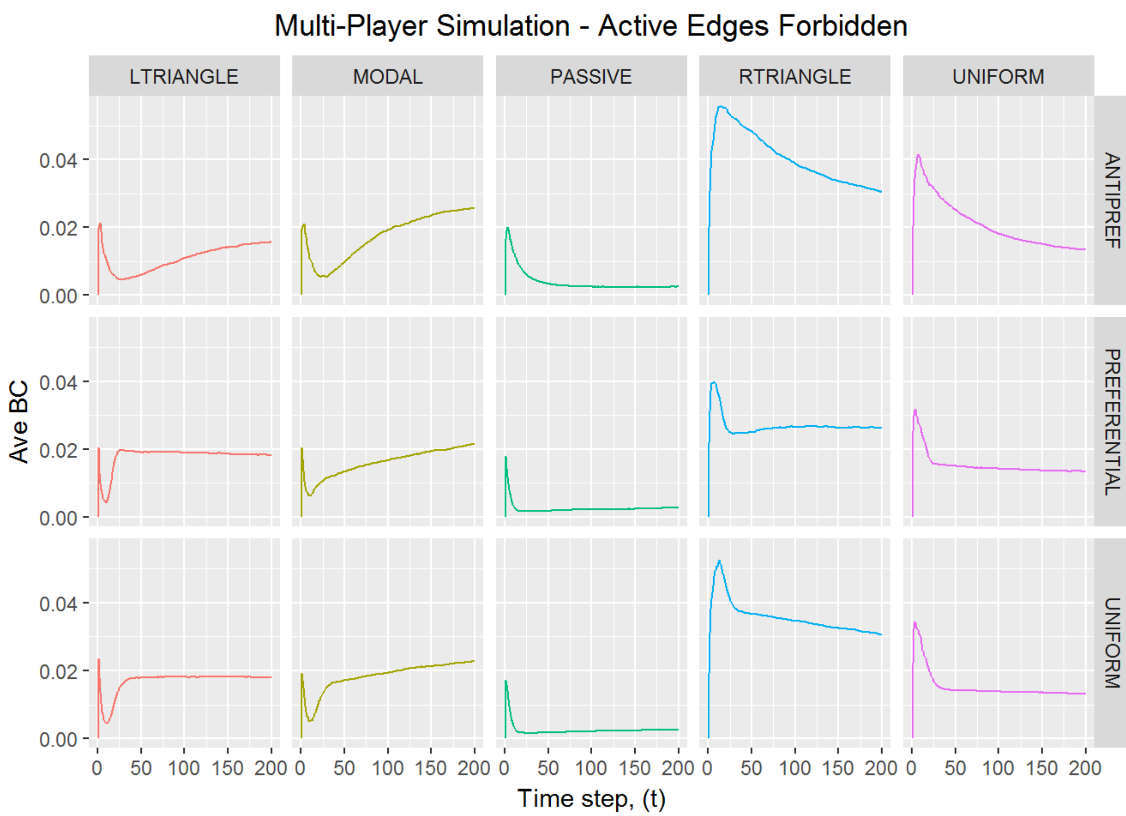
One follow-up worth exploring would be to ascertain whether the *MODAL* players' BC tapers off in growth because the clustering coefficient moves away from their preferences or because the network has far more edges this time around.

Lastly, we note that in this crowded scenario, The choice of *PREFERENTIAL* or *ANTIPREFERENTIAL* degree distribution sampling has a visual but mild effect. The table below tallies the max average BC and the average time to achieve the maximum under the various degree and clustering preferences

ATTACH_MODE	CLUSTER_MODE	TMAX	BMAX
ANTIPREF	PASSIVE	2.30	0.0191016
ANTIPREF	UNIFORM	9.55	0.0580736
ANTIPREF	LTRIANGLE	14.40	0.0324121
ANTIPREF	RTRIANGLE	4.80	0.0523840
ANTIPREF	MODAL	20.65	0.0276953
PREFERENTIAL	PASSIVE	1.20	0.0178030
PREFERENTIAL	UNIFORM	13.25	0.0438786
PREFERENTIAL	LTRIANGLE	33.80	0.0320695
PREFERENTIAL	RTRIANGLE	3.50	0.0370024
PREFERENTIAL	MODAL	10.25	0.0242682
UNIFORM	PASSIVE	1.05	0.0167964
UNIFORM	UNIFORM	10.60	0.0558880
UNIFORM	LTRIANGLE	19.50	0.0322172
UNIFORM	RTRIANGLE	4.65	0.0473946
UNIFORM	MODAL	30.35	0.0247198

5.7 Crowded Network - Forbidding Edges Between Active Nodes

The following results were obtained by running a simulation much like the one before but simply forbidding edges between active nodes. The implementation of this prohibition meant that if active players chose to link, the game would later nullify their action. Broadly speaking this made any edge formation approximately 40% less likely at the onset. This is simple and crude approximation. When looking at the results we have to remember that the active players might have wished to form more edges when uninhibited.

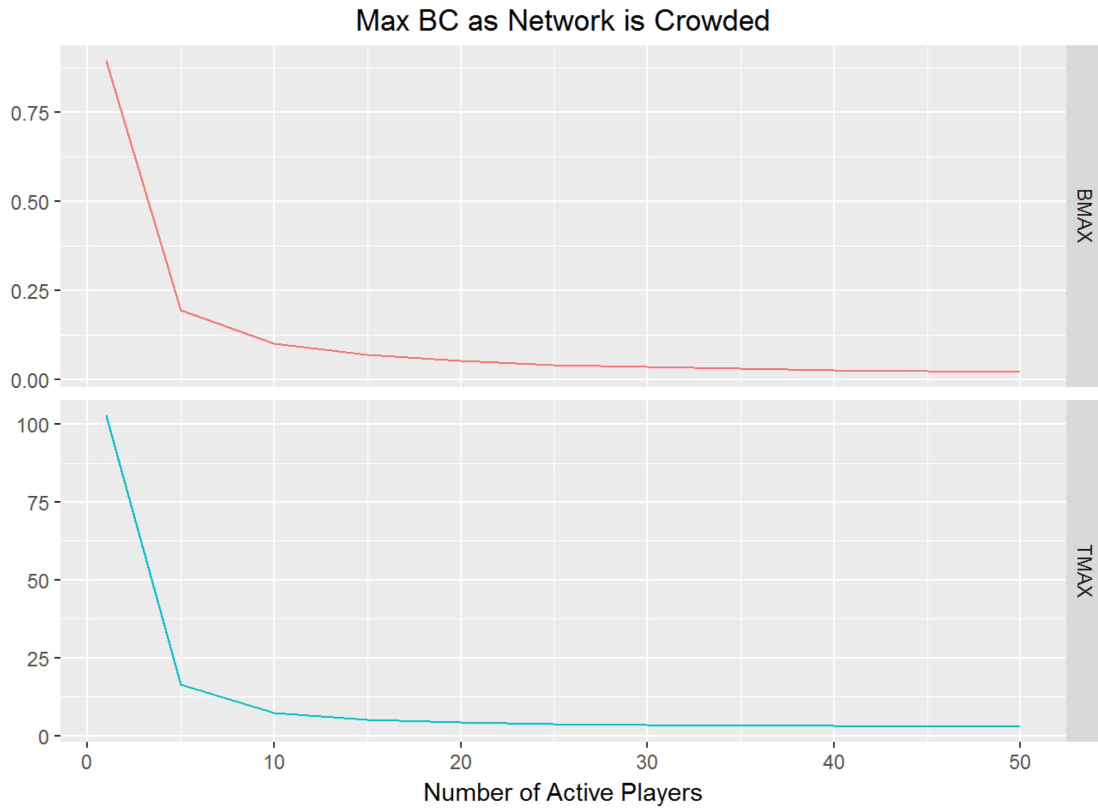


This graph can almost be thought of a core of passive nodes with satellites of various active players. The degree of the passive nodes increases much more than under normal Erdos Renyi conditions. One implication of this is that actives who sample in an *ANTI-PREFERENTIAL* manner experience either increasing betweenness or slower decline after peaking. This model also seems to have broken the tie between *RTRIANGLE* and *UNIFORM* CC preferences. The former is able to achieve a larger BC and maintain it for a longer period of time. Though sampling nodes proportionally or uniformly with respect to degree seems to have a faster adverse effect on the BC, they also seem to taper of in decrease in the long run. This most likely indicates that the overall CC of the typical passive node is low so when an edge forms it is like linking mini-star with many arms. Clearly this is beneficial to increasing one's BC.

ATTACH_MODE	CLUSTER_MODE	TMAX	BMAX
ANTIPREF	PASSIVE	3.50	0.0212240
ANTIPREF	UNIFORM	7.95	0.0480027
ANTIPREF	LTRIANGLE	47.10	0.0241467
ANTIPREF	RTRIANGLE	15.30	0.0591610
ANTIPREF	MODAL	118.50	0.0271750
PREFERENTIAL	PASSIVE	1.25	0.0184125
PREFERENTIAL	UNIFORM	4.05	0.0368025
PREFERENTIAL	LTRIANGLE	36.80	0.0242127
PREFERENTIAL	RTRIANGLE	6.45	0.0437223
PREFERENTIAL	MODAL	124.60	0.0237875
UNIFORM	PASSIVE	1.50	0.0180348
UNIFORM	UNIFORM	4.15	0.0378828
UNIFORM	LTRIANGLE	25.85	0.0251519
UNIFORM	RTRIANGLE	10.30	0.0548255
UNIFORM	MODAL	139.15	0.0238788

5.8 Increasing the Crowdedness of the Network

Clearly, the number of so called *active* players will impose a network effect on how large any single, dogmatic agent can raise his BC. To understand this effect a little better, we simulated over a network of $n = 100$ where all the passive nodes create edges randomly with probability $p = 1/n$ and gradually added more actives. For simplicity, we only considered those whose clustering preferences were *RTRIANGLE* and who sampled degrees in a *UNIFORM* fashion. Though this is not a general approach, we expect a very similar behavior with other combinations.



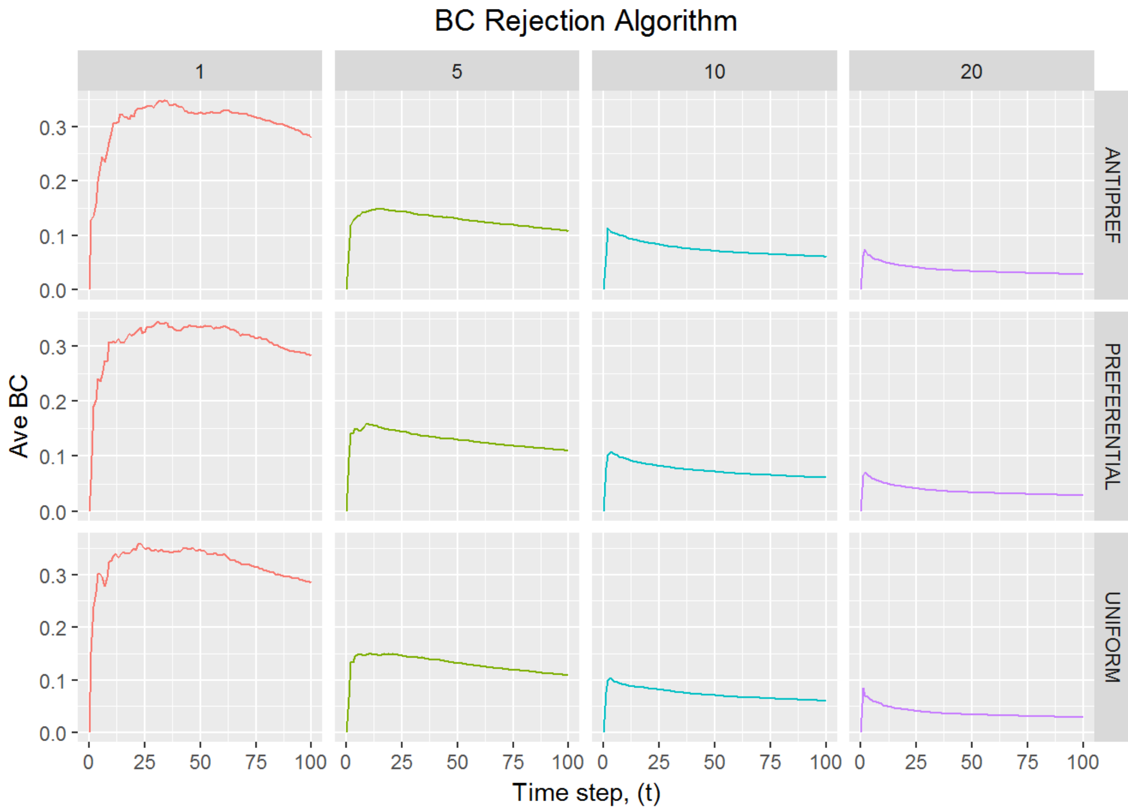
The chart above highlights the notion that as we flood the network with more, symmetric active players they appear to crowd each other out. One limitation of this analysis is that we do not consider edge deletion actions but on the other hand, the probability of edge formation should in theory decrease over time.

5.9 Betweeness Rejection Simulation

In the previous section we let *active* players evolve in accordance to certain preferences over degree and clustering coefficient. In later sections of this report we will explore learning algorithms and compare them to a greedy approach. However, to motivate that analysis we considered a simple Rejection Algorithm not over CC but directly over BC. Target nodes were still found via degree distribution sampling. Once the target node was found, the active agent, s considers the ratio

$$r = BC_s(g + \{s, t\})/BC_s(g)$$

And will accept the new edge, (s, t) if $u \leq r$, where $u \sim Unif(0, 1)$. This algorithm should steer our active player through the complex landscape, allowing for decreases in the hopes they will find better positioning later on. Edges between active players are allowed. Passive node connect in a fashion that induces full graph connectivity ($p = \frac{\log n}{n}$).



One nice feature is that the active nodes find their peak BCs much, much faster than before. Their rate of decline is also slower once they have peaked. The shape and curvature of the decline is qualitatively different that what has been previously observed. Note that whereas an agent may not wish to add more edges, the random behavior may change the BC of a node between rounds. That is, there is no guarantee that node i will start round $t + 1$ with the same BC as what if finished round t with. Furthermore, though we might have wished for the nodes to eventually reject all new edges, the simulations appear not to have converged by timestep 100.

6 Discussion of Simulated Preferences

We provided a simple framework for allowing so called *active* nodes to traverse a network and form connections. Our objective was to assess whether expressing preferences over secondary characteristics of the nodes can guide the active nodes to a higher BC. Active players were allowed to chose one of 3 different modalities over the degree of a node and among 4 different modalities over the clustering coefficient. When we let a lone *active* among a plethora of Erdos Renyi passive players, we found that those actives who preferred smaller clustering coefficients were able to do reasonably well until the network became too well connected. Our approach never found a structure over the passive nodes that let an *LTRIANGLE* player emerge as a victorious strategy. Less easy to explain was the experience of a *MODAL* player (centered at 50% CC); while the network on average made it harder for those players to form links, by the time the were able to form links the centrality may have been diluted.

When we challenged the actives with 9 other copies of themselves and 30 other active players, the crowding made it quite difficult to raise and keep the BC elevated. It seems apparent that the structure of the graph will play an instrumental role in how much scope a probabilistic rule has for reaching a larger BC. A fuller examination of the properties of our model would be to subject our active nodes to graphs like the Watts-Strogatz or even a random graph with a similar degree distribution to that of the Florentine network.

It's worth noting that this work is not meant to be an investigation into true equilibrium states over this game. The absence of a direct cost makes it difficult to reason about notions of pairwise stability over nodes. This work didn't consider an explicit utility function. Instead, we relied on the encoded preferences of the active nodes to steer them towards lower probabilities of link-formation. One possible extension of this would be to either consider random ways to induce edge deletion or a more dynamic way to let agents alter their preferences as they explore the changing network structure. Ultimately, the merits of any rule based simulation will be a mirror reflection of the underlying graph and its evolution. If these dynamics are not conducive to centrality then active players will not be able to hold their position indefinitely and crowding will accelerate the process.

7 Approximate Centrality

We wish to consider scenarios in which greedy agents only have a stochastic view on the network betweenness centrality measures. Our reasoning is two-fold: betweenness centrality is an expensive ($\sim O(|V|^3)$) operation for non-sparse graphs, and in the real world it is unlikely that actors would be able to fully ascertain their exact betweenness centrality. Specifically, we examine whether we observe different network formation behavior by the greedy agents when they act based on stochastic measures.

To this end, we implemented a sampling approximation algorithm for betweenness centrality from Riondato and Kornaropoulos (2016). Unlike the k pivots sampling available in NetworkX, the sampling algorithm builds a full representation of the betweenness of the network with probabilistic guarantees on error bounds using the VC-dimension of range sets of shortest paths. Specifically, the user can tune two variables, ϵ and δ, such that the algorithm produces betweenness measures that are within ϵ of the correct values with at least $1 - \delta$ probability (Riondato and Kornaropoulos (2016)).

The number of samples r needed by the algorithm is given by

$$r \leftarrow \left(\frac{c}{\epsilon^2}\right)\left(\lfloor log_2(\mathrm{VD}(G) - 2)\rfloor + \ln(\frac{1}{\delta})\right)$$

where c is a constant (estimated to be $\sim .5$), ϵ is the additive error bound, $1 - \delta$ is the probability of at most ϵ error, and $\mathrm{VD}(G)$ is the Vertex Diameter of the graph, i.e., the longest shortest path in the graph. $\mathrm{VD}(G)$ can be approximated in $O(|V|+|E|)$ by picking uniformly at random a vertex v, finding all shortest paths from v, and adding together the length of the two longest shortest paths from v. This approximation has the following guarantee:

$$\mathrm{VD}(G) \leq \tilde{\mathrm{VD}}(G) \leq 2\mathrm{VD}(G)$$

As $\mathrm{VD}(G)$ increases sublinearly with the size of the graph, this method scales well to larger graphs and more connected graphs. The algorithm uses a weighted random back-up sampling of shortest paths in the graph for r samples to determine an approximation for betweenness centrality Riondato and Kornaropoulos (2016).

A relatively naive implementation confirmed that runtime gains are realized on larger networks.

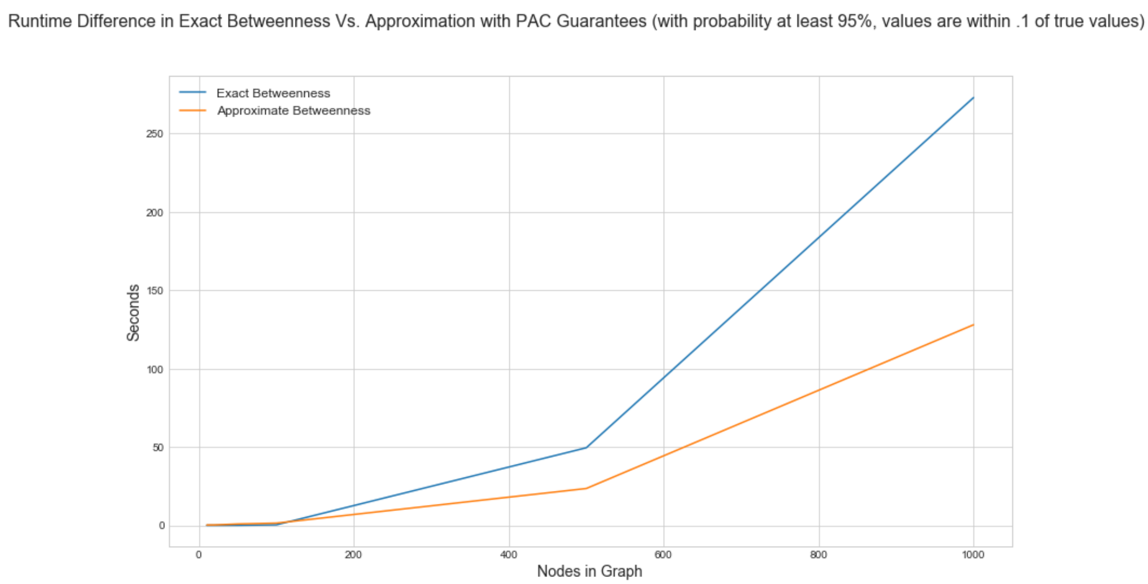


Figure 1: Sampling Approximation of Betweeness Centrality runs faster on larger graphs

7.1 Simulating Approximate Greedy Play

With all members of a network active and using symmetric approximate greedy strategies, the game yields an uneven distribution of betweenness. Given that each player is acting based on a stochastic view of the betweenness centrality, it is not surprising that players' strategies would diverge from optimal play. This result provides a skilled but not fully-random baseline for differentiated betweenness in network games where the outcome of interest is the betweenness centrality. That is, a certain amount of variation in betweenness is to be expected due to information assymetries and imperfect information. Thus, even with non-random symmetric actors we should expect to observe non-uniform realizations of betweenness centrality.

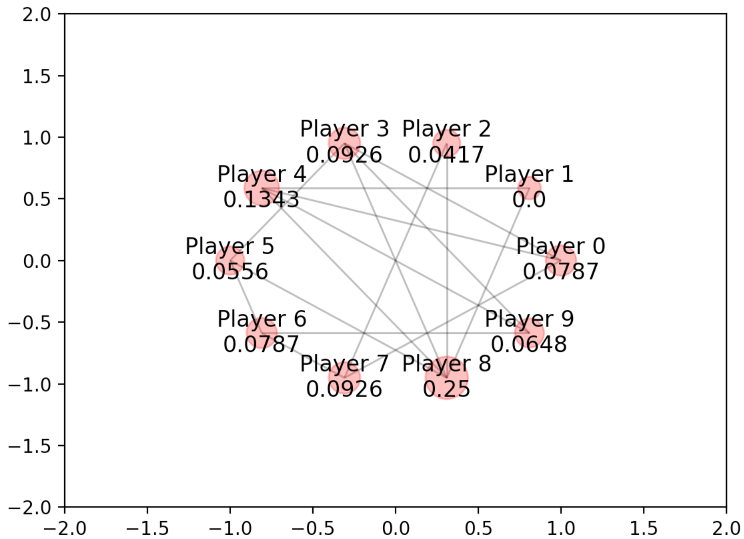


Figure 2: $\epsilon = .2$, $\delta = .05$ all 10 players active

Given the unequal result in 2, we investigate the distribution of betweenness centrality in networks formed under approximate greedy strategies. To characterize the range of betweenness we expect under such uncertainty, we simulate repeat games with approximate greedy players to observe the mean and variance of their betweenness.

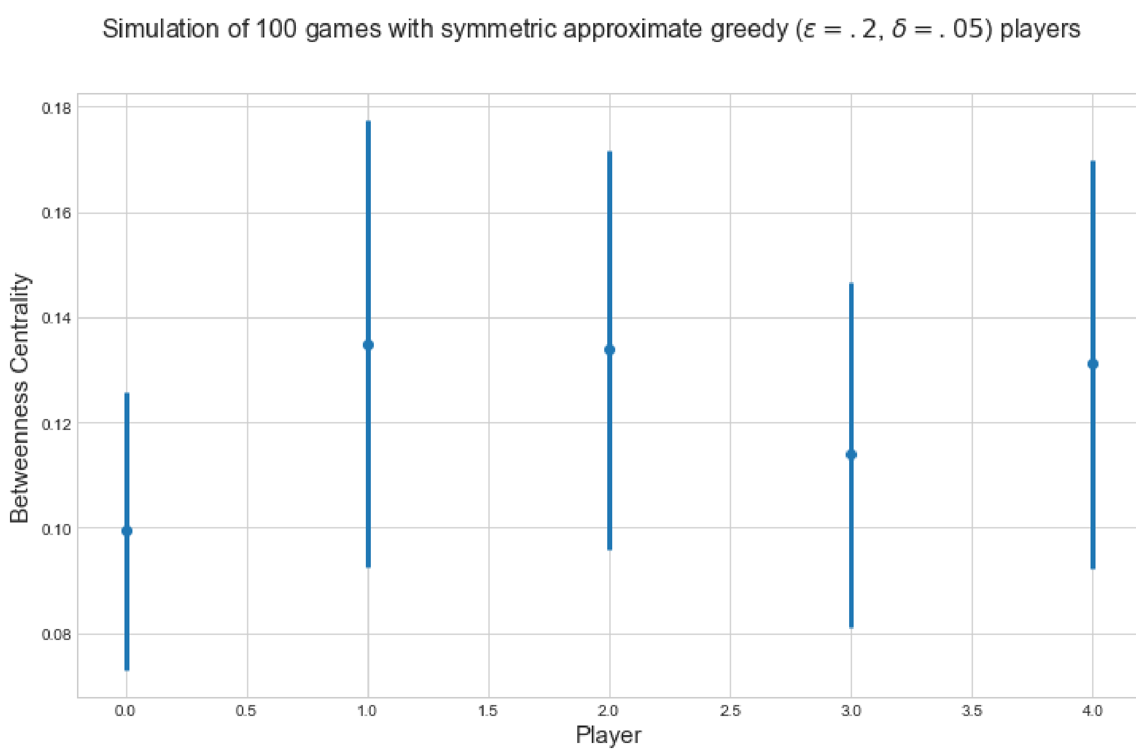


Figure 3: Mean and variance of 100 game simulations

In 3, we can see that there is a moderate amount of variance in the betweenness centrality among agents, but overall the distribution appears roughly uniform. With symmetric approximate agents we don't see extreme outliers like the Medici. An area for further investigation is whether such imbalances arise when agents have differing abilities to accurately gauge their betweenness, for example with vastly different values for ϵ or δ.

8 Learning to Play Optimally

Having characterized game play for agents with sub-optimal heuristcs, we consider the notion of optimal play in the network betweenness game. Moreover, we investigate the following:

1. Can a learning agent can learn to play competitively with an exact greedy agent?
2. Can a learning agent learn a dominant strategy against the exact greedy player?

To explore these questions, we develop a notion of the game as a reinforcement learning problem. That is, the game can easily be re-formulated as the maximization of long-run discounted betweenness centrality. In reinforcement learning, the reward hypothesis states that "All goals can be described by the maximisation of expected cumulative reward" Silver (2017a). The greedy agents are an example of a myopic player with a very low discount rate, who only consider the rewards from the next turn without regard to future steps. That is, the greedy heuristic player's strategy is to always add or remove an edge such that their betweenness centrality is maximized in the next step. A learning agent not only cares about the next step, but also about future discounted rounds. This allows it to consider moves that may not maximize immediate reward but lead to long-term higher rewards (the sum of centrality from each game round). While there have been recent advances in evolutionary strategies such as Salimans et al. (2017), we focus on reinforcement learning for initial investigation because evolution stategies uses black-box optimization and we are potentially interested in learning a value function for the problem.

The reinforcement learning problem consists of an agent that interacts with its environment and observes reward signals. This structure introduces several aspects that are different from traditional supervised learning, such as non *iid* observations and the credit assignment problem (determining how each action in a series of actions contributes to success or to failure). In the case of the network, the agent interacts with its environment by creating an edge, deleting an edge, or doing nothing. Since the number of possible edges it can create or delete increases quadratically with the size of the network, this setting quickly admits a very large action space. Reinforcement learning in such settings with a large number of discrete actions is in fact an area of active research (Dulac-Arnold et al. (2015)). For the sake of rapid exploration, we constrain the networks explored to be smaller (5 to 10 nodes). From observations of the dynamics in larger networks (approximately 500 nodes) tested, we believe that a smaller network does not detract from the quality of the simulations for the interactions of interest.

8.1 Reinforcement Learning Game Specification

Several reward schemes were considered. For example, one reward function was specified as follows to promote actions that increased or maintained high centrality:

- Give the learner +1 reward if it increases its centrality in a given round (or is at 1.0)
- Give learner -1 reward if it decreases its centrality
- Given the learner either 0 or -1 reward otherwise (both were tried)

This seemed to work well but occassionally produced pathological behavior in the form of agents severing their own edges to be able to build their centrality back up again – the discounted future rewards made it worth it to take a short-term penalty. As a result, the we adopted a simpler reward function for the reward r:

$$r_i \leftarrow \text{Agent's centrality at the end of round } i$$

Our experiments were then run using this reward function. The games were each constrained to be 20 rounds. Thus, for each episode of training (one full game of 20 moves), the total reward is the sum of the player's centrality at each step of the game.

$$R_{\text{episode}} = \sum_{i=1}^{20} r_i$$

For our game implementation, we wrote a wrapper around our game class to mimic the OpenAI (2017) API. This helped accelerate model prototyping both by simplifying the interface and allowing code-reuse from the gym ecosystem.

8.2 Policy Gradients

Policy gradient methods seek to maximize the return for a differentiable policy $\pi_\theta(s, a)$. The REINFORCE algorithm uses Monte Carlo sampling of rewards to perform stochastic gradient ascent on a policy's parameters θ.

$$\Delta\theta_t = \alpha \nabla_\theta \log \pi_\theta(s_t, a_t) v_t$$

where v_t is an unbiased estimator of $Q^{\pi_\theta}(s_t, a_t)$ (thus the Monte-Carlo update). The REINFORCE algorithm works as follows (Silver (2017b)):

Algorithm 1: REINFORCE: Monte-Carlo Policy Gradient (Silver (2017b))

Initialize θ arbitrarily
for each episode $\{s_1, a_1, r_2, ..., s_{T-1}, a_{T-1}, r_T\} \sim \pi_\theta$ **do**
  for $t = 1$ to $T-1$ **do**
    $\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(s_t, a_t) v_t$
  end for
end for
return θ

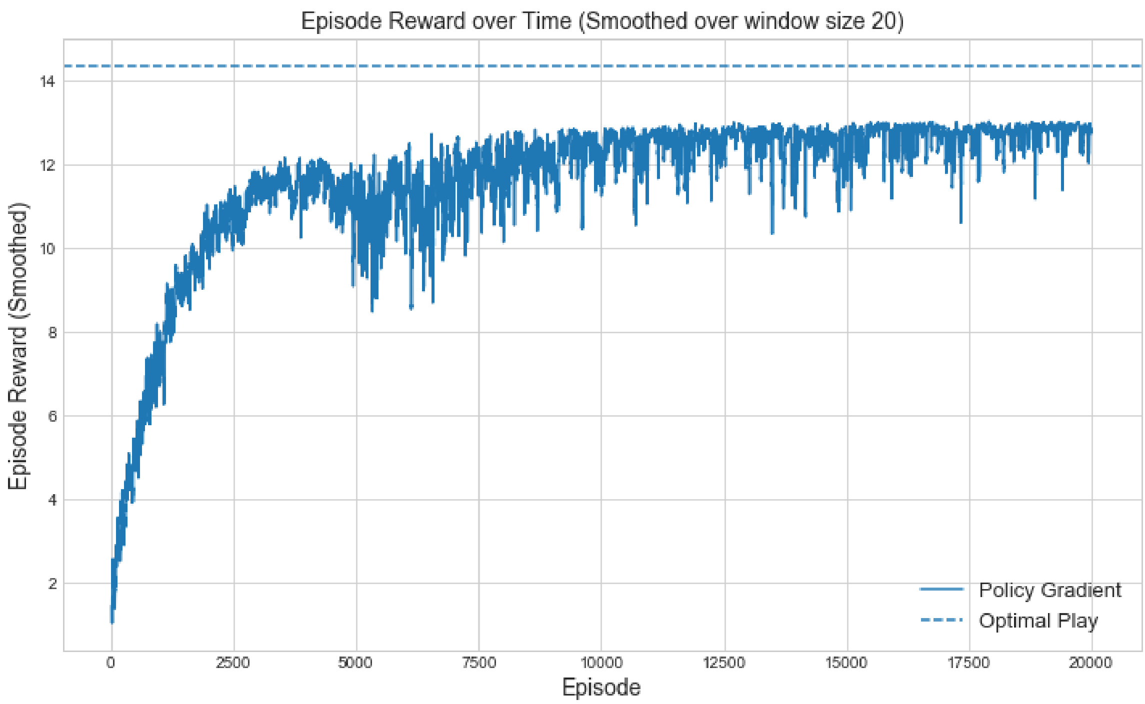


Figure 4: Policy gradient model learns a nearly optimal policy in solo play

First, we considered a policy-gradient method using the REINFORCE algorithm as implemented by Britz (2017). As a baseline, the algorithm was run with a single active node in a graph. We expect that it should learn to form the star network. A solo greedy player with perfect play achieves a total reward ~ 14 across a 20 round game, as seen by the dashed line. The smoothed blue line for the policy gradient indicates the total reward R_{episode} for each game of 20 rounds. The model was trained for 20,000 episodes (games). The linear policy gradient estimator approaches within 1 of this upper bound, achieving near optimality on solo play.

8.3 Greedy Player Duels

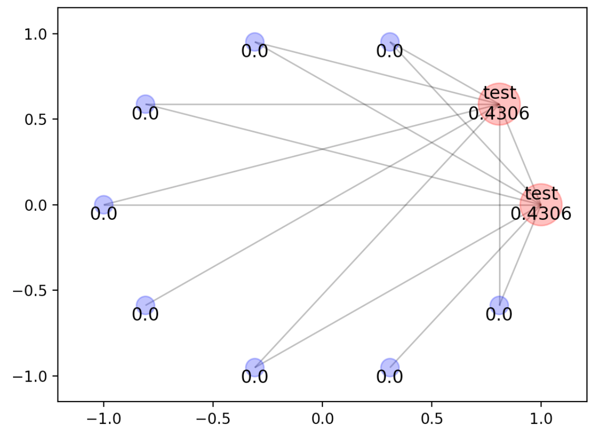


Figure 5: Two greedy players duel to a stalemate

We next turn to consider head-to-head matches (two active players). As a benchmark,we consider the duel between two greedy strategy players. The two players create all of the connections they can to inactive nodes and connect to each other. After that,they enter a cycle of deleting each other's edges and adding their edges back. Because they are symmetric, they end up with identical betweenness centrality. Lastly, we can assess the band of rewards that the greedy players get in this duel (it is a band rather than a line because the reward depends on whether they are in the edge added or deleted phase of the cycle).

8.4 Policy Gradient - REINFORCE vs. Greedy Heuristic

A policy gradient learning agent was trained using REINFORCE for 2000 episodes (games) against a greedy opponent.

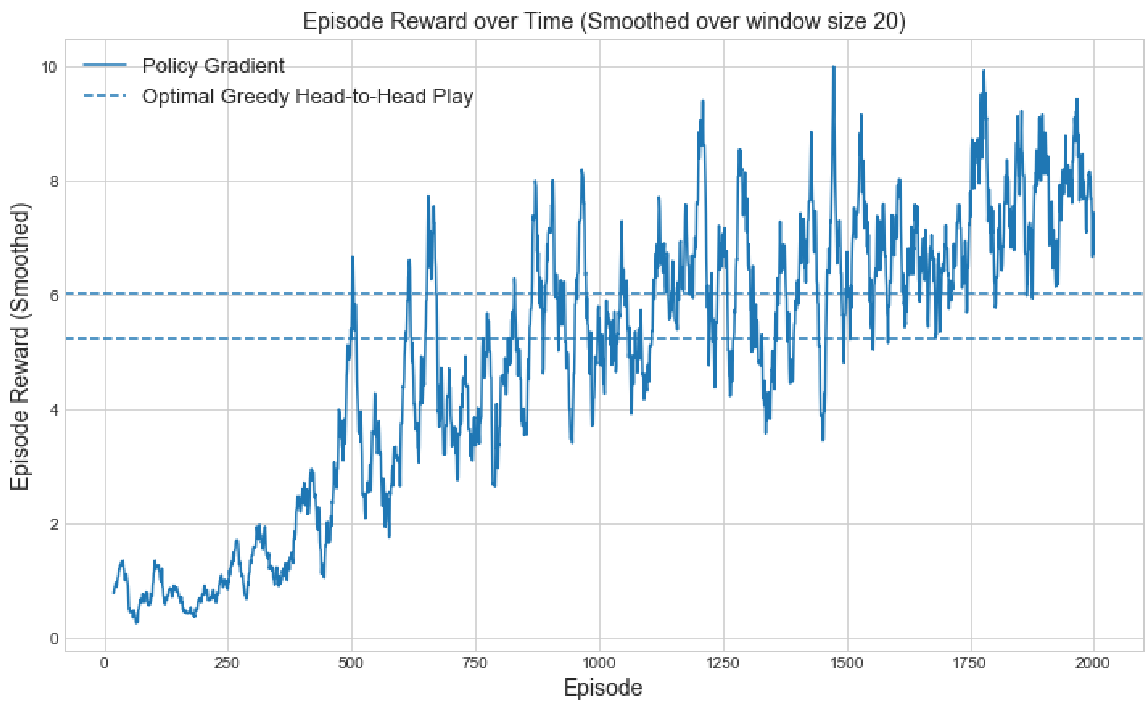


Figure 6: Policy gradient learns a dominant strategy over a greedy player in a duel ($V = 5$)

As we shown in 6, the learning agent not only learns to match the greedy player, but surpasses the greedy strategy in head-to-head play. In our observations, these improvements over the greedy strategy were due to situations in which the agent took forward-looking actions by creating edges between other nodes to short-circuit the betweenness of its greedy rival and force it to delete those edges instead of deleting valuable edges for the learning agent.

8.5 Actor-Critic vs. Greedy Heuristic

In additon to the policy gradient agent, we also consider an actor-critic agent (implementation from Britz (2017)). Rather than use the Monte Carlo estimate of the action-value function as the policy gradient does, actor-critic methods use another function (the critic) to estimate the value of a given state to help find a policy. Rather than use v as an unbiased estimate for $Q^{\pi_\theta}(s_t, a_t)$, actor-critics use $Q_w(s, a)$ to represent the estimate of the action-value function. This changes the policy update for θ to the following:

$$\Delta\theta_t = \alpha \nabla_\theta \log \pi_\theta(s_t, a_t) Q_w(s, a)$$

The critic guides updates to the parameters of the policy, which helps to reduce variance (at the cost of increased bias). The critic can update its parameter w using a range of methods, such as linear TD(0) (Silver (2017b)).

Algorithm 2: Actor Critic (Silver (2017b))

Critic: linear TD(0)
Actor: policy gradient
Initialize s, θ
Sample $a \sim \pi_\theta$
for each episode step **do**
  Sample reward $r = R_s^a$
  Sample action $a' \sim \pi_\theta(s', a')$
  $\delta = r + \gamma Q_w(s', a') - Q_w(s, a)$
  $\theta = \theta + \alpha \nabla_\theta \log \pi_\theta(s_t, a_t) Q_w(s, a)$
  $w \leftarrow w + \beta\gamma\phi(s, a)$
  $a \leftarrow a', s \leftarrow s'$
end for

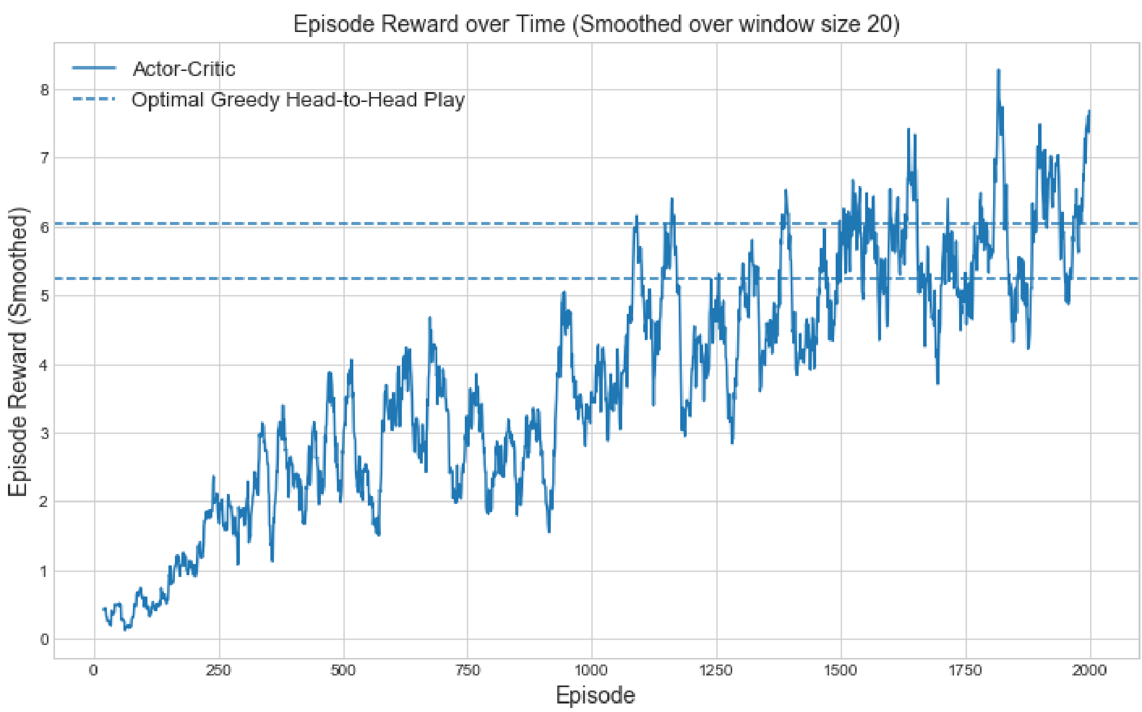


Figure 7: Actor-Critic learns a dominant policy over a greedy player in a duel ($V = 5$)

We train the actor-critic model under identical conditions as the policy gradient model. The actor-critic, using a simple linear policy and a linear TD(0) critic, is also able to match and exceed the greedy strategy using similar techniques as the policy gradient. The advantage of the actor-critic is that the rewards and training are stabilized compared to REINFORCE.

9 Discussion

In this section, we have described approximate betweenness centrality as a means of representing information assymetries and imperfect information for greedy actors. We have also described empirical results for edge formation and centrality dynamics with a learning agent and a greedy strategy, demonstrating that an agent trained with a policy gradient or an actor-critic model is able to match and even exceed a myopic greedy strategy.

In the future, we hope to explore multi-agent extensions of the learning framework to observe if the multi-agent setting yields different dynamics from the greedy dynamic. Additionally, other reward mechanisms have been built into the network environment framework, making extensions to other goals trivial (such as clustering, degree centrality, or eigenvector centrality). Hopefully, future works can also incorporate more recent advances in reinforcement learning, such as TRPO and A3C.

Lastly, it may be the case that betweenness centrality is simply not the right utility function to consider when investigating how players gain influence in networks. It may prove insightful to use empirical data along with inverse reinforcement learning to learn a utility function that can then be used for simulations.

References

Britz, D. (2017). dennybritz/reinforcement-learning: Implementation of reinforcement learning algorithms. python, openai gym, tensorflow. exercises and solutions to accompany sutton's book and david silver's course.

Dulac-Arnold, G., Evans, R., Sunehag, P., and Coppin, B. (2015). Reinforcement learning in large discrete action spaces. *CoRR*, abs/1512.07679.

OpenAI (2017). Openai gym. https://gym.openai.com/. (Accessed on 04/25/2017).

Riondato, M. and Kornaropoulos, E. M. (2016). Fast approximation of betweenness centrality through sampling. *Data Min. Knowl. Discov.*, 30(2):438–475.

Salimans, T., Ho, J., Chen, X., and Sutskever, I. (2017). Evolution strategies as a scalable alternative to reinforcement learning. *CoRR*, abs/1703.03864.

Silver, D. (2017a). Lecture 1: Introduction to reinforcement learning. http://www0.cs.ucl.ac.uk/staff/D.Silver/web/Teaching_files/intro_RL.pdf. (Accessed on 04/25/2017).

Silver, D. (2017b). Lecture 7: Policy gradient.