Dynamic computation of network statistics via updating schema

Jie Sun,[1,*] James P. Bagrow,[2,3,†] Erik M. Bollt,[1,2,‡] and Joseph D. Skufca[1,§]
[1]*Department of Mathematics & Computer Science, Clarkson University, Potsdam, New York 13699-5815, USA*
[2]*Department of Physics, Clarkson University, Potsdam, New York 13699-5820, USA*
[3]*Center for Complex Network Research and Department of Physics, Northeastern University, Boston, Massachusetts 02115, USA*
(Received 26 September 2008; revised manuscript received 9 December 2008; published 30 March 2009)

Given a large network, computing statistics such as clustering coefficient, or modularity, is costly for large networks. When one more edge or vertex is added, traditional methods require that the full (expensive) computation be redone on this slightly modified graph. Alternatively, we introduce here a new approach: under modification to the graph, we update the statistics instead of computing them from scratch. In this paper we provide update schemes for a number of popular statistics, to include degree distribution, clustering coefficient, assortativity, and modularity. Our primary aim is to reduce the computational complexity needed to track the evolving behavior of large networks. As an important consequence, this approach provides efficient methods which may support modeling the evolution of dynamic networks to identify and understand critical transitions. Using the updating scheme, the network statistics can be computed much faster than re-calculating each time that the network evolves. We also note that the update formula can be used to determine which edge or node will lead to the extremal change of network statistics, providing a way of predicting or designing network evolution rules that would optimize some chosen statistic. We present our evolution methods in terms of a network statistics differential notation.

DOI: 10.1103/PhysRevE.79.036116 PACS number(s): 89.75.Fb, 89.75.Hc, 02.70.−c, 05.10.−a

I. INTRODUCTION

Complex networks are useful tools for modeling complicated real life objects and their interactions. Examples include computer networks, social networks, biological networks, etc. [1–6]. Different from the traditional graph theory approach, which emphasizes microstate quantities of each node in the network, recently developed statistical methods [3] allow us to analyze large networks by several important macroscopic statistics meant to summarize the massive amount of information required to completely describe the network. These statistics include such things as degree (number of connections each node has), clustering coefficient [1], assortativity coefficient [7], modularity measure [8], etc. Fast algorithms [9,10] have been developed to compute these statistics for any given network, either represented by adjacency matrix or edge list [11].

Despite that many real world networks evolve, and even grow in time, it is only recently that network models have allowed for this ubiquitous feature. Understanding the evolution of such networks invariably leads quickly to computing network statistics as they evolve in time [12,13]. In this regard, for any evolving network, to measure the corresponding evolution of network statistics, the computation based on static network structure must be done (for the network) at each time step, resulting in a costly (and perhaps, impractical) computation. A missing part in the study of evolving networks is the development of a dynamic algorithm which updates the statistics rather than recomputing the quantity from scratch.

*sunj@clarkson.edu
†j.bagrow@neu.edu
‡bolltem@clarkson.edu
§jskufca@clarkson.edu

In this paper we present an update algorithm based on the knowledge of existing network structure and the changes to the network. Our methods allow us to update relevant statistics of a large network by considering only the adjustments of the network. This philosophy represents a significant theoretical advancement in understanding large scale networks (critical quantities and topological features), and it also leads to efficient algorithms for tracking evolution of large scaled networks in time.

The rest of the paper is organized as follows: In Sec. II, we review the definition of some network statistics and introduce notation that will be used in the paper for these statistics. In Sec. III, we derive update formulas for network statistics upon the change of network structure and compare the computational complexity to the use of standard methods. In Sec. IV, we show examples of application of the update scheme. In Sec. V, we discuss the main results of the paper and give some overview of potential future research. In the Appendix, we provide a simplified, algorithmic representation of our update scheme.

II. DEFINITION AND NOTATION

We represent a network using a graph $G=(V,E)$ where $V=\{1,2,\ldots,N\}$ is the vertex set and $E=\{(i,j)|i \text{ and } j \text{ are connected}\}$ is the edge set. Note that for undirected graphs, if $(i,j)\in E$ then so is (j,i) [18]. In this work, we limit ourselves to undirected, unweighted networks; their graphs possess a symmetric, binary *adjacency matrix* A:

$$a_{ij}=\begin{cases}1, & \text{if } (i,j)\in E;\\ 0, & \text{otherwise.}\end{cases} \tag{1}$$

Let M denote the total number of edges in G. Then

$$M \equiv \frac{1}{2}|E| = \frac{1}{2}\|A\|_F^2 = \frac{1}{2}\sum_{i,j} a_{ij}, \tag{2}$$

where $|\cdot|$ is the cardinality of a set.

Define the neighborhood $N(i)$ of node i as the set of vertices that are adjacent to i, i.e.,

$$N(i) \equiv \{j|(i,j) \in E\} = \{j|a_{ij} = 1\}. \tag{3}$$

Likewise, define the *shared* neighborhood N_{ij} of nodes i and j as

$$N_{ij} \equiv N(i) \cap N(j). \tag{4}$$

The *degree* k_i of node i is the number of nodes it connects to

$$k_i \equiv |N(i)| = \sum_j a_{ij} = \sum_j a_{ji}, \tag{5}$$

since we limit ourselves to undirected networks.

The *clustering coefficient* of node i is defined by [1]

$$C_i \equiv \begin{cases} \dfrac{2\Delta_i}{k_i(k_i-1)}, & \text{if } k_i \geqslant 2; \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

where $\triangle_i$ is the number of triangles that contain i. Then the *average clustering coefficient* [19] of the whole network is simply the average of all C_i's:

$$C \equiv \frac{1}{N}\sum_i C_i. \tag{7}$$

The *assortativity coefficient* r, which describes the correlation of the degree of adjacent nodes of a network [7], is computed as:

$$r \equiv \frac{8M\sum_{(i,j)\in E} k_i k_j - \left[\sum_{(i,j)\in E}(k_i+k_j)\right]^2}{4M\sum_{(i,j)\in E}(k_i^2+k_j^2) - \left[\sum_{(i,j)\in E}(k_i+k_j)\right]^2} = \frac{8Mu - v^2}{4Mw - v^2}, \tag{8}$$

where

$$u \equiv \sum_{(i,j)\in E} k_i k_j, \tag{9}$$

$$v \equiv \sum_{(i,j)\in E}(k_i + k_j), \tag{10}$$

$$w \equiv \sum_{(i,j)\in E}(k_i^2 + k_j^2). \tag{11}$$

Modularity Q [8] measures the quality of a community partition and is typically defined as

$$Q \equiv \frac{1}{2M}\sum_{i,j}\left(a_{ij} - \frac{k_i k_j}{2M}\right)\delta(g_i,g_j) = \frac{1}{2M}\left[S_A - \frac{1}{2M}S_P\right], \tag{12}$$

where $\delta(g_i,g_j)=1$ if nodes i and j are in the same group and zero otherwise, and

$$S_A \equiv \sum_{i,j} a_{ij}\delta(g_i,g_j), \quad S_P \equiv \sum_{i,j} k_i k_j \delta(g_i,g_j). \tag{13}$$

III. UPDATING SCHEMES FOR STATISTICS UPON LOCAL INFORMATION

A. Connecting a new node

The operation of adding an edge to a new node can be decomposed into two successive operations: first, introduce an isolated node that connects to nothing in the network; then add an edge between this node and a previously existing node. In this subsection we discuss the effect of adding an isolated node, and leave the discussion of adding an edge to the next subsection. We use $\tilde{\cdot}$ to represent updated statistics. (Note: this section provides derivation of update formulas. For algorithmic representation of this scheme, the reader should look to the Appendix.)

Since no new edge is introduced, it is easy to obtain the following updating relations:

$$\tilde{N} = N+1, \quad \tilde{M} = M, \quad \tilde{E} = E, \tag{14}$$

and

$$\tilde{a}_{ij} = \begin{cases} a_{ij}, & \text{if } i \neq N+1 \text{ and } j \neq N+1; \\ 0, & \text{otherwise}. \end{cases} \tag{15}$$

Then for other statistics, we have

$$\tilde{k}_i = k_i, \quad i \neq N+1;$$

$$\tilde{k}_{N+1} = 0; \tag{16}$$

and

$$\tilde{C}_i = C_i, \quad i \neq N+1;$$

$$\tilde{C}_{N+1} = 0, \tag{17}$$

so that

$$\tilde{C} = \frac{1}{\tilde{N}}\sum_i \tilde{C}_i = \frac{1}{N+1}\sum_i C_i = \frac{N}{N+1}C. \tag{18}$$

Similarly, $\tilde{r}=r$ since $\tilde{u}=u$, $\tilde{v}=v$, and $\tilde{w}=w$; and $\tilde{Q}=Q$ since $\tilde{S_A}=S_A$, and $\tilde{S_P}=S_P$.

B. Adding an edge between existing nodes

Suppose $a_{pq}=0$ ($p \neq q$ and p, q are not connected), we analyze the impact of connecting p and q on the various statistics of the network; see Fig. 1 for an example.

The goal is to derive computations that are as inexpensive as possible. We use $\tilde{\cdot}$ to represent updated statistics:

$$\tilde{E} = E \cup \{(p,q),(q,p)\}, \tag{19}$$

$$\tilde{M} = M + \Delta^+ M = M + 1, \tag{20}$$

and

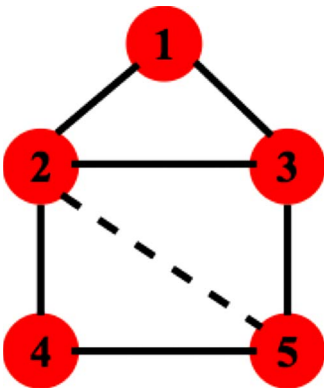

FIG. 1. (Color online) Schematic addition of an edge. The original graph consists of $N=5$ nodes (solid circles) and $M=6$ edges (solid lines). We have, for the original graph, the degree vector $k=[2,3,3,2,2]$, the triangle vector $\triangle=[1,1,1,0,0]$, the clustering coefficient vector $C=[1,\frac{1}{3},\frac{1}{3},0,0]$, and the average clustering coefficient is $\frac{1}{3}$. After an edge is added between node 2 and node 5 (dashed line), the above statistics change to $\tilde{k}=[2,4,3,2,3]$, $\tilde{\triangle}=[1,3,2,1,2]$, $\tilde{C}=[1,\frac{1}{2},\frac{2}{3},1,\frac{2}{3}]$, and the new average clustering coefficient becomes $\frac{23}{30}$.

$$\tilde{a}_{ij} = a_{ij} + \Delta^+ a_{ij} = a_{ij} + \delta_{ip}\delta_{jq} + \delta_{iq}\delta_{jp}, \tag{21}$$

where we use *update delta* Δ^+ to represent the change for statistics upon adding an edge to the existing network, and in the following Δ^- will be used to denote change for statistics upon deleting an existing edge. We will not explicitly specify which edge to add or delete in the update delta notation when there is no confusion.

Based on the above formulas, we can derive schemes for efficiently updating network statistics.

1. Degree

The change in degree for node i is simply

$$\tilde{k}_i = k_i + \Delta^+ k_i = k_i + \delta_{ip} + \delta_{iq}, \tag{22}$$

where

$$\Delta^+ k_i = \delta_{ip} + \delta_{iq}. \tag{23}$$

The above formula indicates that the degree changes only for vertex p and q, so that if one keeps a list of the degree of all vertices of the network, each update takes only two operations when a new edge is added.

2. Clustering coefficient

To compute the new clustering coefficient of each node, and thus the whole network, we need the updated number of triangles at node i:

$$\tilde{\triangle}_i = \begin{cases} \triangle_i, & \text{if } i \notin \{p,q\} \cup N_{pq}; \\ \triangle_i + 1, & \text{if } i \in N_{pq}; \\ \triangle_i + |N_{pq}|, & \text{if } i \in \{p,q\}. \end{cases} \tag{24}$$

Combining this with Eq. (22) and $\triangle_i = \frac{1}{2} C_i k_i (k_i - 1)$, from Eq. (6), we have

$$\tilde{C}_i = \begin{cases} C_i, & \text{if } i \notin \{p,q\} \cup N_{pq}; \\ C_i + \dfrac{2}{k_i(k_i-1)}, & \text{if } i \in N_{pq}; \\ \dfrac{k_i - 1}{k_i + 1} C_i + \dfrac{2|N_{pq}|}{k_i(k_i+1)}, & \text{if } i \in \{p,q\}. \end{cases} \tag{25}$$

Note that whenever the denominator of a fraction is zero, we define the fraction to be zero, in Eq. (25) and throughout. This maintains the consistency that $C_i=0$ if $k_i<2$. Finally, the average clustering coefficient C becomes $\tilde{C}=C+\Delta^+ C$ where

$$\Delta^+ C = \frac{2}{N}\left[\sum_{i \in N_{pq}} \frac{1}{k_i(k_i-1)} + \sum_{i \in \{p,q\}} \left(\frac{|N_{pq}|}{k_i(k_i+1)} - \frac{C_i}{k_i+1} \right) \right]. \tag{26}$$

Note that to update the average clustering coefficient, we need to keep the clustering coefficient for each node in order to apply the update formula, which implies an $O(N)$ storage complexity.

3. Assortativity coefficient

To compute $\tilde{r}$, we need $\tilde{u}$, $\tilde{v}$, and $\tilde{w}$. The update formula for u is

$$\begin{aligned} \tilde{u} &= \sum_{(i,j) \in \tilde{E}} \tilde{k}_i \tilde{k}_j = \sum_{(i,j) \in E} \tilde{k}_i \tilde{k}_j + 2(k_p+1)(k_q+1) \\ &= \sum_{(i,j) \in \hat{E}} k_i k_j + 2 \sum_{i \in N(p)} k_i (k_p+1) \\ &\quad + 2 \sum_{i \in N(q)} k_i (k_q+1) + 2(k_p+1)(k_q+1) \\ &= u + 2\left(\sum_{i \in N(p)} k_i + \sum_{i \in N(q)} k_i \right) + 2(k_p+1)(k_q+1) \\ &= u + \Delta^+ u. \end{aligned} \tag{27}$$

Here $\hat{E}=E\backslash\{(p,q),(q,p)\}$ is the edge set that contains all edges in E but (p,q) and (q,p) and

$$\Delta^+ u = 2\left(\sum_{i \in N(p)} k_i + \sum_{i \in N(q)} k_i \right) + 2(k_p+1)(k_q+1). \tag{28}$$

Similarly, we can obtain update the formula for v and w:

$$\tilde{v} = \sum_{(i,j) \in \tilde{E}} (\tilde{k}_i + \tilde{k}_j) = v + 4(k_p + k_q + 1) = v + \Delta^+ v, \tag{29}$$

where

$$\Delta^+ v = 4(k_p + k_q + 1). \tag{30}$$

For w we have

$$\tilde{w} = \sum_{(i,j) \in \tilde{E}} (\tilde{k}_i^2 + \tilde{k}_j^2) = w + \Delta^+ w, \tag{31}$$

where

$$\Delta^+ w = 6[k_p(k_p+1) + k_q(k_q+1)] + 4. \tag{32}$$

Finally, the new assortativity coefficient can be updated using

$$\tilde{r} = r + \Delta^+ r = \frac{8\tilde{M}\tilde{u} - \tilde{v}^2}{4\tilde{M}\tilde{w} - \tilde{v}^2} = \frac{8(M+1)(u+\Delta^+ u) - (v+\Delta^+ v)^2}{4(M+1)(w+\Delta^+ w) - (v+\Delta^+ v)^2}. \tag{33}$$

4. Modularity

For modularity, we assume that after connecting the nodes p and q, the partitions g_i do not change for any node i. Then the new modularity measure will be

$$\tilde{Q} = \frac{1}{2\tilde{M}}\left[\tilde{S}_A - \frac{1}{2\tilde{M}}\tilde{S}_P\right]. \tag{34}$$

We already have $\tilde{M} = M+1$; we now derive updating formulas for S_A and S_P. By Eq. (13), we have

$$\begin{aligned}\tilde{S}_A &= S_A + \Delta^+ S_A = \sum_{i,j} \tilde{a}_{ij}\delta(g_i,g_j) \\ &= \sum_{i,j} (a_{ij} + \delta_{ip}\delta_{jq} + \delta_{iq}\delta_{jp})\delta(g_i,g_j) = S_A + 2\delta(g_p,g_q),\end{aligned} \tag{35}$$

where $\Delta^+ S_A = 2\delta(g_p,g_q)$; and

$$\begin{aligned}\tilde{S}_P &= S_P + \Delta^+ S_P = \sum_{i,j} \tilde{k}_i\tilde{k}_j\delta(g_i,g_j) \\ &= \sum_{i,j} (k_i + \delta_{ip} + \delta_{iq})(k_j + \delta_{jp} + \delta_{jq})\delta(g_i,g_j) \\ &= S_P + 2\sum_i k_i[\delta(g_i,g_p) + \delta(g_i,g_q)] \\ &\quad + 2[\delta(g_p,g_q) + 1].\end{aligned} \tag{36}$$

However, computing the sum in Eq. (36) for every update is expensive. To avoid this, define the following auxiliary statistics:

$$K_g \equiv \sum_i k_i\delta(g_i,g) \tag{37}$$

with updating scheme

$$\tilde{K}_g = K_g + \Delta^+ K_g = K_g + \delta(g_p,g) + \delta(g_q,g) \tag{38}$$

giving

$$\tilde{S}_P = S_P + \Delta^+ S_P = S_P + 2(K_{g_p} + K_{g_q}) + 2[\delta(g_p,g_q) + 1], \tag{39}$$

where $\Delta^+ S_P = 2(K_{g_p} + K_{g_q}) + 2[\delta(g_p,g_q) + 1]$.

Finally, combining Eqs. (35) and (39) with Eq. (34) gives the updating scheme for Q:

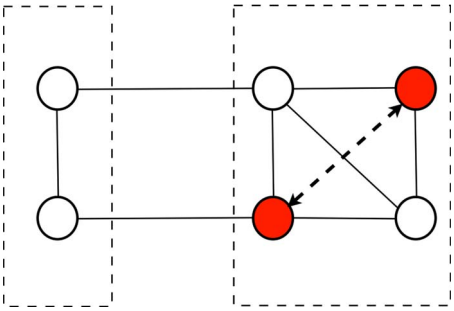

FIG. 2. (Color online) An example that the modularity actually decreases when a new edge is added to vertices within the same group. The dashed oval boxes indicate the preexisting partition of the graph into two groups. Solid lines are the edges in the original graph. Before adding the new edge (dashed arrow line), the modularity is 0.125. After a new edge is added between two vertices in the same group (solid circles) the updated modularity becomes 0.1235.

$$\begin{aligned}\tilde{Q} = Q + \Delta^+ Q = \frac{1}{2(M+1)}\Bigg[&S_A + 2\delta(g_p,g_q) \\ &- \frac{1}{2(M+1)}(S_p + 2[K_{g_p} + K_{g_q}] + 2[\delta(g_p,g_q) + 1])\Bigg].\end{aligned} \tag{40}$$

From Eq. (40) one is able to predict whether the modularity measure Q increases or decreases with the knowledge of existing partition of the graph as well as the edge to be added. For example, if there is a preexisting partition of the graph into two groups, then if a new edge is added in between the two groups, then $\Delta^+ Q < 0$, i.e., the modularity will decrease. On the other hand, if a new edge is added to vertices belonging to the same group, then the modularity increases if the edge is added to the group with smaller total degree; however, adding an edge within a group does not necessarily increase Q if the edge is added into a group with larger total degree; see Fig. 2 as an example.

C. Deleting an existing edge

Now we investigate how network statistics changes when we delete an existing edge in the network; see Fig. 3 for an example.

Suppose $a_{pq} = 1$ ($p \neq q$ and p,q are connected), and we delete this edge, $(p,q) \cup (q,p)$, from our edge set E. Using $\hat{A}$ to represent the updated adjacency matrix, and similarly for other statistics. Then we immediately have

$$\hat{E} = E \setminus \{(p,q),(q,p)\}, \tag{41}$$

$$\hat{M} = M - 1, \tag{42}$$

and

$$\hat{a}_{ij} = a_{ij} + \Delta^- a_{ij} = a_{ij} - \delta_{ip}\delta_{jq} - \delta_{iq}\delta_{jp}. \tag{43}$$

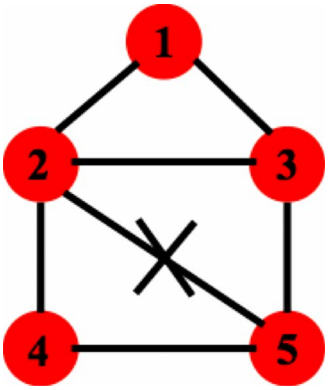

FIG. 3. (Color online) Schematic removal of an edge. The original graph consists of $N=5$ nodes (solid circles) and $M=7$ edges (solid lines), with statistics $k=[2,4,3,2,3]$, $\triangle=[1,3,2,1,2]$, $C=[1,\frac{1}{2},\frac{2}{3},1,\frac{2}{3}]$, and average clustering coefficient $\frac{23}{30}$. After the edge between node 2 and node 5 is deleted, the new statistics are $\hat{k}=[2,3,3,2,2]$, $\hat{\triangle}=[1,1,1,0,0]$ $\hat{C}=[1,\frac{1}{3},\frac{1}{3},0,0]$, and the new average clustering coefficient is $\frac{1}{3}$. Note that this is a symmetric operation to the one shown in Fig. 1.

1. Degree

The change in degree for node i is $\hat{k}_i=k_i+\Delta^- k_i$ where

$$\Delta^- k_i = -\delta_{ip} - \delta_{iq}. \tag{44}$$

2. Clustering coefficient

For the new clustering coefficient, we first obtain the formula for updating the number of triangles containing node i:

$$\hat{\triangle}_i = \begin{cases} \triangle_i, & \text{if } i \notin \{p,q\} \cup N_{pq}; \\ \triangle_i - 1, & \text{if } i \in N_{pq}; \\ \triangle_i - |N_{pq}|, & \text{if } i \in \{p,q\}. \end{cases} \tag{45}$$

Then we obtain the formula for updating C_i:

$$\hat{C}_i = \begin{cases} C_i, & \text{if } i \notin \{p,q\} \cup N_{pq}; \\ C_i - \dfrac{2}{k_i(k_i-1)}, & \text{if } i \in N_{pq}; \\ \dfrac{k_i}{k_i-2} C_i - \dfrac{2|N_{pq}|}{(k_i-1)(k_i-2)}, & \text{if } i \in \{p,q\}. \end{cases} \tag{46}$$

The average clustering coefficient C is updated by $\hat{C}=C+\Delta^- C$ where

$$\Delta^- C = -\frac{2}{N}\left[\sum_{i\in N_{pq}} \frac{1}{k_i(k_i-1)} + \sum_{i\in\{p,q\}} \left(\frac{|N_{pq}|}{(k_i-1)(k_i-2)} - \frac{C_i}{k_i-2}\right)\right]. \tag{47}$$

3. Assortativity coefficient

The updating formulas for u, v, w are $\hat{u}=u+\Delta^- u$, $\hat{v}=v+\Delta^- v$, $\hat{w}=w+\Delta^- w$, where

TABLE I. Comparison of computational complexity.

Statistics	Adjacency matrix	Edge list	Updating scheme
Degree (one node)	$O(N)$	$O(\langle k\rangle)$	$O(1)$
Degree (network)	$O(N^2)$	$O(\langle k\rangle N)$	$O(1)$
Clustering coefficient (one node)	$O(\langle k\rangle N)$	$O(\langle k\rangle^3)$	$O(\langle k\rangle)$
Clustering coefficient (network)	$O(\langle k\rangle N^2)$	$O(\langle k\rangle^3 N)$	$O(\langle k\rangle)$
Assortativity coefficient	$O(N^2)$	$O(\langle k\rangle N)$	$O(\langle k\rangle)$
Modularity measure	$O(N^2)$	$O(\langle k\rangle N)$	$O(1)$

$$\Delta^- u = -2\left(\sum_{i\in N(p)} k_i + \sum_{i\in N(q)} k_i\right) - 2(k_p-1)(k_q-1),$$

$$\Delta^- v = -4(k_p+k_q-1),$$

$$\Delta^- w = -6[k_p(k_p-1)+k_q(k_q-1)] - 4. \tag{48}$$

Then the new assortativity coefficient $\hat{r}$ is given by

$$\hat{r} = \frac{8\hat{M}\hat{u} - \hat{v}^2}{4\hat{M}\hat{w} - \hat{v}^2} = \frac{8(M-1)(u+\Delta^- u) - (v+\Delta^- v)^2}{4(M-1)(w+\Delta^- w) - (v+\Delta^- v)^2}. \tag{49}$$

4. Modularity

For modularity, we again assume that the community partitions g_i are unchanged after disconnecting the edge between p and q. It follows that

$$\hat{S}_A = S_A + \Delta^- S_A = S_A - 2\delta(g_p,g_q), \tag{50}$$

$$\hat{S}_P = S_P + \Delta^- S_P = S_P - 2(K_{g_p} + K_{g_q}) + 2[\delta(g_p,g_q)+1], \tag{51}$$

where K_g is now updated using

$$\hat{K}_g = K_g + \Delta^- K_g = K_g - \delta(g_p,g) - \delta(g_q,g). \tag{52}$$

These now define the updating scheme for $\hat{Q}=(\hat{S}_A - \hat{S}_P/2\hat{M})/2\hat{M}$.

We remark here that the formulas given for the deletion of an edge can also be used for the removal of a node, by noting that the operation of removing a node can be decomposed into removal of edges that node has.

D. On computational complexity

In Table I we compare the computational complexity of using the updating scheme and regular methods. For the update scheme, we assume that the initial statistics are already known, so that the update value listed indicates the computations required to perform a single update to those statistics. For the standard methods, we note that the operations count depends on the data structure used to represent the network. The updating scheme requires $O(1)$ operations to update for sparse graphs and at most $O(\langle k\rangle)$, which has a significant

advantage compared to regular methods when the graph size becomes large.

Our primary focus is developing efficient algorithms for application to problems of dynamic networks, and the computation savings is significant. For example, given a network of N vertices and M edges, if one edge is added to this network, the computation of network statistics need to be remade, using traditional methods (corresponding to columns 2 or 3 in Table I); on the other hand, our update scheme provides an efficient way to update these statistics which requires far less number of operations (corresponding to column 1 in Table I).

One may also consider the process of building a network, which can be viewed simply as an edge-adding algorithm from a starting set of a graph with N nodes and no edges. It takes $\frac{\langle k\rangle N}{2}$ steps to create the network. The update formulas for degree and modularity indicate that computing the entire time sequence of statistics has the same computational complexity as doing the single computation for the final state (using the edge list). The formula for clustering coefficient is more efficient to calculate each value along the way rather than the single computation of the final state, although we also need additional storage to track the clustering coefficient for each node. Computing the entire time vector of assortativity coefficients requires an additional factor $\langle k\rangle$ computations, which is (typically) a minor price.

IV. EXAMPLES OF APPLICATION

In this subsection we show implementation of the above formula to obtain the evolution of some network statistics. We will focus on the case of adding edges between existing nodes, the other two operations will be very similar. The statistics we will calculate are the degree distribution, average clustering coefficient, and modularity measure, although again, the evolution of other statistics can be obtained in the same manner by using the updating scheme. The evolving network models we choose are not intended to mimic real-world nets, but to show the efficiency of the updating scheme.

A. Evolution of degree and clustering coefficient

In Fig. 4 we show the evolution of degree distribution of a typical realization of a growing random graph [14], obtained as follows: start with a random graph of $N=1000$ nodes, with average degree $\langle k\rangle=10$. At each time step, randomly choose two nodes that are not connected, and make an edge between them, until the average degree of the network reaches $\langle\tilde{k}\rangle=20$. The total number of time steps is 5000, which is $O(N)$ in this case. Note that using the updating scheme to obtain the evolution of degree in this case requires $O(N^2)$ (mostly for initial calculation) operations while using regular method would require $O(N^3)$ operations (using the adjacency matrix).

In Fig. 5 we show the evolution of average clustering coefficient of a typical realization of a Barabasi-Albert network [2]. The initial network is a random network with $N=100$ vertices and every pair of vertices is connected with probability $p=0.5$. Then, 5000 new vertices are added in the current network successively. Each time a new vertex is introduced, it connects to two preexisting vertices according to the preferential attachment rule [2], which corresponds to two time steps shown in Fig. 5. The update scheme allows us to efficiently compute the evolution curve shown in Fig. 5, instead of recomputing the average clustering coefficient at each time step, which would have raised the computational requirement by about five orders of magnitude.

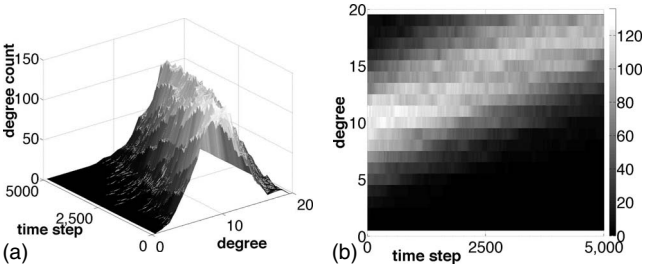

FIG. 4. Evolution of degree distribution of a random growing network. The number of vertices is 1000 in the network. Initially the connection probability of any pair of edges is 0.01, by adding random edges in the network, this probability increases to 0.02 in the end. We show two views of the evolution of the degree distribution as with respect to the process of add successive random edges. In the left panel we see that for any given time, the empirical distribution shows the underlying Poission process, and the peak is moving to larger degree side as time increases. The right panel simply provides an alternate view of the same data, providing clearer visualization of hidden portions of the three-dimensional surface.

B. Evolution of modularity

We start from an initial network with clear partition, constructed as follows: generate an empty graph of N vertices, prescribe a partition of the set $\{1,2,\ldots,N\}$ into two groups

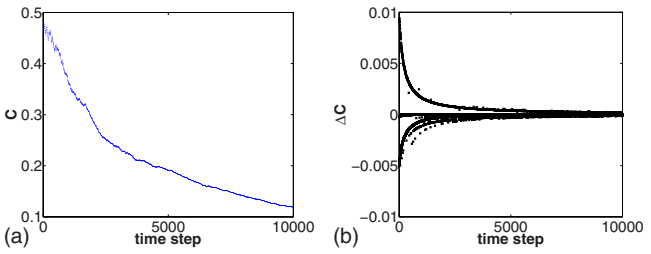

FIG. 5. (Color online) Evolution of the average clustering coefficient C of a growing Barabasi-Albert network. In the left panel we show the evolution of the average clustering coefficient C at each time instance. In the right panel we show the change of average clustering coefficient ΔC. Note that to obtain this curve of the evolution of C, we only need to compute C for the initial network once, which requires $O(\langle k\rangle^3 N)$ operations, and then adopt our update formula, which requires $O(\langle k\rangle)$ operations per step; while if one uses direct computation, it would require $O(\langle k\rangle^3 N)$ to compute each single step, and is highly inefficient. The structure in the right side graph is only observable because we compute the change *at each time step* and would be obscured if one were to compute the change over larger time increments.

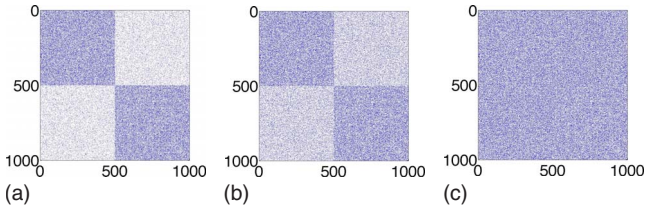

FIG. 6. (Color online) Spy plot at three specific instances for the adjacency matrix. The left panel corresponds to the initial network ($p_1=p_2=0.2$ and $p_{between}=0.05$), where there is a clear community structure. The middle panel corresponds to the time when $p_{between}$ reaches 0.1 where the community structure becomes less apparent. The right panel is the end of the growing process such that $p_{between}=0.2$ and the network is totally random with no community structure.

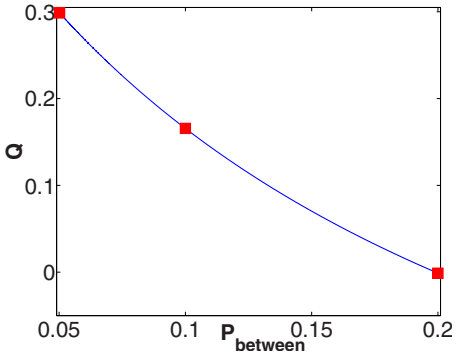

FIG. 7. (Color online) The evolution of modularity Q. Three red squares correspond to the time instances that are shown in Fig. 6

such that the group sizes are N_1, N_2. Randomly connect any pair of vertices in group 1 with probability p_1, and those in group 2 with probability p_2; then randomly connect a vertice in group 1 to a vertice in group 2 with probability $p_{between}$. Probability $p_{between}$ is chosen to be smaller than p_1 and p_2, creating a clear community structure. In our example, we choose $N=1000$, with group 1 composed of nodes $\{1,\ldots,500\}$, with the rest of the nodes forming group 2. We let $p_1=p_2=0.2$ and $p_{between}=0.05$. For the evolutionary process, we add random edges between the groups until the probability of connecting in between groups is the same as the probability of connecting inside the groups (resulting in a completely random network in the end). In Fig. 6 we plot, for three specific time instances, the adjacency matrix of the graph. As more in-between edges are added, the original partition is less valid (the block diagonal structure becomes more vague). Correspondingly, the evolution of modularity measure is shown in Fig. 7.

V. DISCUSSION AND CONCLUSION

In this paper, we derive update formulas for important network statistics (degree, clustering coefficient, assortativity coefficient, modularity) as theoretical and computational tools for analyzing evolving networks. The update formulas are based on singe edge or node updating. An arbitrary change to the graph structured can be viewed as a sequence of these unitary changes, with statistics updated by sequential applications of the formulas we present in this paper. We also show several examples to illustrate the use of the updating scheme, allowing us to efficiently track the evolution of network statistics in situations where traditional methods to compute those statistics would be impractical.

The derivation of the update formula requires that the statistics depend locally on network structure, for example, the update formula for clustering coefficient only requires the knowledge of local information of the vertices that are going to be connected. It becomes very hard, or maybe even impossible, to derive the exact update formula for statistics that depend upon global information of the network, such as the diameter [14] or the Fiedler vector [15] of the network. However, the change of some of these global statistics can be bounded if there is only small change in the graph. For example, the change in the spectra and eigenvectors (including the Fiedler vector) of the graph Laplacian upon adding or deleting a few edges in the graph may be bounded by well-known perturbation results such as those in [15,16]. Our hope is that having introduced the principle, pursuing update schemes for other statistics might provide fruitful research that yields additional tools for analysis of evolving networks.

ACKNOWLEDGMENTS

E.M.B. and J.S. were supported by the Army Research Office under Grant No. 51950-MA. E.M.B. and J.D.S. were supported by the National Science Foundation under Grant No. DMS-0708083. J.P.B. gratefully acknowledges support from the National Science Foundation.

APPENDIX: ALGORITHMS

We give pseudo-code for implementing our update schema upon single change to the network, as described in Sec. III.

Given: A, a symmetric binary matrix corresponding to a simple graph.
Compute initial statistics:
N: number of nodes, M: number of edges;
k: degree vector, C: clustering coefficient vector, $\bar{C}$: average clustering coefficient;
u,v,w,r: assortativity coefficient and related statistics, as described in Sec. II;

S_A, S_P, g, K, Q: modularity measure and related statistics, as described in Sec. II, and in Eq. (37).
if a new node is added to the network without any connection *then*

$C \to [C;0]$, $\bar{C} = \frac{N}{N+1}\bar{C}$;
$A \to [A,0;0]$;
$N \to N+1$;
$k \to [k;0]$.
else if a new edge is added between nodes p and q *then*

---

(1) *Update* C:
Set: $\Delta^+ C = 0$,
Let N_{pq} denote the set of common neighbors of p and q, and
Set: $C_p = C_p - \frac{2}{k_p+1}C_p + \frac{2|N_{pq}|}{k_p(k_p+1)}$, $\Delta^+ C = \Delta^+ C - \frac{2}{k_p+1}C_p + \frac{2|N_{pq}|}{k_p(k_p+1)}$,
Set: $C_q = C_q - \frac{2}{k_q+1}C_q + \frac{2|N_{pq}|}{k_q(k_q+1)}$, $\Delta^+ C = \Delta^+ C - \frac{2}{k_q+1}C_q + \frac{2|N_{pq}|}{k_q(k_q+1)}$,
for every $i \in N_{pq}$ *do*
Set: $C_i = C_i + \frac{2}{k_i(k_i-1)}$ $\Delta^+ C = \Delta^+ C + \frac{2}{k_i(k_i-1)}$,
end for
Update the average clustering coefficient: $\bar{C} = \bar{C} + \frac{\Delta^+ C}{N}$.

---

(2) *Update* r:
Set: $u = u + 2(\Sigma_{i\in N(p)} k_i + \Sigma_{i\in N(q)} k_i) + 2(k_p+1)(k_q+1)$,
Set: $v = v + 4(k_p + k_q + 1)$,
Set: $w = w + 6[k_p(k_p+1) + k_q(k_q+1)] + 4$,
Update the assortativity coefficient: $r = \frac{8(M+1)u - v^2}{4(M+1)w - v^2}$.

---

(3) *Update* Q:
Set: $S_A = S_A + 2\delta(g_p, g_q)$,
Set: $S_P = S_P + 2(K_{g_p} + K_{g_q}) + 2[\delta(g_p, g_q) + 1]$,
Set: $K_g = K_g + \delta(g_p, g) + \delta(g_q, g)$,
Update the modularity measure: $Q = \frac{1}{2(M+1)}[S_A - \frac{1}{2(M+1)}S_P]$.

---

(4) *Update elementary statistics:*
Set: $A(p,q) = A(q,p) = 1$,
$M \to M+1$,
Set: $k_p = k_p + 1$, $k_q = k_q + 1$.

---

else if an existing edge between nodes p and q is removed *then*

---

(1) *Update* C:
Set: $\Delta^- C = 0$,
Let N_{pq} denote the set of common neighbors of p and q, and
Set: $C_p = C_p + \frac{2}{k_p-2}C_p - \frac{2|N_{pq}|}{(k_p-1)(k_p-2)}$, $\Delta^- C = \Delta^- C + \frac{2}{k_p-2}C_p - \frac{2|N_{pq}|}{(k_p-1)(k_p-2)}$,
Set: $C_q = C_q + \frac{2}{k_q-2}C_q - \frac{2|N_{pq}|}{(k_q-1)(k_q-2)}$, $\Delta^- C = \Delta^- C + \frac{2}{k_q-2}C_q - \frac{2|N_{pq}|}{(k_q-1)(k_q-2)}$,
for every $i \in N_{pq}$ *do*
Set: $C_i = C_i - \frac{2}{k_i(k_i-1)}$ $\Delta^- C = \Delta^- C - \frac{2}{k_i(k_i-1)}$,
end for
Update the average clustering coefficient: $\bar{C} = \bar{C} + \frac{\Delta^- C}{N}$.

---

(2) *Update* r:
Set: $u = u - 2(\Sigma_{i\in N(p)} k_i + \Sigma_{i\in N(q)} k_i) - 2(k_p-1)(k_q-1)$,

Set: $v=v-4(k_p+k_q-1)$,
Set: $w=w-6[k_p(k_p-1)+k_q(k_q-1)]-4$,
Update the assortativity coefficient: $r=\frac{8(M-1)u-v^2}{4(M-1)w-v^2}$.

(3) *Update Q*:
Set: $S_A=S_A-2\delta(g_p,g_q)$,
Set: $S_P=S_P-2(K_{g_p}+K_{g_q})+2[\delta(g_p,g_q)+1]$,
Set: $K_g=K_g-\delta(g_p,g)-\delta(g_q,g)$,
Update the modularity measure: $Q=\frac{1}{2(M-1)}[S_A-\frac{1}{2(M-1)}S_P]$.

(4) *Update elementary statistics:*
Set: $A(p,q)=A(q,p)=0$;
$M\rightarrow M-1$;
Set: $k_p=k_p-1$, $k_q=k_q-1$.
end if

[1] D. J. Watts and S. H. Strogatz, Nature (London) **393**, 440 (1998).
[2] A. L. Barabasi and R. Alberet,Science **286**, 509 (1999).
[3] R. Alberet and A. L. Barabasi, Rev. Mod. Phys. **74**, 47 (2002).
[4] R. Pastor-Satorras, A. Vazquez, and A. Vespignani, Phys. Rev. Lett. **87**, 258701 (2001).
[5] A. Vazquez, R. Pastor-Satorras, and A. Vespignani, Phys. Rev. E **65**, 066130 (2002).
[6] S. Zhou, Phys. Rev. E **74**, 016124 (2006).
[7] M. E. J. Newman, Phys. Rev. Lett. **89**, 208701 (2002).
[8] M. E. J. Newman, Proc. Natl. Acad. Sci. U.S.A. **103**, 8577 (2006).
[9] URL https://networkx.lanl.gov/wiki
[10] URL http://vlado.fmf.uni-lj.si/pub/networks/pajek/
[11] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. (MIT Press, Cambridge MA, 2001).
[12] G. Kossinets and D. J. Watts, Science **311**, 88 (2006).
[13] A. Gautreau, A. Barrat, and M. Barthelemy, e-print arXiv:0811.1051.
[14] B. Bollobas, *Random Graphs*, 2nd ed. (Cambridge University Press, Cambridge, England, 2001).
[15] M. Fiedler, Combinatorics, and Graph Theory **25**, 57 (1989).
[16] J. W. Demmel, *Applied Numerical Algebra* (SIAM, Philadelphia, PA 1997).
[17] M. E. J. Newman, SIAM Rev. **45**, 167 (2003).
[18] For undirected graphs we could have eliminated the element (j,i) in E if (i,j) has been put in E, but including both of them allows easier generalization to directed graphs.
[19] There is an alternative definition of the average clustering coefficient of a network in [17], and the formula for updating this alternative C can be derived easily based on updating the number of triangles and triples in the network.