Group Tracking on Dynamic Networks

James P. Ferry
Metron, Inc.
Reston, VA. U.S.A.
ferry@metsci.com

Abstract – This paper develops a Bayesian method for inferring the evolution of hidden groups from the signatures they leave in dynamic network data. Such methods are well established for detecting groups in static networks. The dynamic generalization is based on a Markov process model for joint group–graph evolution, which is used to produce a sequential Bayesian filter for the probabilities of the group membership hypotheses. This filter is demonstrated in a simple scenario to show how both positive information (changes in network structure) and negative information (periods of no change) may be combined to track group membership optimally.

Keywords: Tracking, group finding, network, dynamic network, Markov process, Bayesian filter.

1 Introduction

One key challenge in network science is processing vast quantities of data into information sufficiently rich and concise to be of value to a human analyst. A particularly useful inference to make from the data is the underlying group structure of the network. Many real-world networks exhibit link patterns suggestive of underlying groups of vertices that influence the network structure. For example, transactional data such as email or phone call records could be used to infer the existence of groups of individuals engaged in organized activity. The groups found in this manner are often known, benign entities (e.g., companies), but after filtering these out, analysis of the entities not previously known can be fruitful. There are various related techniques for finding such groups.

Most methods assume that vertices are partitioned into groups, although there are exceptions such as [1] that allow vertices to belong to more than one group (or to none). A measure of the quality of a partition is then needed to select the best one. Some methods employ variants of Girvan–Newman modularity [2] for this purpose; others such as [3] use physics-based measures. Hastings [4] pursues a more formal approach, defining a model which yields the joint probabilities of all possible group-partition and network states. He then employs Bayesian inference to determine the partition probabilities given the network. A similar approach is presented in Section 2. However, despite the varying approaches to group-finding, the resulting computations required are similar, and the emphasis in the literature is on their optimization.

In this paper, the focus is different. A second key challenge in network science is that network data is dynamic, as are the underlying groups that may be influencing it. When confronted with dynamic network data, the problem is not to detect groups, but to track them. After introducing the static problem in Section 2, an analogous extension to the dynamic case is developed in the Section 3. This extension is based on a Markov process model for the continuous-time evolution of the network and the groups influencing it. The result of a formal analysis is a sequential Bayesian filter that yields the exact evolution of every group membership hypothesis. Section 4 gives explicit examples of the performance of the Bayesian filter for small dynamic networks. Because the space of hypotheses is vast even for small networks, the exact formulation developed here is not yet practical. However, it serves as a principled basis for on-going work on developing efficient approximations.

2 The joint group–graph model

Static inference problems involve two components: a state space and a process that generates evidence based on the hidden state of the system. Let $\mathbb{Z}_n = \{1, 2, \ldots, n\}$ denote a set of $n$ objects, and $\mathbb{Z}_m$ a set of $m$ groups. We let $\phi : \mathbb{Z}_n \rightarrow \mathbb{Z}_m$ denote a group assignment, which partitions the objects into groups—this is the state component of the problem. A group assignment may be expressed as an $n$-tuple of integers in $\mathbb{Z}_m$. The first panel of Figure 1 is an example of an assignment $\phi$ of 6 objects into 3 groups. Using the color scheme green = 1, yellow = 2, and purple = 3, the assignment in Figure 1 may be written $\phi = (1, 3, 3, 2, 3, 2)$, indexing the vertices counterclockwise.

![Figure 1: Example group assignment and graph](image-url)
The set of (unordered, distinct) pairs of elements in $Z_n$ is denoted $Z_n^{(2)}$. We let $\kappa : Z_n^{(2)} \rightarrow Z_r$ denote a graph, which specifies one of $r$ edge types for each edge $(v, w) \in Z_n^{(2)}$—this is the evidence component of the problem. A graph may be expressed as a tuple of integers in $Z_n$, ordering the edges $\{1,2\}$ through $\{n-1,n\}$. The graph $\kappa = (1,3,1,1,1,1,2,1,1,4,1,3,2,1)$ is shown in the second panel of Figure 1, using the color scheme white = 1, black = 2, blue = 3, and red = 4. This definition generalizes the usual $r=2$ definition of a graph (interpreting type 1 as “no-edge”). The last panel shows $\phi$ and $\kappa$ together.

The Bayesian approach to group finding is based on specifying a model for the probability of all pairs $(\phi, \kappa)$. Given such a model, it is simple, in principle, to determine the probability of all possible group assignments $\phi$ given an observed graph $\kappa$. We define the model $H(n, p, Q)$ as follows. An instance $(\phi, \kappa)$ of $H(n, p, Q)$ is an assignment $\phi$ for $n$ vertices and a graph $\kappa$ on these vertices. The parameter $p$ is an $m$-vector specifying the group assignment probabilities. Each vertex is assigned a group from 1 to $m$ independently, the probability of assigning a vertex to group $i$ being $p_i$. The parameter $Q$ is an $m \times m$ array of $r$-vectors $q_{ij}$. The probability of an edge with endpoints in groups $i$ and $j$ being of type $k$ is the $k^{th}$ component of this vector, denoted $q_{ij,k}$. We require $p$ and $q_{ij}$ to be stochastic (i.e., to be non-negative and sum to 1), and, because edges are undirected, we require $q_{ij} = q_{ji}$.

The prior probability $\Pr(\phi)$ of a group assignment $\phi$, and the probability of a graph $\kappa$ given the assignment $\phi$ are

$$\Pr(\phi) = \prod_{v=1}^n p_{e(v)}, \quad \Pr(\kappa | \phi) = \prod_{e \in Z_n^{(2)}} q_{\phi(e_1)\phi(e_2),\kappa(e)},$$

where $e_1$ and $e_2$ are the endpoints of an edge $e$. Bayesian inversion yields the probability of the assignment $\phi$ given the graph $\kappa$:

$$\Pr(\phi | \kappa) = \frac{\Pr(\kappa | \phi) \Pr(\phi)}{\Pr(\kappa)},$$

where $\Pr(\kappa)$ is just a normalizing constant. Thus it is a simple matter, in principle, to compute $\Pr(\phi | \kappa)$ for all $\phi$. The difficult part of group finding is to perform a calculation such as (2) in an efficient manner. The goal of this work, however, is to generalize group finding to group tracking. To do this requires specifying a dynamic process on the state space of the inference problem. This is accomplished by generalizing the static $(\phi, \kappa)$-valued model $H(n, p, Q)$ to a $(\phi, \kappa)$-valued stochastic process.

3 The joint group-graph process

To generalize $H(n, p, Q)$, we replace the vector $p$ with an $m \times m$ matrix $A$, and the vectors $q_{ij}$ with $r \times r$ matrices $B_{ij}$. The $B_{ij}$ matrices are collected in an $m \times m$ array denoted $B$, and the dynamic generalization of $H(n, p, Q)$ is denoted $\mathcal{H}(n, A, B)$. Whereas the vectors $p$ and $q_{ij}$ directly determine probabilities, the matrices $A$ and $B_{ij}$ determine the rates of change of probabilities. That is, they are the transition rate matrices for Markov processes [5].

A Markov process is the continuous-time version of the (more familiar) Markov chain, with an analogous theory and nomenclature. For example, the transition matrices of Markov chains are required to be (left) stochastic: i.e., non-negative with column sums of 1. Analogously, transition rate matrices are required to be quasipositive (i.e., off-diagonal elements must be non-negative) with column sums of 0. The reason for using Markov processes rather than chains is twofold: (1) it is more general because a process can be easily down-sampled to a chain if desired; and (2) the matrices required for inference end up being much sparser, due to the limited number of events that can dominate the dynamics over an infinitesimal time span.

A sample from the process $\mathcal{H}(n, A, B)$ is a time history $\{(\phi_t, \kappa_t) : t \geq 0\}$ of group-assignment / graph pairs, where the initial condition $(\phi_0, \kappa_0)$ is either specified or assumed to be drawn from the stationary distribution of $\mathcal{H}(n, A, B)$. Section 3.1 constructs the Markov process for the group assignment $\{\phi_t : t \geq 0\}$. Section 3.2 constructs the Markov process for the graph $\{\kappa_t : t \geq 0\}$, conditioned on $\{\phi_t : t \geq 0\}$. Section 3.3 combines these results to produce the transition rate matrix for the joint process, and uses this to derive an inference procedure.

3.1 The group assignment process

Just as the group for each vertex is assigned independently in the static case, here the evolution of the group of each vertex is taken to be independent of the other vertices. We use $p(t)$ to denote the vector of probabilities that a vertex belongs to groups 1 through $m$ at time $t$. Although this usage of $p$ is similar to that of the static case, it no longer represents a parameter of the system, but rather a quantity whose evolution is determined by the system via

$$\dot{p} = Ap,$$

where solution is $p(t') = e^{(t'-t)A}p(t)$. Note that $e^{(t'-t)A}$ is the transition matrix for the Markov chain mapping $t$ to $t'$. The next step in the construction is to combine the independent processes on each vertex to form the joint process for all vertices. Independent Markov processes may
be combined into a Markov process on their product space, the transition rate matrix of which is equal to the Kronecker sum of the individual rate matrices [6]. It is defined as

\[
(A \otimes B)_{(i',j')(k,l)} = a_{ij} \delta_{jj} + \delta_{ij} b_{jj}.
\]

(4)

(Though the notation is identical, the Kronecker sum is not the direct sum.) The probabilities of all possible values of the group assignment \( \phi_i \) at time \( t \) forms an \( m^n \)-vector \( P(t) \) satisfying

\[
P(t) = A P(t) ,
\]

(5)

where the transition rate matrix \( A \) is the \( n \)-fold Kronecker sum of \( A \):

\[
A = \bigoplus_{i=1}^n A_i.
\]

(6)

### 3.2 The graph process

We now suppose that the groups of all vertices have been specified throughout time by \( \{ \phi_i : t \geq 0 \} \). For a specific edge \( e = \{v,w\} \), the group histories of its endpoints are given by \( \{ \phi_v(t) : t \geq 0 \} \) and \( \{ \phi_w(t) : t \geq 0 \} \). Given these, we may define the stochastic process for the type \( \kappa_e(t) \) of the edge. The probabilities of the edge being of type \( 1 \) through \( r \) at time \( t \) are given by a vector \( q_e(t) \), satisfying

\[
q_e = B_{\phi_v(t) \phi_w(t)} q_e .
\]

(7)

Because the edges are undirected, we require \( B_{ij} = B_{ji} \).

The solution of (7) must be specified piecewise. When the groups \( i = \phi_v(s) \) and \( j = \phi_w(s) \) are constant over an interval \( t \leq s < t' \), \( q_e(s) = B_{ij} q_e(t) \). The time-history of \( q_e(t) \) is obtained by piecing together such fragments into a continuous function. These edge processes are conditionally independent given \( \{ \phi_i : t \geq 0 \} \), so the Markov process for the graph \( \kappa_t \) may be written as follows. Let \( Q(t) \) denote the \( r^{n(n-1)/2} \)-vector of probabilities of all possible graphs \( \kappa_t \). The evolution of \( Q(t) \) is given by

\[
\dot{Q} = B_{\phi} Q ,
\]

(8)

where the rate matrix \( B_{\phi} \) is a Kronecker sum of \( B_{ij} \) matrices:

\[
B_{\phi} = \bigoplus_{e \in E[2]} B_{\phi(e_1) \phi(e_2)} .
\]

(9)

The entire time-history of \( Q(t) \) is obtained by piecing together fragments satisfying (8) into a continuous function.

### 3.3 Inference on the joint process

Consider the \( m^{n^2(n-1)/2} \)-vector \( \mathcal{R}(t) \) of the probabilities of all group assignment / graph pairs \( (\phi, \kappa_t) \) at time \( t \). This vector obeys

\[
\dot{R} = CR ,
\]

(10)

where \( C \) is formed in a manner similar to a Kronecker sum:

\[
c_{(\phi',\kappa')/(\phi,\kappa)} = A_{\phi' \phi} \delta_{\kappa' \kappa} + \delta_{\phi' \phi} B_{\phi' \phi \kappa' \kappa} .
\]

(11)

The matrix \( C \) may be used to solve the inference problem of determining the probability distribution of the group assignment \( \phi_i \) at time \( t \) given the time history of the graph \( \{ \kappa_t : t \geq 0 \} \). This time history may be decomposed into periods during which the graph remains constant, punctuated by times at which a single edge changes type. Accordingly, we decompose \( C \) into its \( \kappa' = \kappa \) and \( \kappa' \neq \kappa \) parts. The expression in (11) is not of this form because the \( B_{\phi} \) matrix in the second term has a non-zero diagonal, so we move this diagonal to the first term, which yields

\[
c_{(\phi',\kappa')/(\phi,\kappa)} = A_{\phi' \phi} \delta_{\kappa' \kappa} + \delta_{\phi' \phi} B_{\phi' \phi \kappa' \kappa} .
\]

(12)

where

\[
A_{\phi' \phi} = A_{\phi' \phi} + \delta_{\phi' \phi} B_{\phi' \phi \kappa' \kappa} ,
\]

\[
B_{\phi' \phi' \kappa' \kappa} = B_{\phi' \phi' \kappa' \kappa} - \delta_{\phi' \phi} B_{\phi' \phi \kappa' \kappa} .
\]

(13)

The \( A_{\phi' \phi} \) matrices govern the evolution of the group-assignment probabilities \( \phi \) while the graph \( \kappa \) remains constant, whereas the \( B_{\phi' \phi} \) govern the changes when \( \kappa \) jumps. Explicit formulas for \( A_{\phi' \phi} \) and \( B_{\phi' \phi} \) may be obtained from (4), (6), (9), and (13), which yield

\[
A_{\phi' \phi} = \sum_{v=1}^{n^2(n-1)/2} a_{\phi(v) \phi(v)} + \sum_{e \in E[2]} b_{\phi(e_1) \phi(e_2) \phi(e)} \delta_{\phi' \phi} ,
\]

if \( \phi' = \phi \)

\[
A_{\phi' \phi} = \sum_{v=1}^{n^2(n-1)/2} a_{\phi'(v) \phi(v)} ,
\]

if \( \phi' = \phi \) except at vertex \( v \)

\[
0 ,
\]

otherwise,

(14)

and

\[
b_{\phi' \phi' \kappa' \kappa} = b_{\phi(e_1) \phi(e_2) \phi' \phi(e)} \delta_{\phi' \phi} ,
\]

if \( \kappa' = \kappa \) except at \( e \)

\[
0 ,
\]

otherwise.

(15)

A formal derivation (omitted for brevity) yields the following governing equations for group assignment probabilities. In these equations \( K \) and \( \Phi \) denote random variables, and \( \kappa \) and \( \phi \) specific instances; bold variables denote entire time histories rather than instantaneous
values; and the superscripts ‘−’ and ‘+’ distinguish between the status before and after any change to the graph:

\[
\Pr(\Phi^+_i = \phi | K^+_i = \kappa^+_i) \propto \\
\sum_{\phi} \exp \left( A^+_i \cdot (t' - t) \right) \Pr(\Phi^+_i = \phi | K^+_i = \kappa^+_i)
\]

when \( \kappa_s = \kappa^+_s \) for \( t \leq s < t' \),

and

\[
\Pr(\Phi^+_i = \phi | K^+_i = \kappa^+_i) \propto \\
b_{\phi(\kappa_1)\phi(\kappa_3)}c_{\phi(\kappa_2)} \Pr(\Phi^-_i = \phi | K^-_i = \kappa^-_i)
\]

provided \( \kappa^+_i \) differs from \( \kappa^-_i \) at one edge \( e \).

Equation (16) indicates that to update the vector of probabilities of group assignments from time \( t \) to \( t' \) while the graph remains some constant \( \kappa \), the vector is multiplied by the matrix \( \exp( A^- (t' - t) ) \), then normalized to sum to 1. Equation (17) indicates that to update this vector when a single edge \( \{v, w\} \) changes from type \( k \) to type \( k' \), the probability of each group assignment \( \phi \) is multiplied by \( b_{\phi(\kappa_1)\phi(\kappa_3)}c_{\phi(\kappa_2)} \), and then normalized. (When \( i \) edges change at exactly the same time, one averages over all \( i! \) orders of applying (17) serially.)

4 Examples

4.1 Simple example: \( m = r = 2, n = 3 \)

To demonstrate the above formulas, we consider a case with \( m = 2 \) groups (black = 1 and white = 2), \( r = 2 \) edge types (off = 1 and on = 2), and \( n = 3 \) vertices. We will use a transition rate matrix \( A \) which stipulates that the average rate for a vertex to change group is once per time unit, whether this be from black to white or vice versa:

\[
A = \begin{pmatrix}
-1 & 1 \\
1 & -1
\end{pmatrix}, \quad p^\infty = \begin{pmatrix} 0.5 \\
0.5 \end{pmatrix}.
\]

Here \( p^\infty \) is the stationary vector of \( A \), which specifies the expected distribution of black and white vertices after transient behavior has decayed.

The processes for edges turning on and off are governed by these transition rate matrices:

\[
B_{11} = B_{22} = \begin{pmatrix}
-3 & 1 \\
3 & -1
\end{pmatrix}, \quad B_{12} = \begin{pmatrix}
-1 & 3 \\
1 & -3
\end{pmatrix}.
\]

Using seconds as a time unit, (19) indicates that between vertices of the same group, edges that are off (column \( k = 1 \) of \( B_{11} \) and \( B_{22} \)) turn on at an average rate of \( 3 \) times per second, whereas those that are on (column \( k = 2 \)) turn off only once per second on average. Conversely, between vertices in different groups (see matrix \( B_{12} \)), edges turn off quickly (rate \( 3 \)), but turn on more slowly (rate \( 1 \)).

The edges in this example are homophilic: they tend to connect vertices in the same group. The stationary vector of the matrix \( B_{11} \) is \((0.25,0.75)^T\), so if two vertices each remained black (i.e., in group 1), then over time there would be an edge connecting them 75% of the time. The fact that vertices change complicates the asymptotic analysis of edge type, however. The limiting joint distribution of an edge’s type and the groups of its endpoints is given by the stationary vector of a \( m^2 \times r \)-dimensional square matrix \( C \) which may be realized as the \( n = 2 \) case of \( C \) in (11):

\[
c_{\phi(\kappa_1)\phi(\kappa_3)}c_{\phi(\kappa_2)} = (\delta_1 \theta_1 + \delta_2 \theta_2) \delta_{\phi(\kappa_1)\phi(\kappa_3)}c_{\phi(\kappa_2)} , \quad \delta_1 \theta_1 + \delta_2 \theta_2 = b_{\phi(\kappa_1)\phi(\kappa_3)}c_{\phi(\kappa_2)} , \quad \delta_1 \theta_1 + \delta_2 \theta_2 = b_{\phi(\kappa_1)\phi(\kappa_3)}c_{\phi(\kappa_2)} .
\]

The stationary vector of \( C \) provides the asymptotic distribution of edge types between vertices in groups \( i \) and \( j \). In this case, it indicates that edges predominate over non-edges within groups by the ratio 5:3, whereas non-edges predominate by 5:3 between groups.

Figure 2 demonstrates the inference problem for this simple scenario. The sequence of graphs at the top of the figure represents the evidence \( \{\kappa(t): 0 \leq t \leq 1\} \). The graph initially has no edges. An edge appears at time \( t = 0.074 \), and another at \( t = 0.119 \). The first edge then disappears at \( t = 0.631 \). The inference problem is to determine the probabilities of each of the eight possible group hypotheses. Because of the symmetry of the scenario, no evidence can distinguish between an assignment and its opposite. For example, the green line in Figure 2 represents the probability of all vertices being black, and also the probability of all vertices being white. The blue, purple, and red lines represent similar pairs of assignments.

The behavior in Figure 2 may be understood qualitatively as follows. The probability of the vertices being all black (green line) decays initially because this situation is conducive to edges existing, whereas no edges are initially in evidence. When an edge appears, the probability of each group assignment \( \phi \) jumps up or down depending on whether \( \phi \) assigns the edge’s endpoints to the same or different groups. The bottom sequence of group assignments represents the ground truth \( \phi(t) \), which was used in the generation of \( \kappa(t) \). The line corresponding to this ground truth value is shown in yellow.

To perform the probability computation shown in Figure 2, we begin, for convenience, by setting the initial assignment probabilities to the distinct values 0.19, 0.13, 0.11, and 0.07 for the green, red, blue, and purple cases,
respectively. These are used (twice each) to form the initial 8-dimensional vector $P_0 = \Pr(\Phi_t^+ = \phi \mid \kappa_t^+ = \kappa_t^0)$ of group assignment probabilities needed in (16). Next, we form the $8 \times 8$ matrix $A_t^\uparrow$ according to (14) for the initial graph $\kappa = (1,1,1)$, and compute $\exp(A_t^\uparrow t)P_0$ for $0 \leq t \leq 0.074$. This quantity is a vector whose entries are the probabilities of a group assignment $\phi$ and the graph $\kappa$ persisting. Therefore the sum of this vector is not 1, but rather the probability of the graph $\kappa$ persisting through time $t$ (at time $t = 0.074$, for example, this probability is 0.624). However, because the graph is given as data, we normalize $\exp(A_t^\uparrow t)P_0$ to get the probabilities of the group assignments given the data. At time $t = 0.074$, these probabilities are 0.145, 0.140, 0.124, and 0.091 for the green, red, blue, and purple cases, respectively.

Applying the jump computation at $t = 0.074$ is much simpler. According to (17), we just multiply each of the probabilities by element (2,1) (because the edge changes to type 2 from type 1) of the $B_{ij}$ matrix for which $i$ and $j$ are the groups of the endpoints of the edge that changes. These are both 1 (or both 2) for the green and red cases, but are different for the blue and purple. Hence the green and red probabilities (0.145 and 0.140) are multiplied by 3, while the others (0.124 and 0.091) are multiplied by 1. These are then normalized to give 0.203, 0.197, 0.058, and 0.042 for the green, red, blue, and purple cases, respectively, and used as the initial values $P_0$ with a new $A_t^\uparrow$ matrix for the next phase of evolution.

### 4.2 Larger example: $m = 3$, $r = 4$, $n = 12$

We now consider an example with $m = 3$ groups, represented in what follows by the same color scheme as Figure 1: green = 1, yellow = 2, and purple = 3. The transition rate matrix $A$ and its stationary vector $p^\infty$ are

$$A = \begin{pmatrix} -1.3 & 0.7 & 0.5 \\ 0.8 & -1.6 & 1.0 \\ 0.5 & 0.9 & -1.5 \end{pmatrix}, \quad p^\infty = \begin{pmatrix} 0.318 \\ 0.360 \\ 0.322 \end{pmatrix}, \quad (21)$$

so yellow vertices are slightly favored.

The example has $r = 4$ edge types (again using the color scheme of Figure 1: white = 1, black = 2, blue = 3, and red = 4). This requires the specification of six $4 \times 4$ matrices $B_{ij}$ comprising $B$. These are listed in compact form below: the top row giving elements (1,2), (1,3), and (1,4) (the rates of black, blue, and red edges turning off) and the bottom giving (2,1), (3,1), and (4,1) (the rates of turning on). The other off-diagonal elements are zero (i.e., edges can only turn on and off, not change color directly), and the diagonal elements are determined by the requirement that column sums are zero.

$$B_{11} = \begin{pmatrix} 16 & 28 & 39 \\ 6.5 & 2.2 & 0.1 \end{pmatrix}, \quad B_{12} = \begin{pmatrix} 85 & 9.7 & 93 \\ 4.2 & 3.3 & 0.8 \end{pmatrix},$$

$$B_{13} = \begin{pmatrix} 7.5 & 26 & 28 \\ 1.5 & 3.1 & 0.3 \end{pmatrix}, \quad B_{14} = \begin{pmatrix} 91 & 14 & 130 \\ 6.3 & 2.2 & 1.3 \end{pmatrix},$$

$$B_{22} = \begin{pmatrix} 15 & 30 & 14 \\ 3.7 & 2.7 & 0.9 \end{pmatrix}, \quad B_{23} = \begin{pmatrix} 97 & 6.6 & 82 \\ 4.0 & 1.4 & 1.2 \end{pmatrix}. \quad (22)$$

The stationary vector of $C$ defined in (20) provides the limiting distribution of edge types. The white or “non-edge” type predominates in all cases: in the $n = 12$ case below, the expected number of (non-white) edges is only 15.6. Among these are an average of 3.7 black and 1.8 blue edges.
edges within groups, and 1.9 black and 7.4 blue edges between groups. Hence, the black edges are homophilic, and the blue are heterophilic. There are very few red edges (0.75 on average), of which many (36%) occur between vertices in group 3. Ad hoc or rule-based systems would have difficulty exploiting these somewhat weak tendencies (toward homophily in black edges, or group 3 in red) consistently. The formal approach outlined here not only processes this information optimally, but also exploits the duration information for the edges (optimally). The expected duration of an edge of type $k > 1$ between vertices in groups $i$ and $j$ is given by $1/h_{ij,k}$ (i.e., the elements of the top rows in (22)). Thus, black edges are not only twice as likely to occur within group 2 than between groups 1 and 3, but such edges also tend to last 12 times longer.

### 4.2.1 Simulation and results

A simulation of $H(n,A,B)$ was run using $n = 12$ vertices. It was initialized with vertices 1 through 3 in group 1 (green), 4 through 8 in group 2 (yellow) and 9 through 12 in group 3 (purple). First a time history of the group of each vertex was obtained using the rate matrix $A$ in (21). This resulted in a total of 19 times at which vertices changed groups over the span $0 \leq t \leq 0.8$. Given this instance of $\{\phi_t : 0 \leq t \leq 0.8\}$, the types of all the edges were evolved though time $t = 0.8$ using the $B_{ij}$ matrices in (22). The final frame ($\phi_{0.8}$) is part of Figure 3. The assignment $\phi_{0.8}$ is represented by the colors at the centers of the twelve vertices (the colors on the outer band will be discussed shortly), the three groups each having four members at $t = 0.8$. In the graph $\kappa_{0.8}$ we observe that two of the three black edges are homophilic, five of the six blue edges are heterophilic, and a red edge is within group 3. This is consistent with the expected numbers of such events.

![Figure 3: Snapshot of edges and groups at $t = 0.8$](image)

The edge data $\kappa : 0 \leq t \leq 0.8$ generated by the simulation was processed using (16) and (17) to infer the probabilities of all possible group assignments over time. The matrix exponential in (16) is an expensive computation, so it was approximated with second-order Runge–Kutta time stepping. The result of the inference at the final time is shown in the outer bands of the vertices in Figure 3. It is impractical to display the probabilities of all $3^{12} \approx 530,000$ possible group assignments. Instead, these have been marginalized down to the probabilities that each vertex is in each of the different groups. These probabilities are represented by the angular extent of each group’s color around each vertex.

To assess performance, we define the accuracy $f_j(v)$ at a vertex $v$ at some time $t$ for a given run to be the computed probability of $v$ being in the group that ground truth indicates is correct. Seven of the vertices have an accuracy of over 70%, but vertices 4, 6, 9, and 11, have accuracies of only 16%, 37%, 12%, and 21%, respectively. Vertices 4 and 11 have recently changed groups (at $t = 0.766$ and $t = 0.701$, respectively), and it takes time for the inference to catch up with group changes. This may or may not contribute to the low accuracy at vertex 6 as well (which changed group at $t = 0.549$), but it certainly does not explain vertex 9, which remained in the same group throughout the simulation. The reason vertex 9 has low accuracy is simply due to the difficulty level of the parameters in this scenario. It is possible to get “misleading” edges (such as black edges between different groups) which can drive the inference away from the ground truth result instead of toward it. As the number of vertices increases, however, the law of large numbers makes such misleading behavior much rarer.

It is worth emphasizing that $f_{0.8}(9) = 12\%$ is an exact calculation of optimal inference from the data—it cannot be “improved.” Its meaning is that if this scenario were run a vast number of times, and the tiny subset of those runs were chosen in which the data $\kappa : 0 \leq t \leq 0.8$ were the same as in this run, then vertex 9 would, in fact, be in group 3 in 12% of those runs. The accuracy values $f_j(v)$ are plotted in Figure 4 for all vertices at time increments of 0.0009. Also plotted are the times of the 19 instances of vertices changing groups. For example, the blue hash mark at $t = 0.353$ is for vertex 5 (changing from group 2 to 3). The values of $f_j(v)$ are plotted as dots of the same color blue rising from an accuracy of 7% to 26% over the time range $[0.353,0.4]$.

There are various phenomena driving the form of Figure 4. When a vertex changes group, there tends to be a large drop in its accuracy. Typically, the accuracy is quite large before a vertex changes group, in which case the accuracy immediately after the group change must be quite small. This is one reason for jumps, but most jumps are due to edges changing type, which happens more frequently.
These jumps are typically much smaller than the jumps due to vertices changing groups, but can be as large as 30%.

There are also intervals in Figure 4 where the accuracy changes smoothly because the graph remains constant. The interplay between the inference when edges change types and over these constant intervals is somewhat subtle. For example, although black edges are twice as prevalent within group 2 as between groups 1 and 3, black edges turn on four times as frequently between groups 1 and 3 (rate = 6.3) as within group 2 (rate = 1.5). Therefore a black edge turning on initially favors the 1-3 hypothesis (for the groups of its endpoints) over the 2-2 hypothesis by a factor of four. However, because black edges turn off twelve times more frequently between groups 1 and 3 (rate = 91) as within group 2 (rate = 7.5), as the edge persists it increasingly favors hypothesis 2-2. E.g., in the absence of other information, if a black edge persists for 0.1 seconds, then the initial factor of four preference for the 1-3 hypothesis is converted to a factor of fifty in favor of 2-2.

Figure 4 shows that accuracy recovers quickly from vertices changing group. Figure 5 quantifies this. Eight of the nineteen group-change events satisfy the following conditions: (a) high accuracy before the change, and (b) a sufficient time record afterwards to recover (before another the vertex changes again or the run ends). Figure 5 plots the accuracy of each vertex versus time. The black line is the geometric mean of the times required by the vertices to reach various levels of accuracy from 16% to 86%.

5 Conclusions and future directions

This work demonstrates that it is possible to infer the group structure of a dynamic network from its edge data alone. This has been accomplished on a dataset far too small for static group finding techniques to work. The group inference was possible only because the data was enriched by a temporal dimension, because the processes generating the data were known exactly, and because the data was utilized optimally. The derivation and computational results yield a number of key insights into group tracking.

The first insight is that the although the entire system of evolving groups and edge types can be large and complex, it may be described fairly simply in terms of a single rate matrix $C$, which itself has a great deal of exploitable structure due to its systematic construction via Kronecker sums (and the similar construct $(11)$).

The second insight is that it is important to work with Markov processes rather than Markov chains. Although it is conceptually simpler to discretize the problem to work within the framework of Markov chains, the resulting matrices would be dense because they would have to incorporate the possibility of an arbitrary number of edge changes occurring within the time interval of a single step. The dense transition matrices obtained in this way would be numerically costly to work with and would obscure the underlying structure. The way to fix this problem would be to make the step size sufficiently small that it is unlikely that more than one edge changes type in a single time step—but to do this would be to reproduce the Markov process version (in a rather laborious manner).

A third insight about the problem is that it breaks into two cases: a jump case, involving only $B$, and a constant-$\kappa$ case involving both $A$ and $B$. Because $A$ is entirely absent in the jump case, and causes problems in the constant-$\kappa$ case by making the evolution operator non-diagonal, it is tempting to set it to zero when $A \ll B$ in order to simplify the problem. This would be inappropriate without compensating for the role that $A$ plays. In the case where $A$ actually is 0, the accumulation of evidence allows one to achieve virtual certainty about the group assignment for all the vertices. However, setting $A$ to zero artificially allows a false confidence in one’s group assignments to grow because one does not take into account the fact that, despite all the evidence pointing to one group, the group of the vertex may have changed recently. The danger of this is that after a vertex changes group, the artificially high confidence in its group becomes an artificially low probability estimate for the new group. This probability can become exponentially low (e.g., $10^{-100}$) without the mitigating influence of $A$, and hence take much longer to correct via the evidence than if it were only realistically low.
(e.g., 0.01). However, it may be possible to achieve the same effect that $A$ provides in a simple, efficient way, such as setting cutoffs on the probability estimates.

A related insight is that it does not suffice to account for the jump events only. It may seem natural to do so—the jumps events are the only time when something happens. Yet the negative information of things failing to change is equally important. Figure 4 shows that there are significant effects both due to jumps in probability when edges change type as well as rapid changes in probability when edges retain their type.

To make the ideas here practical, it will be necessary to develop an approximation based on retaining only a much smaller set of probabilities, such as the $mn$ marginal probabilities on each vertex depicted in Figure 3. The most straightforward way of doing this is to stipulate independence among the Markov processes that specify the groups of each vertex: not just a priori independence, but independence given the graph data as well. Preliminary results using this method have been obtained and they show promise. The results in Section 4.2.1 teased out group structure that is virtually invisible to the human eye because it constitutes a weak signal which must be processed optimally in order to find it. In contrast, the signal of group structure can be extremely strong in large networks, just difficult to find given the deluge of data.

In addition to efficiency, there are a number of other issues to address to obtain a practical method. First of all, real data is unlikely to provide ground-truth graph values with perfect accuracy. For example, for a case in which edges represent the social relationships between people, many such relationships may not be manifest in, say, cell phone, email, or HUMINT data. The model must be extended to include a measurement likelihood function which describes the distribution of the observed graph data given the underlying ground-truth graph data. Similarly there may be an association problem between the vertices used in the data (e.g., cell phone numbers) and the underlying ground truth vertices (e.g., individuals using the cell phones). It is also questionable whether the Markov process formulation of the problem is sufficiently flexible to cope with real-world data. The temporal characteristics of the data may not be purely stochastic, but rather a stochastic perturbation of deterministic patterns, such as the variation of container shipping transactions within the constraints of trade routes and seasonal shipping patterns. Semi-Markov processes [7] offer more realism, but unfortunately, formation of a joint rate matrix as the Kronecker sum of the rate matrices of its component processes is not valid for semi-Markov processes. Similarly, the Erdős–Rényi-like simplicity of the model $\mathcal{H}(n,A,B)$ may not have sufficient structure to represent real-world networks in sufficient detail. In particular, real-world social networks tend to be organized into hierarchical structures and to have a strong correlation with geography. Indeed, real-world data may not even be well modeled by simple graphs, but require multigraphs to represent the possibility of multiple, simultaneous links between entities, or even hypergraphs to represent $n$-ary relations rather than simply binary edges. In short, there are many challenges in extending the work begun here to realistic scenarios.

The framework in this paper provides a rigorous foundation to begin addressing these challenges. It is hoped that the understanding of the structure and phenomena of dynamically evolving groups on large graphs under the relatively simple model $\mathcal{H}(n,A,B)$ will lead to the understanding necessary to construct viable group tracking methods for dynamic networks in the real world.

This research was partially supported by ONR Contract N0001409C0563. The author thanks Oren Bumgarner for performing the computations in Section 4.2.1, and the reviewers for helpful suggestions.

References


