



Sequential detection of temporal communities by estrangement confinement

Vikas Kawadia¹ & Sameet Sreenivasan²

¹Raytheon BBN Technologies, Cambridge MA 02138, ²Social and Cognitive Networks Academic Research Center, Rensselaer Polytechnic Institute, Troy NY 12180.

SUBJECT AREAS:

STATISTICAL PHYSICS,
THERMODYNAMICS AND
NONLINEAR DYNAMICS

APPLIED PHYSICS

MATHEMATICS AND
COMPUTING

SOFTWARE

Received
13 June 2012

Accepted
24 September 2012

Published
9 November 2012

Correspondence and
requests for materials
should be addressed to
V.K. (vkawadia@bbn.
com)

Temporal communities are the result of a consistent partitioning of nodes across multiple snapshots of an evolving network, and they provide insights into how dense clusters in a network emerge, combine, split and decay over time. To reliably detect temporal communities we need to not only find a good community partition in a given snapshot but also ensure that it bears some similarity to the partition(s) found in the previous snapshot(s), a particularly difficult task given the extreme sensitivity of community structure yielded by current methods to changes in the network structure. Here, motivated by the inertia of inter-node relationships, we present a new measure of partition distance called *estrangement*, and show that constraining estrangement enables one to find meaningful temporal communities at various degrees of temporal smoothness in diverse real-world datasets. Estrangement confinement thus provides a principled approach to uncovering temporal communities in evolving networks.

Community detection has been shown to reveal latent yet meaningful structure in networks such as groups in online and contact-based social networks, functional modules in protein-protein interaction networks, groups of customers with similar interests at online retailers, disciplinary groups of scientists in collaboration networks, etc.¹. Temporal community detection aims to find how such communities emerge, grow, combine and decay in networks that evolve with time. Temporal communities can provide robust network-based insights into complex phenomena such as the evolution of inter-country trade networks, the emergence of celebrities in social media, the formation of distinct political ideologies, the spread of epidemics, trends in venture investment, etc.

Static community detection¹ partitions a network into groups of nodes such that the intra-group edge density is higher than the inter-group edge density. A partition can be specified by labels assigned to the nodes in the network, and a group of nodes with the same label constitutes a community. Methods used to discover communities in static networks find a partition of nodes which optimizes some quality (objective) function that quantifies how community-like the partition is. For time-varying networks, given time snapshots, temporal community detection assigns labels to nodes in each snapshot, and the set of {node, time} pairs that get the same label constitutes a *temporal community*. We define a temporal community structure as a partitioning of the {node, time} pairs over all snapshots that optimizes an appropriate quality function. We focus on the *sequential version* of the temporal community detection problem, where one is allowed to do computations only on the current snapshot while using limited information from the past. Sequential methods are useful in situations where the number of snapshots is large, or fast computation of temporal communities is important as new snapshots become available.

A popular approach to detecting temporal communities is to find static communities independently in each snapshot using some quality function and then “map” communities between snapshots to preserve labels when possible. Examples of this approach include the map-equation method² and the clique percolation method³. However, these methods do not explicitly use the partitions found in past snapshots to inform the search for the optimal partition on the current snapshot. We argue (and show empirically in *Results*) that mapping independently detected communities is likely to miss crucial temporal communities as most quality functions used for static community detection are highly degenerate and extremely sensitive to changes in the network. This has been demonstrated specifically for *modularity*⁴, one of the earliest proposed and still commonly used quality functions, though several others have subsequently been introduced. Good *et al.*⁵ show that for many real-world networks, the modularity landscape is highly degenerate and disordered with numerous partitions yielding similar

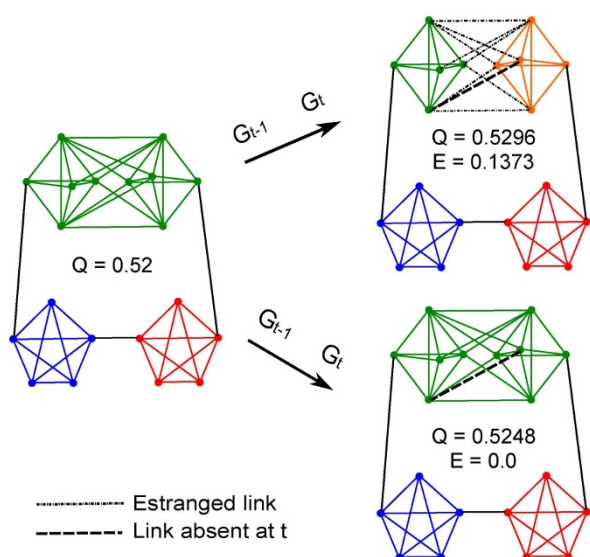


Figure 1 | An example illustrating the detection of temporal communities via estrangement confinement. The network on the left, G_{t-1} consists of 20 nodes and 52 links, and a maximal modularity partition of this network consists of three communities represented by the three colors ($Q = 0.52$). In the next snapshot, the network has evolved to G_t which differs from G_{t-1} only in the absence of a single link, indicated by the dotted line. The top right and bottom right networks both represent the same network G_t , but indicate distinct choices of community partitions available. The partition shown on the top right, P_1^t consists of 4 communities, and is the partition that gives the highest modularity $Q_1^t = 0.5296$. The partition P_2^t for G_t shown on the bottom right which preserves the node partition chosen for G_{t-1} has a slightly lower modularity of $Q_2^t = 0.5248$. The partition P_1^t with higher modularity, however, makes 7 links *estranged*. The estranged links (shown in gray) are those intra-community links at $t-1$ that change to inter-community links at t . Notice that links in the orange community of P_1^t despite having changed their community affiliation from $t-1$ to t are not estranged since they are still intra-community links. In contrast to P_1^t , the partition P_2^t yields no estranged links. Estrangement, E , defined as the fraction of estranged links at t is therefore 0 for P_2^t but $7/51 = 0.13$ for P_1^t . Maximizing modularity while constraining estrangement to a low value (e.g. 0.05) therefore yields P_2^t as the partition for G_t , yielding a smoother temporal progression of the community structure from $t-1$ to t .

values of modularity and constituting distinct local maxima. Importantly, they also show that other community quality functions are also likely to have degenerate quality functions. Moreover, the quality function landscape is highly sensitive to changes in the network, as shown by Karrer *et al.*⁶ for modularity on several synthetic and real networks. Sensitivity implies it is very likely that a rather distinct community structure is detected even when the network changes slightly, which, when coupled with the degeneracy of the quality function landscape, makes consistent mapping of independently detected communities across snapshots very difficult.

To counter these challenges, it is important to use the past community structure when searching for good partitions in the current snapshot to maintain some temporal contiguity between subsequent partitions. Obviously, independent maximization of modularity (or some other quality function) on each snapshot has no incentive to maintain such a temporal contiguity between partitions. Also, the naive approach of initializing the search for a good partition of the current snapshot at the preceding snapshot's optimal partition has the serious drawback in that it fails to detect the birth of new communities (unless a significant number of new nodes are added) since most partition search methods decrease or keep constant the number of communities found.

We propose a principled approach to find meaningful temporal communities that limits the search for near-optimal partitions to those partitions in the current snapshot that bear some similarity to the partitions found in previous snapshots. One of the key challenges is to find a measure of this partition-similarity (or distance) that is appropriate for comparing partitions of different snapshots of an evolving network. None of the existing measures of partition distance, such as Variation of Information (VI)⁵, are suitable for comparing partitions of nodes in distinct snapshots because they do not consider edges, and therefore cannot account for changes in network structure. In particular, we require a measure that is tolerant of differences in partitions when the network has changed significantly but penalizes dissimilar partitioning when there are only minor changes in the network.

We present a novel measure of partition distance, called *estrangement*, which quantifies the extent to which neighbors continue to share community affiliation. This is motivated by the empirical observation that it is some form of social inertia inherent to group affiliation choices that prevents the community structure from changing abruptly^{7,8}. The estrangement between two time-ordered snapshots is defined as the fraction of edges that stop sharing their community affiliation with time. In other words, estrangement is the fraction of intra-community edges that become inter-community edges as the network evolves to the subsequent snapshot, as illustrated in Fig. 1.

Our method of detecting temporal communities consists of maximizing modularity in a snapshot subject to a constraint on the estrangement from the discovered partition in the previous snapshot. The amount of estrangement allowed controls the *smoothness* of the evolution of temporal communities, and varying it reveals various levels of resolution of temporal evolution of the network. The estrangement constrained modularity maximization problem described above is at least as hard as modularity maximization which is NP-complete⁹. Moreover, known heuristic methods for unconstrained modularity maximization are not directly applicable to the constrained version. However, we show that the dual problem constructed using Lagrangian relaxation can be tackled by adapting techniques used for unconstrained modularity maximization, specifically a version of the Label Propagation Algorithm (LPA)^{10,11}.

Some recent proposals for detecting temporal communities, similarly to ours, use the past community structure. Mucha *et al.*¹² extend the notion of random walk stability, introduced by Lambiotte *et al.*¹³, to multi-slice networks and show that optimization of this stability yields coherent temporal communities (Incidentally, estrangement can be interpreted as *temporal stability* as we show in SI). However, their method is not sequential as it requires all slices (snapshots) to be aggregated into a stacked graph by introducing arbitrary weighted links between node copies in different slices. No principled method is presented for picking the weights of the inter-slice links. Our method is closest to *evolutionary clustering* introduced by Chakrabarti *et al.*¹⁴ where the quality of a community partition is measured by a combination of its *snapshot cost* and its *temporal cost*. However, unlike our method, the work of Chakrabarti *et al.*¹⁴, does not prescribe specific relative contributions of the two costs, or demonstrate the effect of varying these contributions. Furthermore, the partition distance measure and the optimization techniques we use are different. *GraphScope*¹⁵ finds temporal communities by breaking the sequence of graph snapshots into *graph segments* and finding good communities within each graph segment such that the total cost of encoding the sequence of graphs is minimized. However, it can only be used on unweighted networks. Subsequent techniques such as *FacetNet*¹⁶ and *MetaFac*¹⁷ apply the evolutionary clustering approach to partitions derived from a generative mixture model approximation of the network adjacency matrix. A distinctive drawback of generative models in the context of community detection is the necessity of providing a priori, the number of communities in the network, or using community quality function based methods to find the most suitable

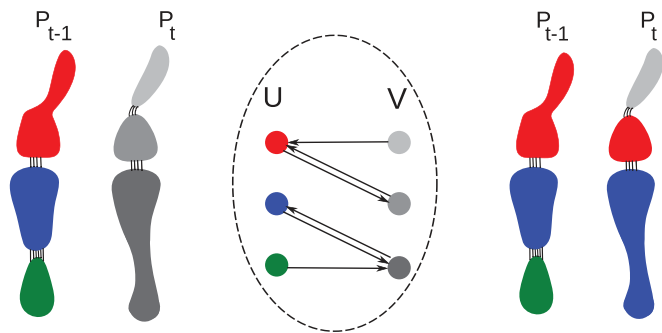


Figure 2 | Mapping community labels from time $t - 1$ to time t . The left panel shows the situation after estrangement confinement has found a partition of the graph at time t , consisting of three communities. Two of these have arisen due to an uneven split of the red community at $t - 1$, and one due to the merging of the blue and green communities at time $t - 1$. The mapping procedure causes fewest nodes to change labels from $t - 1$ to t . The center panel shows the bipartite construction that the mapping procedure uses. Here, nodes on the left (set U) represent communities at $t - 1$ and nodes on the right (set V) represent those at time t . Each node in U has an outgoing link to the node in V with whom its Jaccard overlap is maximal. Similarly each node in V has an outgoing link to the node in U with whom its Jaccard overlap is maximal. For simplicity, we say that each node points to its maximal overlap partner in the other set. Once these links are drawn, the mapping procedure allows inheritance of labels only between pairs of nodes which have bidirectional links between them, i.e., a node in U (community at $t - 1$) passes on its label to a node in V (community at t) only if they are maximal overlap partners of each other. Consequently, a node in U which is not bidirectionally connected to any node in V , does not pass on its label (e.g., the green node in U). Similarly, a node in V which is not bidirectionally connected to any node in U , does not inherit a label, and therefore obtains a new label (e.g., the topmost node in V). The progression of appropriately labeled communities from $t - 1$ to t after the mapping step is shown in the panel on the right.

number of communities a posteriori. Also, these modeling techniques assume that the networks are generated by a given stochastic data model. However, as argued by Breiman¹⁸, the utility of such techniques is limited by the accuracy of the models which are generally difficult to design for complex networks. In contrast, our approach is based on empirically observed social inertia in community affiliation and does not try to model the possibly complex evolution of the network itself. Thus, in summary, our method is sequential, does not need any generative model for network structure or evolution, and is applicable to both weighted and unweighted networks.

Results

Our key results include the definition of the novel partition distance measure of estrangement, a formulation of the problem of finding temporal communities as a constrained optimization problem, an efficient agglomerative method to solve the problem that relies critically on the locally decomposable nature of estrangement, and an analysis of the temporal communities found by our method in various synthetic and real complex networks.

Problem formulation. Given network snapshots G_{t-1} , G_t and the partition P_{t-1} that represents the community structure at time $t - 1$, find a partition P_t of G_t that solves the following constrained optimization problem:

$$\begin{aligned} & \underset{\mathcal{P}}{\text{maximize}} && Q(\mathcal{P}) \\ & \text{subject to} && E(\mathcal{P}) \leq \delta. \end{aligned} \quad (1)$$

Here Q is a quality function for the community structure in a snapshot, \mathcal{P} denotes the space of all partitions, and E is a measure of distance or dissimilarity between the community structure at times t and $t - 1$. The formulation above is based on the intuition that temporal communities can be detected by optimizing for quality in the current snapshot while ensuring that the distance from the past community structure is limited to a certain amount, as specified by the parameter δ . Smaller values of δ imply greater emphasis on temporal contiguity whereas larger values of δ place greater focus on finding better instantaneous community structure. Hence, we refer to δ as the temporal divergence, or simply divergence. We emphasize that our formulation is independent of the specific community structure quality function used. In this paper, we use modularity⁴, a widely studied and tested quality function, which is defined as:

$$Q = \frac{1}{2M} \sum_{u,v} \left(A_{uv} - \frac{k_u k_v}{2M} \right) \delta(l_u, l_v), \quad (2)$$

where A is the adjacency matrix for the network, k_x is the degree of node x , l_x is the label assigned to x in this partition, and M is the total number of edges in the network. $\delta(i, j)$ is 1 if and only if $i = j$, and 0 otherwise. Here a partition P is specified by the labels $\{l_1, l_2, \dots, l_N\}$ assigned to the nodes. Modularity has also been generalized to weighted networks¹⁹.

For measuring partition distance, we use our novel measure of estrangement which we now define precisely. Given network snapshots G_{t-1} , G_t and partitions P_{t-1} and P_t , an edge (u, v) in G_t is said to be *estranged* if $l_u \neq l_v$ in P_t , given that u and v were neighbors in G_{t-1} and $l_u = l_v$ in P_{t-1} . Estrangement is now defined as the fraction of estranged edges in G_t . Note that equality of labels is required only within partitions, not across partitions. Estrangement can be written as:

$$E = \frac{\sum_{u,v \in G_t} Z_{uv} (1 - \delta(l_u^t, l_v^t))}{2M} \quad (3)$$

where $Z_{uv} = \delta(l_u^{t-1}, l_v^{t-1}) \sqrt{A_{uv}^{t-1} A_{uv}^t}$, and A_{t-1} and A_t are the adjacency matrices of G_{t-1} and A_t respectively. The square root term ensures that the definition applies to weighted networks as well, where M is taken to be the sum of the weights of all the edges in the network. Specifically, the term $\sqrt{A_{uv}^{t-1} A_{uv}^t}$ implies that if the weight of an edge whose endpoints continue to share labels changes from time $t - 1$ to t , we take the geometric mean of the weights when computing the partition distance. Estrangement can take values between 0 and 1, with 0 estrangement implying maximum possible similarity between the community structure in the two snapshots of the network and a value of 1 implying maximum possible dissimilarity.

Duality based optimization approach. Greedy local optimization methods used for modularity maximization cannot be directly used to solve the constrained optimization problem in Eq. 1, since the space of solutions is now confined to the set of partitions which respect the constraint. We use the Lagrangian duality approach for constrained optimization. Henceforth, for notational simplicity, unless otherwise stated, all quantities of interest are with respect to the current snapshot G_t . Following the dual formulation²⁰, we write the Lagrangian L and the Lagrange dual function g corresponding to the primal problem (Eq. 1) as:

$$\begin{aligned} L(\mathcal{P}, \lambda) &= Q - \lambda(E - \delta) \\ g(\lambda) &= \sup_{\mathcal{P}} L(\mathcal{P}, \lambda) \end{aligned} \quad (4)$$

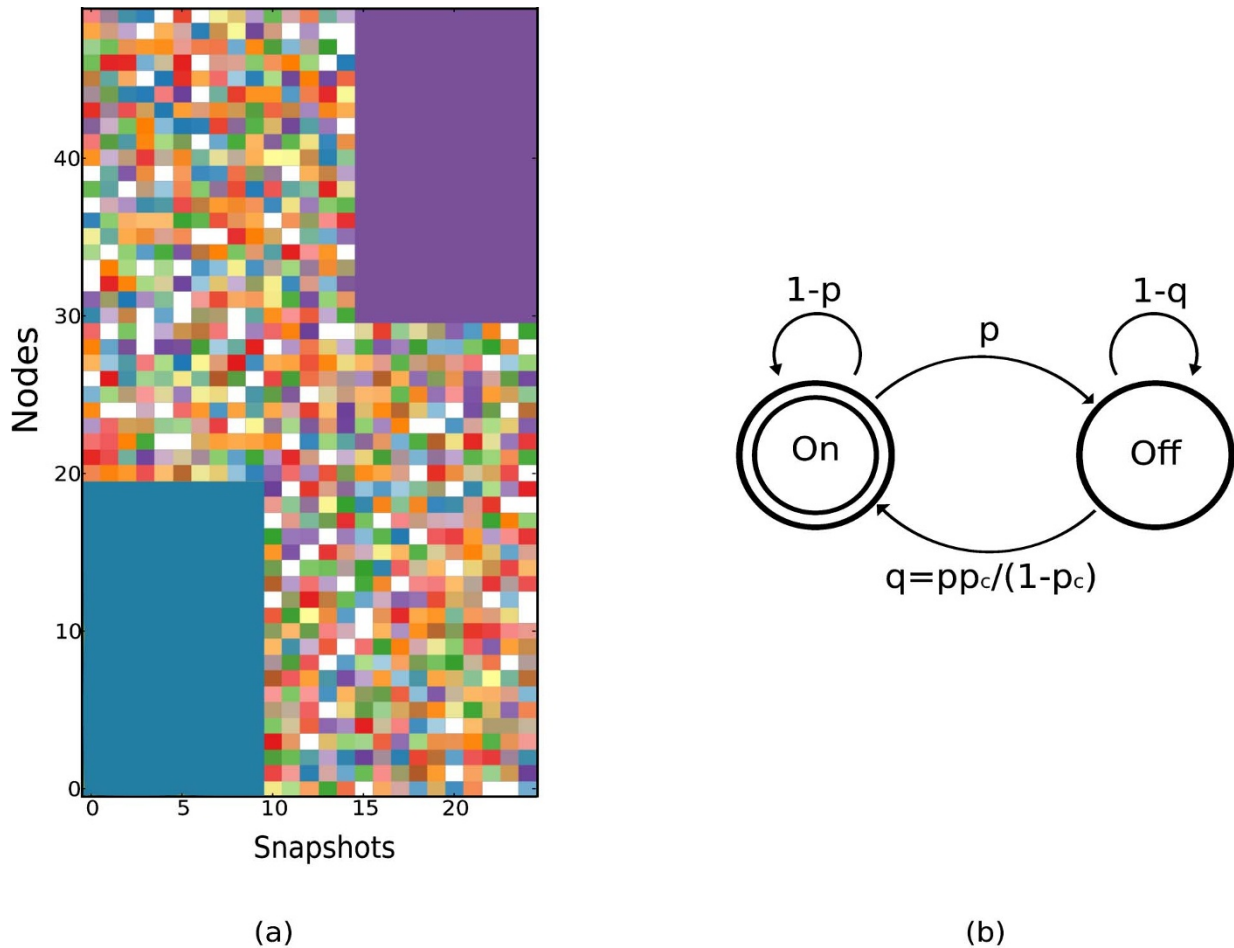


Figure 3 | (a) An example of ground truth temporal communities that are used to generate evolving synthetic networks according to the markovian evolution shown in (b) and explained below. We use an impressionistic visualization to show the temporal communities, that we refer to as the *evolution chart*, in which nodes are shown along the Y axis and the snapshot number along the X axis. Each “pixel” in the evolution chart corresponds to a particular node at a given time, and the color represents the community of that node at that time t . The left panel shows a network consisting of $n = 50$ nodes with two temporal communities consisting of 20 nodes each. The first community arises from the markovian evolution of nodes 0 – 19 over the first 10 snapshots, while the second arises from the markovian evolution of nodes 30 – 49 over the last 10 snapshots. (b) Schematic of Markovian evolution for intra-community edges, where an edge that exists in the current snapshot disappears with probability p in the subsequent snapshot, while a non-existing edge appears with probability q which is chosen such that the group density p_c is preserved.

where λ is the Lagrange multiplier. For every value of λ , the function $g(\lambda)$ yields an upper bound to the optimal value Q^* of the primal problem. We are interested in the value of λ that yields the smallest upper bound, which would in turn give us the best estimate of Q^* subject to the constraint on E . This dual problem corresponding to the primal problem in Eq. 1 is:

$$\begin{aligned} &\text{minimize} && g(\lambda) \\ &\text{subject to} && \lambda > 0 \end{aligned} \tag{5}$$

If the minimum of $g(\lambda)$ occurs at λ^* , the optimal partition for a given snapshot is one that yields the supremum of $L(\mathcal{P}, \lambda^*)$ over all partitions.

Solving the dual problem to find the best partition requires computing the Lagrange dual function $g(\lambda)$, which itself involves a maximization. We show that the Lagrange dual can be computed by adapting known methods for unconstrained modularity maximization. We introduce a hierarchical version of LPA¹⁰, which we refer to as HPLA, and which works by greedily merging communities that provide the largest gain in the objective function, and then repeating the procedure on an *induced* graph in which the communities from the previous steps are the nodes. In general, variants of LPA can be

constructed by modifying the local objective function that the label update is maximizing. Barber and Clark¹⁰ propose one such variant, LPAm, for modularity maximization. We construct the label update rule for HPLA in a similar vein for the optimization problem given by Eq. 4. Recall that a partition \mathcal{P} is specified by the labels $\{l_1, l_2, \dots, l_N\}$ assigned to the nodes. Then, in HPLA, each node x updates its label l_x following the rule:

$$l_x = \arg \max_l \left(N_{xl} - \frac{k_x K_l}{2M} + \frac{k_x^2}{2M} \delta(l_x, l) + \lambda O_{xl} \right), \tag{6}$$

where $N_{xl} = \sum_u A_{ux} \delta(l_u, l)$, $O_{xl} = \sum_{u \neq x} Z_{ux} \delta(l_u, l)$ and $K_l = \sum_u k_u \delta(l_u, l)$. Here O_{xl} is the extra term that arises due to the constraint on E . We show in Methods that the above update rule converges to a local optimum of $L(\mathcal{P}, \lambda)$, and also that the optimization of L is further improved by the additional hierarchical procedure present in HPLA. We note that HPLA works well for optimizing $L(\mathcal{P}, \lambda)$ because estrangement, similarly to modularity, can be decomposed into node-local terms which allows L to be optimized by each node updating its label based on those in its neighborhood.

Once the Lagrange dual has been computed, we solve the dual problem (Eq. 5) of finding the best Lagrange multiplier by using

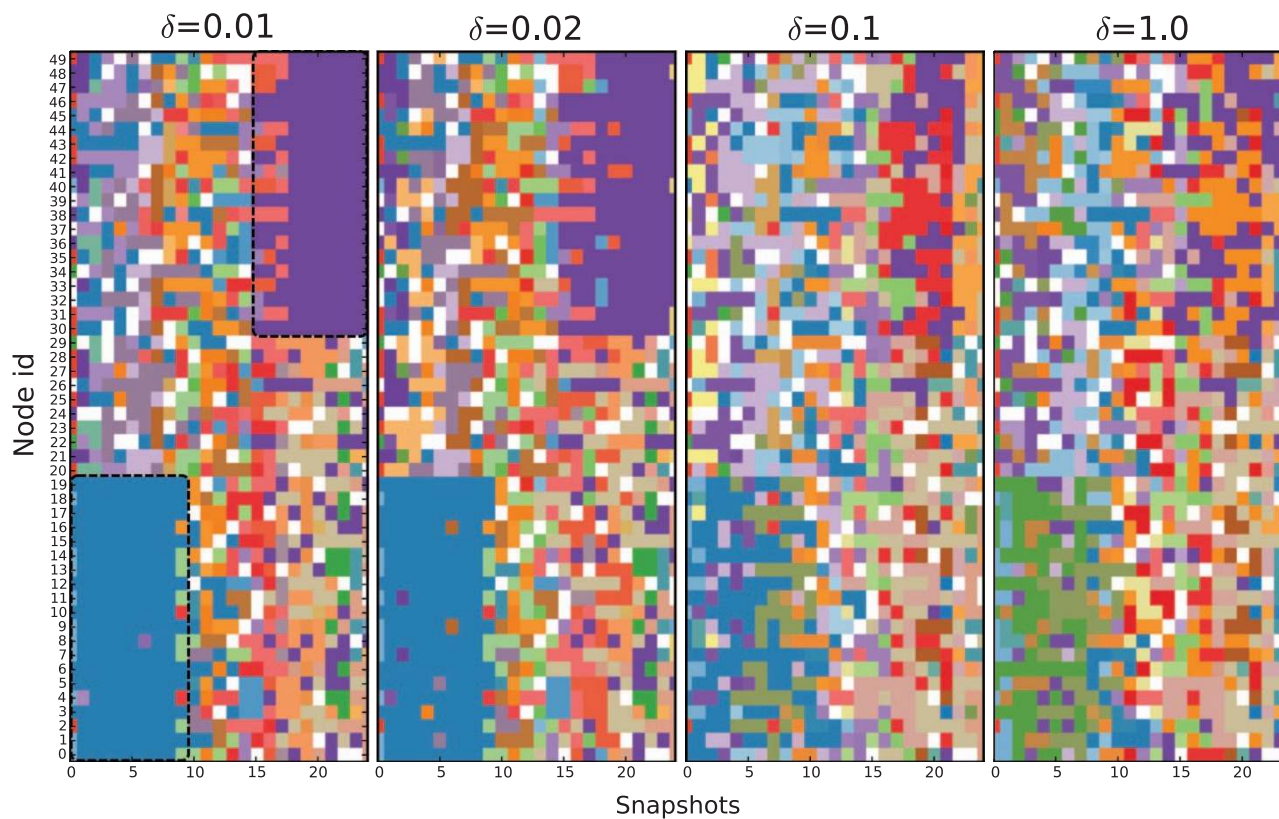


Figure 4 | Estrangement confined modularity maximization allows detection of temporal communities in the benchmark network presented in Fig. 3. The markovian evolution parameters (described in text) for this network are set to $p = 0.6$ and $q = p p_c / (1 - p_c) = 0.4$, which preserve the edge density ($p_c = 0.4$) within a dense group (unless it ceases to exist). The density of connections in the random network background is $p_r = 0.05$. The ability to detect the temporal communities diminishes as the constraint on estrangement is relaxed ($\delta \rightarrow 1$) with the poorest results obtained for independent modularity maximization ($\delta = 1$).

Brent's method²⁴ which is commonly used for non-differentiable objective functions (see *Methods*).

The optimization procedure and the HLP update rule presented above apply to weighted networks as well by considering k_u to be the *strength* of node u instead of the degree, where strength is defined as the sum of the weights of adjacent edges, and by considering M to be sum of the weights of all the edges in network.

Finally, after the best community partition for G_t has been found, we need to find an appropriate mapping of communities at time t to those found at time $t - 1$. We use a mutually maximal matching procedure illustrated in Fig. 2. Specifically, we map those communities across two consecutive snapshots that have the maximal mutual Jaccard overlap between their constituent node-sets (Jaccard similarity of two sets is defined as the size of their intersection set divided by the size of their union), and generate new identifiers only when needed.

Temporal communities in synthetic and empirical networks. Next, we apply the estrangement confinement method to synthetic and real networks and show the temporal communities obtained by varying the temporal divergence allowed, and their relation to ground-truth or meta-data where available.

We start by describing our method to generate realistic synthetic benchmarks for testing temporal communities. Given a target temporal community structure, we generate a snapshot sequence consisting of dense groups (corresponding to the communities) embedded in a random background, with links in the dense groups undergoing markovian evolution and thus giving rise to a temporal community that persists over some period of time. An example target temporal community structure is shown in Fig. 3 which consists of two

temporal communities of 20 nodes each that exist for the first 10 and the last 10 snapshots respectively in 25 snapshots of a 50 node network. Each of the remaining nodes is a community by itself which lasts for exactly one snapshot, or equivalently, does not belong to any temporal community.

The initial snapshot in the synthetic networks consists of an instance of an Erdős-Rényi random graph ($ER(n, p_r)$), among the n nodes where any edge exists independently with probability p_r , and intra-community edges exist with an additional probability of p_c (over the background probability of p_r). Subsequent snapshots are generated by first creating a new random instantiation of ($ER(n, p_r)$), and enforcing a markovian evolution for the edges within a temporal community while it exists in the target temporal community structure. Specifically, an edge that exists in the current snapshot disappears with probability p in the subsequent snapshot, while a non-existing edge appears with probability q (Fig. 3). The markovian evolution thus gives rise to a temporal community that persists over some period of time depending on the values of p and q chosen, since these parameters control the edge density within the community. For the choice $q = \frac{p p_c}{(1 - p_c)}$ the initial edge density within the community is preserved in the subsequent evolution. Using this prescription, we generate different sequences of network evolution for the ground truth temporal communities shown in Fig. 3 by varying p_c and p and setting $p_r = 0.05$.

In Fig. 4, we show the evolution chart of our results on the above synthetic networks with $p_c = 0.4$ and $p = 0.6$. This implies that the average density of edges inside the dense groups is 0.4 and 60% of the edges change in each snapshot. For low enough values of δ , our method is able to detect the temporal communities, even in this rapidly evolving network. Independent modularity maximization

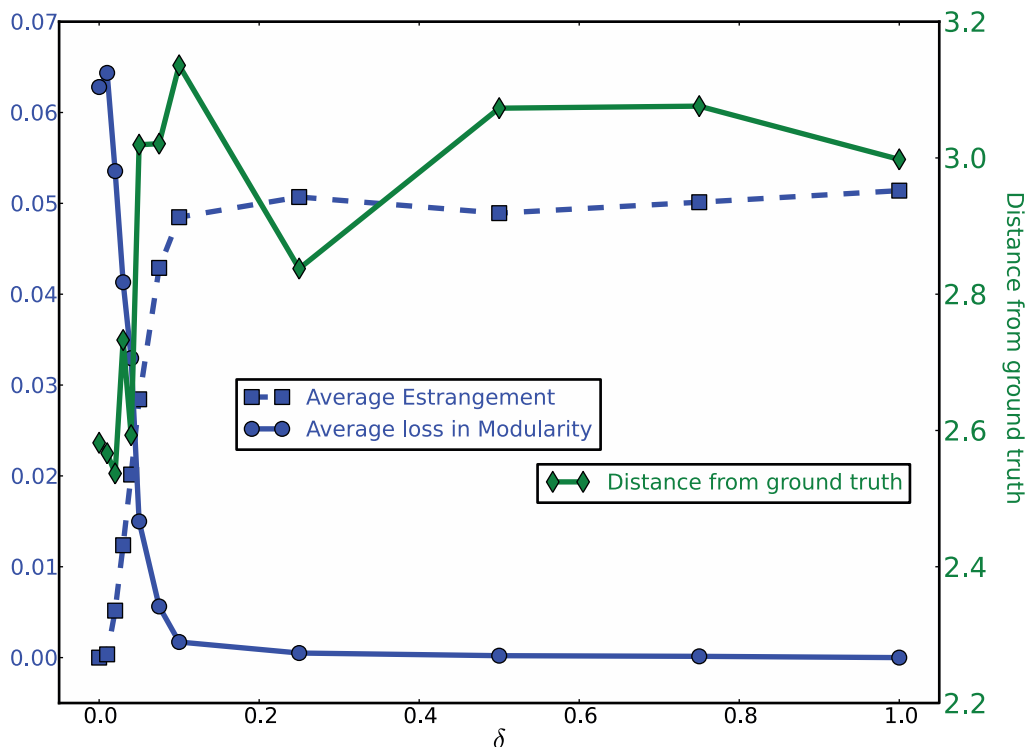


Figure 5 | Effect of varying the temporal divergence on the distance from the ground truth for benchmark evolving network described in Fig. 4. The average loss in modularity (over all snapshots) due to the estrangement constraint decreases as the constraint is relaxed. Estrangement, on the other hand, increases as the constraint is relaxed. The distance of the obtained temporal partition (labeled on the right y axis) from the benchmark ground truth as quantified by VI is lowest around $\delta = 0.05$. The range of this distance varies roughly between 2.5 and 3.1 (labeled on the right y axis) as δ is varied, which is significant since VI is a logarithmic measure (see Methods). Thus, by varying the constraint on estrangement we get a different set of temporal communities which could all be meaningful. The average loss in modularity relative to the average gain in estrangement indicates the range of values of δ which might yield meaningful temporal communities.

(which corresponds to $\delta = 1$) is unable to detect the temporal communities as shown in the rightmost panel in Fig. 4. For a more quantitative comparison of the detected temporal communities to those known to be present in the ground truth, we use VI which is a common metric to evaluate the distance between two partitions of a set⁵. A static community is a partitioning of the set of nodes of the network, while a temporal community is defined as a partition of the set of {node,time} pairs. Thus using VI we can measure the distance of the partition of {node,time} pairs produced by a temporal community detection algorithm from the partition defined in the ground truth shown in Fig. 3 (see Methods for details).

In Fig. 5, we show the effect of varying δ on the synthetic network used in Fig. 4. Low values of δ yield low estrangement but also yield a lower value of modularity compared to what would result from unconstrained modularity maximization. Thus, reduction in estrangement comes at the expense of modularity. There appears to be no “correct” value of δ for obtaining a meaningful structure, but in-practice very low values of δ (0.05 or less) provide smooth communities. Fig. 5 shows that VI is about 0.5 lower (which is substantial since VI is a logarithmic measure) for low values of δ than the maximum VI seen. Despite fluctuations in VI values due to the stochasticity inherent in greedy optimization on partition space, the VI curve demonstrates clearly that significantly lower VI values (relative to the characteristic size of fluctuations) are achieved below some value of δ . It is difficult to estimate this threshold value of δ , but an empirical plot like Fig. 5 can provide insights into the range of δ values to which the detection can be restricted. A possible heuristic, that works well in practice, is choosing values of δ lower than the point at which the average loss in modularity roughly equals the average estrangement. However, this is an ad-hoc prescription and a limitation of our method is that the desired smoothness is not

determined a priori. Similar difficulties are also inherent in several other methods^{12,14,16,17}.

We now compare our method with other known methods on a series of synthetic networks generated by varying p_c and p which corresponds to varying the density of edges inside a community and the rate at which they evolve, respectively. As shown in Fig. 6, our method consistently detects a temporal community structure that is most similar to the ground truth as compared to those found by the multislice modularity method¹² and independent modularity maximization in each snapshot along with label-mapping. For our method we pick the minimum value of VI that is achieved as δ is varied between 0 and 0.1. A minimum of VI is usually attained at a value of δ between 0.0 and 0.05. For the multislice method we pick the minimum VI achieved by varying the inter-slice coupling ω between 0.05 and 1. We find that multislice modularity method finds the two communities, but is less adept at detecting the temporal variation, i.e., the birth and death of the large temporal communities for even small values of ω . Furthermore, for even marginally high values of ω , (e.g. 0.2), it finds large spurious temporal communities. The performance of all three methods improves with increase in intra-community edge density. The rate of change, p , has a noticeable effect on performance only for low values of p_c .

Having shown the performance of our method on a range of synthetic benchmarks, we next turn to the analysis of a real network: the human contact network data provided by the Reality-mining project²¹ which tracked the mobility of about hundred individuals over nine months. A contact is registered when the Bluetooth devices being carried by the individuals come within 10 m of each other. The evolution chart in Fig. 7 shows the temporal communities resulting from applying estrangement confinement to snapshots created by aggregating contacts between individuals over a week (except over

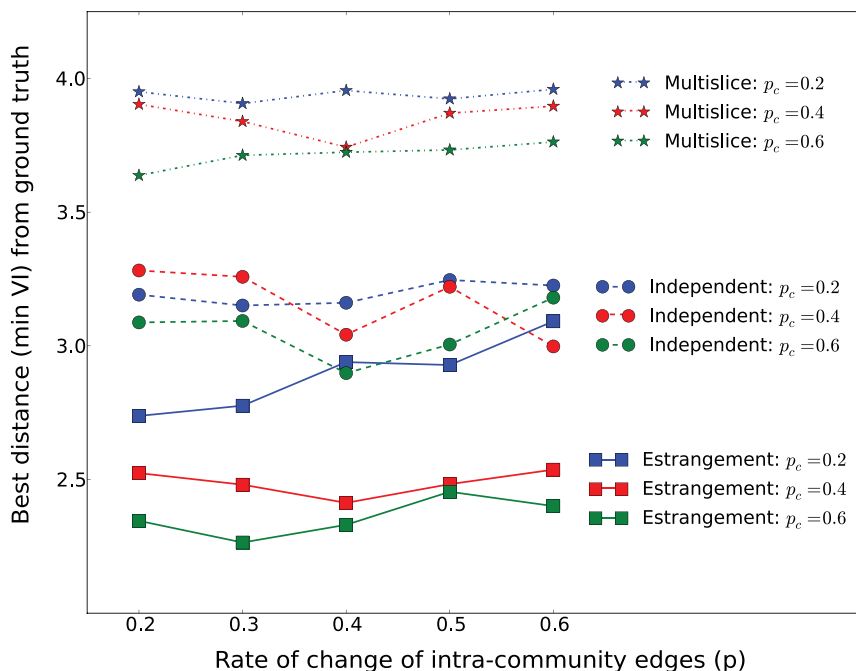


Figure 6 | Estrangement confined modularity maximization (curves labeled Estrangement) consistently finds a temporal community structure that is most similar to the ground truth temporal community structure as compared to multislice modularity maximization (curves labeled Multislice) and independent modularity maximization within each snapshot (curves labeled Independent). Note that even for independent modularity maximization, we perform the label mapping step between successive partitions. The distance between the two temporal partitions is measured using VI (see text). The comparison is done on a range of evolving synthetic networks (constructed as described in Fig. 3(b)) obtained by varying the rate of change of intra-community edges, p , for different values of intra-community edge density, p_c . For each benchmark we show the minimum VI obtained by Estrangement as the constraint δ is varied and similarly the minimum VI obtained by Multislice as the inter-slice coupling is varied.

vacation weeks in December) thus creating a weighted time evolving network, where in each snapshot the weight on an edge represents the number of contacts between the corresponding individuals. The nodes are ordered on the Y axis by the tuple of labels they take over time, where the labels in the tuple itself are ordered by the frequency of acquiring that label. Ties are broken by the time of first appearance of nodes. This ordering causes the nodes in a temporal community to appear contiguously. We illustrate communities and events that can be correlated with ground truth in Fig. 7.

Finally, we analyze a time-evolving weighted network consisting of United States senators where the weight on an edge represents the similarity of their roll call voting behavior. The data was obtained from *voteview.com* and the similarities between a pair of senators was computed following Waugh *et al.*²² as the number of bills on which they voted similarly, normalized by the number of bills they both voted on. The network consists of 111 snapshots corresponding to congresses over 220 years and 1916 unique senators. In Fig. 8, we show the evolution chart for $\delta = 0.05$, the value at which loss in modularity roughly equals the gain in estrangement.

A broad feature that is observed for all values of temporal divergence is the emergence of two dominant voting communities with time. The party affiliation of the majority of the constituent nodes within these communities allows us to identify them as the temporal streams which culminate in the present day Democratic and Republican parties (Fig. 8(a)). These features were previously observed by Mucha *et al.*¹². However, in contrast to their method, ours is sequential and does not need to construct and analyze the stacked network comprising of all snapshots. In addition to the dominant Democratic and Republican streams, we also detect two minor communities that consist of senators who predominantly vote in alignment with one of the two dominant communities, but have occasional switches to the other. One of these detected minor communities consists predominantly of Democratic members of the *conservative*

coalition (Fig. 8(b)). The second minor community found consists of several moderate Democrats and left-leaning Republicans (Fig. 8(c)).

Another feature we find is the reduction with time in the number of senators whose aggregating voting behavior over the duration of a congress are not aligned with the rest of their party. Fig. 8(d) shows the number of such “atypical” senators over time. Notice that after the year 1995, there is only one such senator detected by our method, whereas prior to 1995, a much larger number of senators voted differently from the bulk of their party.

Discussion

We have presented a novel approach to detect temporal communities based on a constrained optimization formulation. A critical piece of the formulation is the definition of estrangement, an effective measure of partition distance between snapshots of a time-varying network that is motivated by the tendency of nodes to maintain similarity of community affiliations with their neighbors. The constraint on estrangement allows us to pick solutions from the highly degenerate and sensitive modularity landscape that maintain temporal contiguity without compromising the current community structure. Our solution technique using Lagrangian duality relies on the fact that estrangement can be decomposed into local, single node terms. Our method operates on one snapshot at a time thus allowing us to compute temporal communities in a sequential manner, which is particularly useful for large networks. Notably, even if all snapshots are available to us in advance, estrangement provides a non-trivial but intuitive control parameter using which a broad range of temporal smoothness can be probed, potentially enabling community discovery on many temporal scales. We demonstrate that meaningful temporal communities can be found by estrangement constrained modularity maximization. In particular, our demonstrations on empirical networks are corroborated by available ground

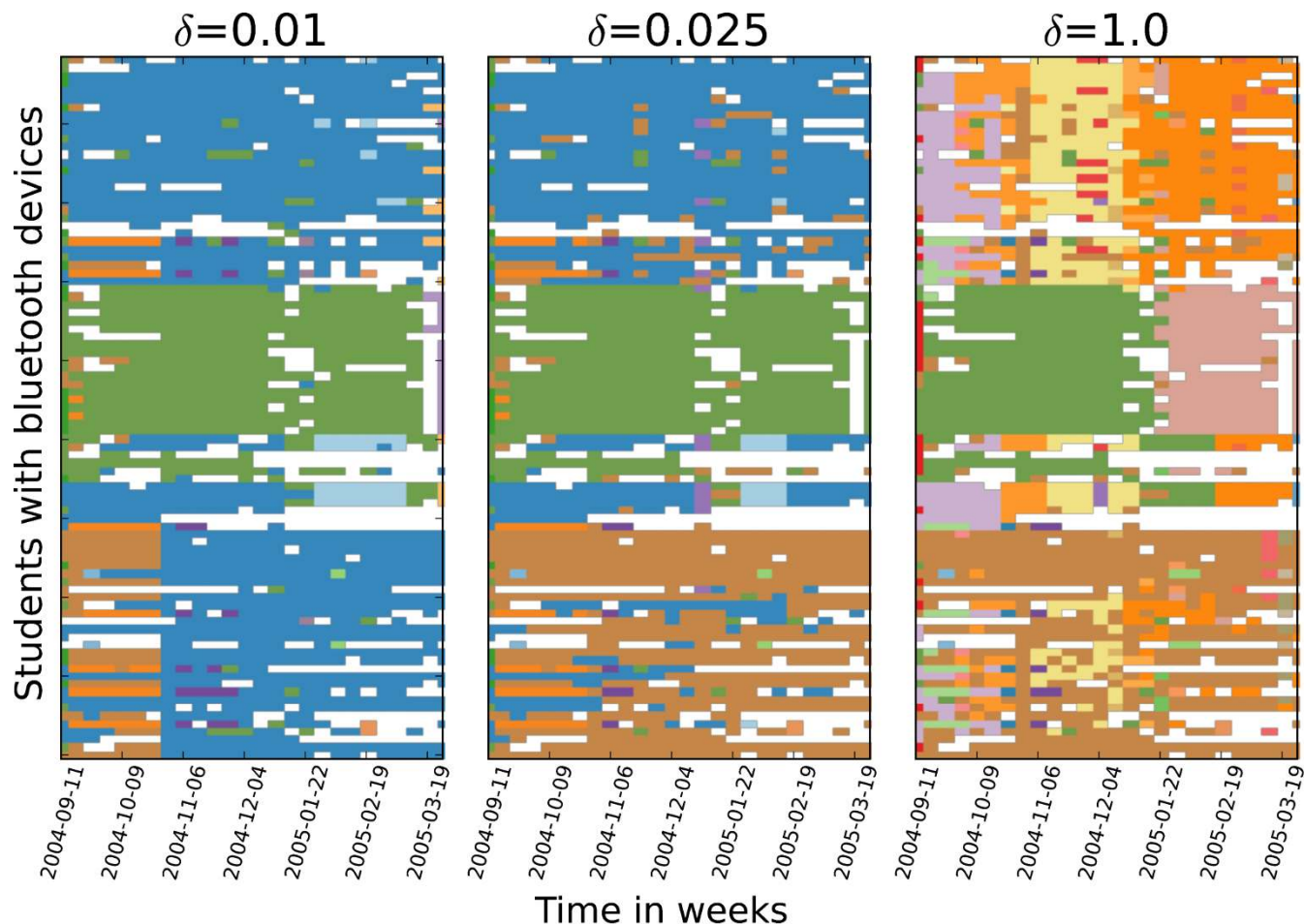


Figure 7 | Temporal communities seen in the reality mining network²¹. The network consists of two communities predominantly at lower values of δ , one corresponding to staff and students at the MIT Media Lab (blue), and the other corresponding to students at the MIT Sloan School of Business (green). As δ is increased the average size of the communities decreases and the number of communities increases, a consequence of the decreasing temporal contiguity.

truth and by previous studies which used non-sequential methods to discover temporal communities.

Several issues are worthy of further study. A limitation of our method is that it does not provide a specific prescription for choosing values of the constraint δ that lead to meaningful temporal communities. Such a prescription will improve the utility of the method in practice. Another important issue is determining the granularity at which the time varying network is snapshotted. If the snapshots are made too frequently, there may not be enough density of edges to discover communities, whereas aggregating for too much time may prevent detection of some evolving patterns. In this work, we assume that there is a natural timescale of interest for creating snapshots, such as the one defined by biennial congressional elections in the case of the senator voting similarity network. In general, such natural timescales can perhaps be found by analyzing the frequency spectrum of some relevant variable in the dataset²¹. A related issue is that of sporadic interruptions in data collection which could affect the calculation of estrangement as well as the mapping of communities between snapshots. The effect of interruptions can be mitigated by using a history of the extent to which nodes share community affiliations to compute estrangement. Also, estrangement is generalizable to the case of overlapping communities (see SI) which could reveal further interesting features in community evolution.

Methods

We describe our Lagrangian duality based method for estrangement constrained optimization of modularity. As summarized in *Results*, we first need to compute the

Lagrange dual function (Eq. 4), which we show can be computed by adapting known methods for unconstrained modularity maximization. The key to computing the dual lies in exploiting the property that estrangement is decomposable, similarly to modularity, into single node (or local) contributions. We utilize a hierarchical version of the Label Propagation Algorithm¹⁰ to compute the dual. This method, which we refer to as HPLA, works by greedily merging communities that provide the largest gain in the objective function, and then repeating the procedure on an *induced* graph in which the communities from the previous steps are the nodes. Once this method of computing the Lagrange dual has been determined, we solve the dual problem of finding the best Lagrange multiplier by using Brent's method²⁴ which is commonly used for non-differentiable objective functions. We now present the above steps in greater detail.

HPLA update rule for computing the Lagrange dual. We compute the Lagrange dual $g(\lambda)$ for a given λ , using HPLA in which each node x updates its community identifier (l_x) following the rule:

$$l_x = \arg \max \left(N_{xl} - \frac{k_x K_l}{2M} + \frac{k_x^2}{2M} \delta(l_x, l) + \lambda Q_{xl} \right), \quad (7)$$

where $N_{xl} = \sum_u A_{ux} \delta(l_u, l)$, $O_{xl} = \sum_{u \neq x} Z_{ux} \delta(l_u, l)$ and $K_l = \sum_u k_u \delta(l_u, l)$. Here O_{xl} is the extra term that arises due to the constraint on E . Next we show that this update rule indeed performs a greedy maximization of the Lagrangian. Following Barber and Clark¹⁰, we expand Q and write $L(\mathcal{P}, \lambda)$ as:

$$L(\mathcal{P}, \lambda) = \frac{1}{2M} \left(\sum_u \sum_v \left(A_{uv} - \frac{1}{2M} k_u k_v + \lambda Z_{uv} \right) \delta(l_u, l_v) \right) \quad (8)$$

Here we have taken advantage of the fact that the first term $\frac{\sum_{uv} Z_{uv}}{2M}$ in E (Eq. 3) is independent of the partition and does not affect the optimization. To see the effect of a label update for a single node x , we separate terms of Eq. 8 into contributions from x and those from all other nodes. Doing so yields:

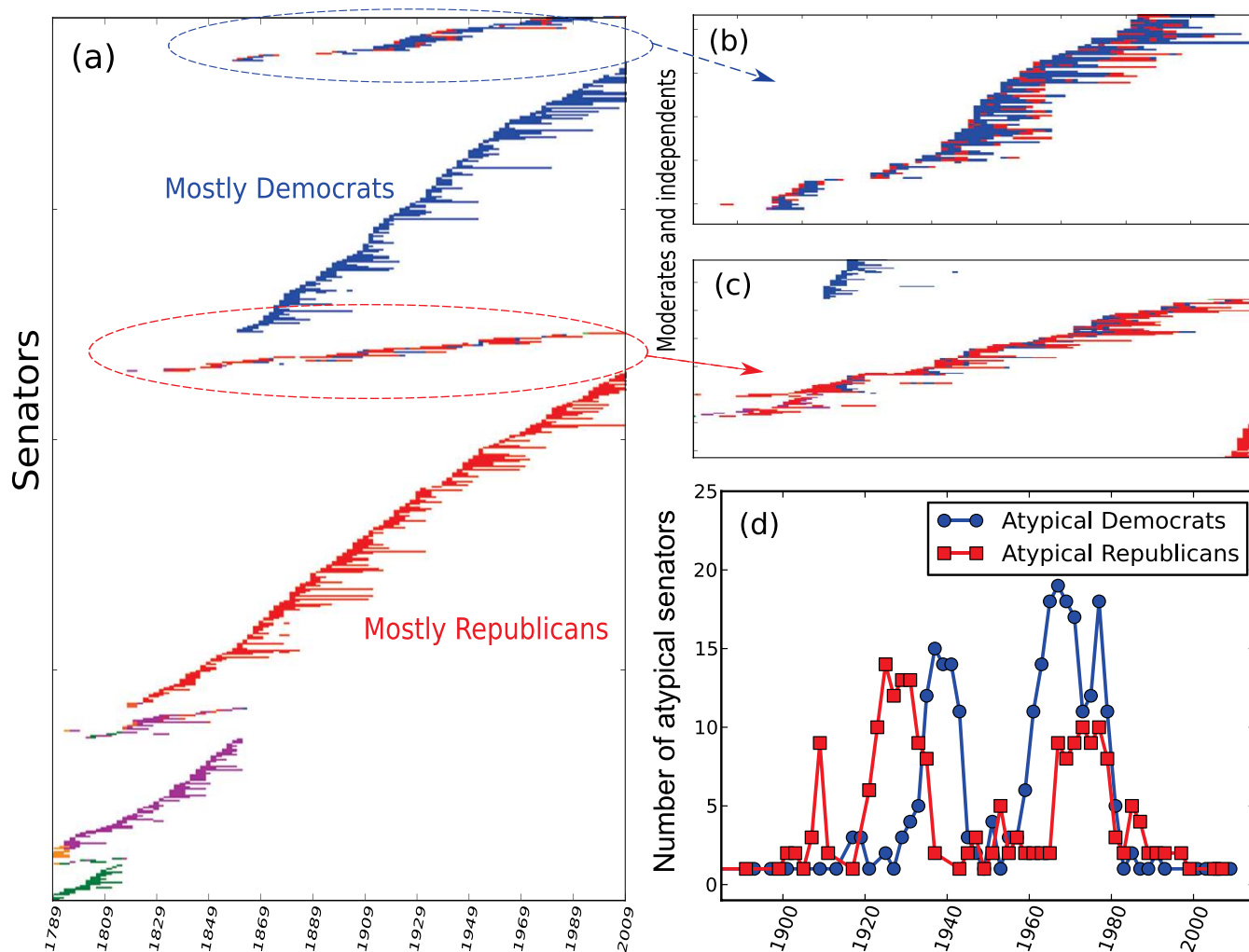


Figure 8 | The different temporal communities observed in the senator voting similarity network. (a) The dominant temporal communities can be identified with the modern day Republican (in red) and Democratic (in blue) parties. (b) and (c) Minor communities are also found, which consist of republicans and democrats who often differ in their voting behavior from the majority membership of their party, as well as independents. (d) The number of atypical senators defined as those whose temporal communities are different from their declared party affiliations. Our results indicate that the number of such senators has been decreasing with time, perhaps indicating increasing polarization.

$$\begin{aligned}
 L = & \frac{1}{2M} \left(\sum_{u \neq x} \sum_{v \neq x} \left(A_{uv} - \frac{1}{2M} k_u k_v + \lambda Z_{uv} \right) \delta(l_u, l_v) \right) \\
 & - \frac{1}{2M} \left(A_{xx} - \frac{1}{2M} k_x^2 + \lambda Z_{xx} \right) \\
 & + \frac{1}{2M} \left(2 \sum_u \left(A_{ux} - \frac{1}{2M} k_u k_x + \lambda Z_{ux} \right) \right) \delta(l_u, l_x)
 \end{aligned} \tag{9}$$

where in the interest of brevity, we have introduced the shortened notation L to mean $L(\mathcal{P}, \lambda)$. The first two terms in L (R.H.S. of Eq. 9) are unaffected by the label update of node x , so we focus on the last term. Since our goal is to greedily optimize L via label updates, we update the label of node x to one that results in the maximal gain in L . Thus, the desired post-update label is:

$$\begin{aligned}
 l_x = & \arg \max_l \frac{1}{2M} \left(2 \sum_u \left(\left(A_{ux} - \frac{1}{2M} k_u k_x + \lambda Z_{ux} \right) \right) \delta(l_u, l) \right) \\
 = & \arg \max_l \left(N_{xl} - \sum_{ux} \frac{1}{2M} k_u k_x \delta(l_u, l) + \sum_{ux} Z_{ux} \delta(l_u, l) \right)
 \end{aligned} \tag{10}$$

where we have used the fact that $\sum_u A_{ux} \delta(l_u, l)$ is simply the number of neighbors of x with label l , which we denote by N_{xl} . The diagonal terms (i.e. terms with $u = x$) in the remaining sums of the above equation do not have any bearing on the maximization and can be ignored. Then, using:

$$\sum_{u \neq x} \frac{1}{2M} k_u k_x \delta(l_u, l) = \frac{k_x K_l}{2M} - \frac{k_x^2}{2M} \delta(l_x, l)$$

and writing $\sum_{u \neq x} Z_{ux} \delta(l_u, l)$ as O_{xl} (also, $K_l = \sum_u k_u \delta(l_u, l)$), we see that Eq. 10 reduces to Eq. 7. It follows that the HPLA label update rule maximizes the gain in L . The optimization of L in HPLA is further improved by adopting an additional hierarchical step after the labels have converged to a local maximum of L . We detail this hierarchical procedure below.

Hierarchical procedure in HPLA for computing the Lagrange dual. Once the sequence of label updates has converged on the original graph on which L is being maximized, we build a new *induced graph* which contains the communities of the original graph as nodes. Links between pairs of nodes in the new graph have weights equal to the total number of links between the two communities in the original graph that they correspond to. Then, L can be further increased by updating the labels of nodes in the induced graph iteratively, following Eq. 6. Importantly, this is possible only because L remains invariant in the transformation from the original graph to the induced graph (see SI). This alternating procedure of label updates followed by the induced graph transformation is recursively applied until we reach a hierarchical level where the converged value of L is lower than that obtained at the previous level. The partition found at the penultimate level before termination is chosen as the one optimizing L . This hierarchical procedure for optimizing L is similar in spirit to the one used in the Louvain algorithm²³ for optimizing Q .

Details on solving the dual problem. Having found a way to compute $g(\lambda)$ we can solve the dual problem and determine the value of λ at which $g(\lambda)$ is minimized. The challenge here is that $g(\lambda)$ is not differentiable and moreover, it is expensive to evaluate. We use Brent’s method which is often used to optimize non-differentiable



scalar functions within a given interval. In our case, $g(\lambda)$ is the scalar function, and we minimize it within a suitably large range of λ . We use an implementation provided by python's scientific library, SciPy, in the form of `scipy.optimize.fminbound()`. For all experiments in this work, $\lambda_{min} = 0$ and $\lambda_{max} = 10$.

Furthermore, to mitigate issues due to the local nature of the algorithm and the degeneracy of the modularity landscape (and therefore the $L(P, \lambda)$ landscape), we perform several independent runs of HPLPA for a given λ and pick the run which yields the highest value of $g(\lambda)$. We perform at least 10 runs of HPLPA as we start Brent's method and increase the number of runs by 10 with every iteration that narrows the search interval for λ . Near the optimum value of λ , we perform at least 150 runs to compute the Lagrange dual. For the synthetic benchmarks we increase the number of runs by 5 at every iteration. Once we identify the value $\lambda = \lambda^*$ for which $g(\lambda)$ is minimized, the partition which yielded $g(\lambda^*)$, from among the many independent runs for $\lambda = \lambda^*$, is chosen as the optimal partition for the given snapshot. In practice, due to the degeneracy of the $L(P, \lambda)$ landscape for any λ , we have to go slightly above λ^* to ensure that the optimal partition lies within the feasible region.

Our implementation is available at <https://github.com/kawadia/estrangement>.

Comparing detected temporal communities with those in ground-truth:

Variation of Information. We utilize Variation of Information (VI)([5]) to quantify how far the temporal community partitions detected by the algorithms - estrangement confinement, multislice modularity maximization, independent modularity maximization - are from those that exist in the ground-truth. Given partitions P and P' of the set of {node,time} pairs, the VI between them is defined as:

$$VI(P, P') = -\frac{1}{n} \sum_{i \in P, j \in P'} n_{ij} \log \left(\frac{n_{ij}^2}{n_i n_j} \right)$$

where n is the total number of {node,time} pairs, n_i and n_j denote the number of {node,time} pairs in the temporal community i in P and the temporal community j in P' respectively, and n_{ij} is the number of nodes common to both i in P and j in P' .

The ground truth partition for our synthetic networks consists of two large temporal communities defined by the subsets of nodes having higher edge density and undergoing markovian evolution (Fig. 3). Each remaining {node,time} pair (which is not part of either temporal community) is assumed to be a temporal community by itself. The latter is perhaps an extreme assumption, but necessitated by the difficulty of appropriately defining ground truth communities within subsequent random instantiations of an Erdős-Rényi network. To alleviate the punitive nature of this definition, and to account for the fact that even within random graphs, communities consisting of more than one node may exist, we only consider those community pairs in the evaluation of VI for which at least one of the communities is of size greater than ten nodes. Thus, small communities of size greater than one but less than ten detected within the random background do not penalize VI despite not exactly corresponding to the ground truth.

For purposes of comparison, we also run the multislice modularity maximization algorithm on the synthetic networks. This was done using code publicly available at: <http://netwiki.amath.unc.edu/GenLouvain/GenLouvain>. The results shown in Fig. 6 are for the temporal community partitions with the lowest VI from the ground-truth obtained over values of $\omega = 0.05, 0.1, 0.2, 0.4, 0.6, 0.8, 1.0$, with 50 independent runs for each value of ω .

- Fortunato, S. Community detection in graphs. *Physics Reports* **486**, 75–174 (2010).
- Rosvall, M. & Bergstrom, C. T. Mapping change in large networks. *PLoS ONE* **5**, e8694 (2010).
- Palla, G., Barabási, A. & Vicsek, T. Quantifying social group evolution. *Nature* **446**, 664–667 (2007).
- Newman, M. E. J. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences* **103**, 8577–8582 (2006).
- Good, B. H., de Montjoye, Y.-A. & Clauset, A. Performance of modularity maximization in practical contexts. *Physical Review E* **81**, 046106 (2010).
- Karrer, B., Levina, E. & Newman, M. E. J. Robustness of community structure in networks. *Phys. Rev. E* **77**, 046119 (2008).
- Hannan, M. T. & Freeman, J. Structural inertia and organizational change. *American Sociological Review* **49**, 149–164 (1984).
- Ramasco, J. J. & Morris, S. A. Social inertia in collaboration networks. *Phys. Rev. E* **73**, 016122 (2006).

- Brandes, U. *et al.* On modularity-np-completeness and beyond. *ITI Wagner, Faculty of Informatics, Universität Karlsruhe (TH), Tech. Rep* **19**, 2006 (2006).
- Barber, M. J. & Clark, J. W. Detecting network communities by propagating labels under constraints. *Phys. Rev. E* **80**, 026129 (2009).
- Raghavan, U. N., Albert, R. & Kumara, S. Near linear time algorithm to detect community structure in large-scale networks. *Physical Review E* **76**, 036106 (2007).
- Mucha, P., Richardson, T., Macon, K., Porter, M. & Onnela, J. Community structure in time-dependent, multiscale, and multiplex networks. *Science* **328**, 876–878 (2010).
- Lambiotte, R., Delvenne, J.-C. & Barahona, M. Laplacian dynamics and multiscale modular structure in networks. arXiv:0812.1770v3 (2010).
- Chakrabarti, D., Kumar, R. & Tomkins, A. Evolutionary clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, 554–560 (2006).
- Sun, J., Faloutsos, C., Papadimitriou, S. & Yu, P. S. Graphscope: parameter-free mining of large time-evolving graphs. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, 687–696 (2007).
- Lin, Y.-R., Chi, Y., Zhu, S., Sundaram, H. & Tseng, B. L. Facetnet: a framework for analyzing communities and their evolutions in dynamic networks. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, 685–694 (2008).
- Lin, Y.-R., Sun, J., Sundaram, H., Kelliher, A., Castro, P. & Konuru, R. Community discovery via metagraph factorization. *ACM Trans. Knowl. Discov. Data* **5**, 17:1–17:44 (2011).
- Breiman, L. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science* **16**, 199–231 (2001).
- Newman, M. E. J. Analysis of weighted networks. *Phys. Rev. E* **70**, 056131 (2004).
- Boyd, S. & Vandenberghe, L. *Convex Optimization* (Cambridge University Press, 2004).
- Eagle, N. & (Sandy) Pentland, A. Reality mining: sensing complex social systems. *Personal Ubiquitous Comput.* **10**, 255–268 (2006).
- Waugh, A., Pei, L., Fowler, J., Mucha, P. & Porter, M. Party polarization in congress: A network science approach. arXiv:0907.3509 (2010).
- Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *Journal of Stat Mech.: Theory and Experiment* **2008**, P10008 (2008).
- Brent, R. *Algorithms for minimization without derivatives* (Dover Publications, 2002).

Acknowledgments

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions are those of the authors and not of the sponsors.

We thank Stephen Dabideen for testing our implementation, adding documentation and preparing it for release.

Author contributions

VK and SS performed the research and prepared the manuscript. VK implemented the method and analyzed the data.

Additional information

Supplementary information accompanies this paper at <http://www.nature.com/scientificreports>

Competing financial interests: The authors declare no competing financial interests.

License: This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivative Works 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-nd/3.0/>

How to cite this article: Kawadia, V. & Sreenivasan, S. Sequential detection of temporal communities by estrangement confinement. *Sci. Rep.* **2**, 794; DOI:10.1038/srep00794 (2012).