Dynamic Consensus Community Detection and Combinatorial Multi-Armed Bandit

Domenico Mandaglio
DIMES - University of Calabria, Rende (CS), Italy
d.mandaglio@dimes.unical.it

Andrea Tagarelli
DIMES - University of Calabria, Rende (CS), Italy
andrea.tagarelli@unical.it

***Abstract*—Community detection and evolution has been largely studied in the last few years, especially for network systems that are inherently dynamic and undergo different types of changes in their structure and organization in communities. Because of the inherent uncertainty and dynamicity in such network systems, we argue that temporal community detection problems can profitably be solved under a particular class of multi-armed bandit problems, namely combinatorial multi-armed bandit (CMAB). More specifically, we propose a CMAB-based methodology for the novel problem of dynamic consensus community detection, i.e., to compute a single community structure that is designed to encompass the whole information available in the sequence of observed temporal snapshots of a network in order to be representative of the knowledge available from community structures at the different time steps. Unlike existing approaches, our key idea is to produce a dynamic consensus solution for a temporal network to have unique capability of embedding both long-term changes in the community formation and newly observed community structures.**

***Index Terms*—Community detection in temporal networks, Multi-armed bandit problems, Complex network models**

I. Introduction

The problem of identifying the community behavior at any given time is often jointly considered with the need for modeling the change events in the communities and tracking their evolution [5]. While there exist various models for time-varying network data (i.e., series of snapshots, interval graphs, or interactions), detecting, monitoring and correlating the events of community evolution is particularly challenging. In this context, one issue is related to making an appropriate choice of timestep width that can provide sufficient resolution to detect temporal events. An even bigger issue is that the community evolution events are of different type (e.g., birth/death, growth/decay, merge/split), and may occur at different rates (i.e., smoothly or drastically, at varying degrees).

Despite the variety of methodologies developed for the community detection and evolution problem, each of the existing approaches is designed to address a limited subset of challenges by adopting a particular perspective on the problem [9]. Some methods provide heuristics that try to discover a sequence of mappings for the community structures independently derived at each time step (e.g., [10], [2], [20]); by contrast other methods aim to detect a community structure for the current topology as dependent on the structure(s) from prior time step(s), according to some parameter models to control the temporal smoothness (e.g., [12], [8], [21], [22]). Further strategies include updating a community structure in order to reflect newly observed changes [23], [17], [1]), or aggregating the various snapshots of the network in order to enable a static community detection method (e.g., [16]).

All the aforementioned approaches nonetheless share the nature of graph-based unsupervised learning paradigm to address the community detection problem. However, this may not be in principle the best way to do, primarily because of the inherent uncertainty about the *environment*, i.e., the temporal network system, and the *interactions* within it, i.e., structural changes and consequent decisions to take about the node memberships and structure of the communities. Within this view, *reinforcement learning* is instead conceived to learn from interrelated actions with unknown "rewards" ahead of time, and choose which actions to take in order to maximize the reward. A further key aspect is to achieve a trade-off between making decisions that yield high current rewards, or *exploitation*, and making decisions that discard immediate gains in favor of better future rewards, or *exploration*.

Multi-armed bandit (MAB) problems are well-suited to model the aforementioned trade-off [18], [14]. However, they cannot be directly applied to our problem since they deal with individual actions to take at any time. In this work, we focus on a particular extension of MAB, called *combinatorial multi-armed bandit* (CMAB) [3], [7], to deal with choosing *a set of* actions, i.e., a set of community assignments that constitute a whole community-structure. Moreover, the exploration-exploitation trade-off would correspond to *balancing over time between the need for embedding long-term changes observed in the community formation and the need for capturing short-term effects and newly observed community structures*.

In this regard, we devise a solution to the problem of community detection in a temporal network by introducing the novel concept of *dynamic consensus community structure*, that is, loosely speaking, a community structure that encompasses the knowledge about newly observed as well as the previously

ASONAM '19, August 27-30, 2019, Vancouver, Canada

ACM ISBN 978-1-4503-6868-1/19/08
http://dx.doi.org/10.1145/3341161.3342910

detected communities in a temporal network. Surprisingly, little research has been conducted on the temporal counterpart of the *consensus community detection* problem [15], [19]. One main issue is that the consensus community structure is to be inferred from a knowledge base (i.e., set of community structures) that is not fully available at a given initial time, but it evolves over time along with the associated temporal network. In [6], a representative clustering solution is determined by aggregation of multiple runs of an MCMC algorithm; however, the approach is restricted to dynamic stochastic block model graphs, and focuses on some dynamics of community only (i.e., birth, death, split, merge). In [13], the common structure in the snapshots of a temporal network is studied based on the optimization of a function incorporating Markov steady-state matrices, similarity matrices and community membership matrices; however, the approach assumes there are same nodes and number of communities for each snapshot (resp. slice) of the temporal (resp. multiplex) network. Inter-snapshot relationships are captured in [4] by grouping nodes based on topological similarity, instead of structural equivalence.

We formulate the novel problem of dynamic consensus community detection in time-evolving networks, and originally propose a general algorithmic scheme based on the CMAB paradigm. One important feature of our approach is that, unlike existing ones, it does not require to match and/or track the evolution of communities over time, and it does not depend on specific community-change events or on restricted graph models. Our proposed algorithmic scheme is conceived to be versatile in terms of the bandit strategy as well as in terms of the static community detection algorithm used to identify the communities at each snapshot network.

II. Problem Statement

We are given a set $\mathcal{V}$ of *entities* (i.e., users) in a social environment, and a *temporal network* $\mathcal{G}$ as a series of graphs over discrete time steps $(G_1, G_2, \ldots, G_t, \ldots)$, where $G_t = \langle V_t, E_t \rangle$ is the graph at time t, with set of nodes V_t and set of undirected edges E_t. We denote with $\mathcal{G}_{\leq t}$ a *series of graphs observed until time* t. Each node in V_t corresponds to a specific instance from the set $\mathcal{V}_t \subseteq \mathcal{V}$ of entities that occur at time t. The snapshot graphs can share different subsets of entities.

Given any G_t, we denote with $\mathcal{C}^{(t)}$ a *community structure* for G_t, which is a set of non-overlapping communities, and is assumed to be unrelated to any other $\mathcal{C}^{(t')}$ $(t' \neq t)$, both in terms of number of communities and set of entities involved. We will use the term *dynamic ensemble* at time t, to refer to a set of community structures *incrementally* provided along with the snapshot graphs observed until time t, and we denote it as $\mathcal{E}_{\leq t} = \{\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(t)}\}$. We consider the following problem:

> **Input:** Given the temporal graph sequence $\mathcal{G}_{\leq t}$ and associated dynamic ensemble $\mathcal{E}_{\leq t}$, for any time $t \geq 1$,
>
> **Goal:** Compute a community structure, called *dynamic consensus community structure* and denoted as $\mathcal{C}^*_{\leq t}$, which is designed to encompass the information from $\mathcal{G}_{\leq t}$ to be representative of the knowledge available in $\mathcal{E}_{\leq t}$.

Given $\mathcal{G}_{\leq t}$ and $\mathcal{E}_{\leq t}$, the representation model underlying the dynamic consensus being discovered over time is a matrix $\mathbf{M}$ we call ***dynamic co-association*** (or ***consensus***) ***matrix*** (DCM). The size of this matrix is initially $\mathcal{V}_t \times \mathcal{V}_t$ with $t = 1$, and at a generic time t is $|\mathcal{V}| \times |\mathcal{V}|$. The (i, j)-th entry of $\mathbf{M}$, denoted as m_{ij}, stores the probability of co-association for entities $v_i, v_j \in \mathcal{V}$, i.e., the probability that v_i and v_j are assigned to the same community, in the observed timespan.

Computing meaningful co-associations for the nodes in the temporal network and properly maintaining and updating the consensus community structure over time is challenging. On the one hand, we want to avoid (re)computation of the consensus from scratch, e.g., from a predetermined, finite set of community structures as in conventional consensus community detection [15], [19]; on the other hand, we also do not want to depend on any mechanism of tracking of the evolution of communities [5]. More importantly, the dynamic consensus community structure should be able to embed long-term changes in the community formation as well as to capture short-term effects and newly observed community structures.

To address the above problem, we adopt a perspective that is different from the typical unsupervised learning approach to community detection problems. We argue that the dynamic consensus community detection problem is well-suited to be solved under a *reinforcement learning* (RL) framework [18]. If we interpret the decisions to learn about the assignments of nodes to communities as *interrelated actions*, with *unknown* rewards ahead of time, then it emerges the need for learning which actions to take in order to maximize a reward, which is related to how much benefit is gained by node assignments to communities. By learning from interactions, RL becomes particularly useful when there is uncertainty in the learning environment: this clearly holds in our setting due to the dynamics of the network, the evolution of its structural changes, and consequent effect on the community structure.

Actions affect not only the immediate reward, but also the next step in taking actions, and so the subsequent rewards. Thus, a further key aspect in our problem is the dilemma between "exploitation", i.e., making decisions that yield high current rewards, vs. "exploration", i.e., making decisions that sacrifice current gains with the prospect of better future rewards. *Multi-armed bandit* (MAB) refers to a class of stochastic resource allocation problems in the presence of alternative (competing) choices, that are paradigms of the exploration-exploitation trade-off. In this work, we focus on a particular class of MAB problems, called *combinatorial multi-armed bandit* (CMAB), whose distinguishing key is in the need for choosing *a set of* actions at any time.

A. Background on combinatorial multi-armed bandit (CMAB)

In the traditional MAB framework, there exists a set of m *arms*, associated with a set of random variables $\{X_{i,t} \mid 1 \leq i \leq m, t \geq 1\}$, whose values range in $[0, 1]$. $X_{i,t}$ indicates the random outcome of *triggering*, or playing, the i-th arm in the t-th round. The random variables $\{X_{i,t} \mid t \geq 1\}$ associated to the i-th arm are independent and identically distributed. Moreover, in a *non-stationary context*, those variables may change [11]. Also, variables of different arms may not be independent.

CMAB is an extension of MAB that introduces the concept of *superarm* as a set of (base) arms that can be triggered together [3], [7]. At each round t, a superarm A is chosen and the outcomes of the random variables $X_{i,t}$, for all $a_i \in A$, are revealed. Moreover, the base arms belonging to A may probabilistically trigger other base arms not in A, thus revealing their associated outcomes. Let $R_t(A)$ be a random variable denoting the *reward* obtained at round t by playing superarm A. This reward depends, linearly or non-linearly, on the base arms that constitute the superarm and other possibly triggered base arms. The objective of a CMAB method is to select at each round t the superarm A that maximizes the *expected reward* $\mathbb{E}[R_t(A)]$, in order to eventually maximize the *cumulative expected reward* over all rounds. According to the exploration-exploitation trade-off, at each trial the bandit may decide to choose the superarm with the highest expected reward (given the current mean estimates for the base arms) or to select a superarm discarding information from earlier rounds with the aim of discovering the benefit from adopting some previously unexplored arm(s) [7], [3].

III. Translating the problem of dynamic consensus community structure into CMAB

In our context, each pair of entities $\langle v_i, v_j \rangle$ in $\mathcal{G}_{\leq t}$ is hypothetically associated with an unknown distribution (with unknown mean μ_{ij}) for the probabilities of co-association over time, whose mean estimate is the entry m_{ij} in DCM. Each observation of a community structure of a snapshot network, can be considered as a sample from such distributions. Moreover, these may change their mean over time, thus our CMAB setting is non-stationary (cf. Sect. II-A): in fact, for groups of entities which tend to maintain their membership to stable communities over time, we will observe a similar degree of co-association between pairs of entities belonging to the same, stable community; however, in general, the network structure along with the communities is subjected to several changes.

Each pair of entities $\langle v_i, v_j \rangle$ corresponds to a base arm, whose semantics is "*to assign v_i and v_j to the same community at a given time*". We will use symbol $c_i^{(t)}$ to denote the community of v_i at round t. A superarm A at round t is a set of arms, i.e., a set of pairs $\langle v_i, v_j \rangle$ such that $c_i^{(t)} = c_j^{(t)}$.

Playing a superarm A at each round t corresponds to a two-stage process: (i) inducing a community structure from the played superarm and (ii) performing stochastic relocation of nodes to neighbor communities. The stochastic nature of the process depends on both the random order with which we consider the node relocations and on the fact that, according to the optimization of a quality criterion, an improvement due to relocation is accepted with a certain probability. Intuitively, this allows us to account for uncertainty in the long-term overall quality improvement of the consensus due to local relocations at a given time; for instance, it is unknown if the relation that explains two users share the same community at a given time could become meaningless in subsequent times.

Algorithm 1 General scheme of CMAB algorithm for Dynamic Consensus Community Detection

Input: Temporal graph sequence $\mathcal{G}_{\leq T}$ ($T \geq 1$), bandit strategy $\mathcal{B}$, (static) community detection method $\mathcal{A}$.
Output: Dynamic consensus community structure $\mathcal{C}^*_{\leq T}$.

```
1:  Initialize the dynamic consensus matrix M
2:  for t = 1 to T do
3:      if B decides for EXPLORATION then
4:          Find a community structure C^(t) on G_t using A
5:      else {EXPLOITATION}
6:          Partition the DCM-graph using A
7:          Infer a community structure C^(t) on G_t based on the DCM-graph partitioning
8:      end if
9:      Project the community memberships from C^(t) onto G_{≤t}
10:     Stochastic optimization of C*_{≤t}
11:     Update the DCM matrix M based on C*_{≤t}
12: end for
13: return C*_{≤T}
```

After playing a superarm A, the rewards associated to the entity pairs (base arms) corresponding to the status of communities after the relocation phase, are revealed; these pairs include both the nodes that did not move from their community and the arms $\langle v_i, v_j \rangle$ triggered with the accepted relocations, i.e., such that node v_i was moved to the community of v_j. Furthermore, for the base arms that were neither selected nor triggered (i.e., pairs of nodes that were not in the same community before and after the relocation phase), we assume an implicit reward of zero that corresponds to the observation of the "no-coassociation" event. (This is in line with the possibility in CMAB of enabling the probabilistic triggering of *all* base arms.) The reward of a superarm corresponds to the *quality* of the community structure at the end of the relocation phase, which is a non-linear function of the base arms' rewards. More specifically, we might resort to *modularity* as quality criterion for a community structure.

IV. An algorithmic scheme for the CMAB-based Dynamic Consensus Community Detection problem

To solve the dynamic consensus community detection problem, we propose the general scheme in Algorithm 1.

This starts with the initialization of the dynamic consensus matrix $\mathbf{M}$ as an identity matrix (Line 1); in fact, at the initial time, no information has been processed yet, and hence each entity-node has co-association with itself only.

At each round t, the algorithm chooses to perform either exploration or exploitation, according to a given bandit strategy ($\mathcal{B}$). Intuitively, in the exploitation phase, we seed an oracle (i.e., a conventional method for community detection) with the mean estimates of co-association of the current DCM to infer the communities in the new snapshot graph observed at time t; by contrast, in the exploration phase, the new communities are identified using the t-th graph only. In either phase, the community structure generated at time t is finally used to produce a superarm that will correspond to the dynamic consensus community structure up to t ($\mathcal{C}^*_{\leq t}$).

At each round t, in either of the phases, the algorithm invokes a community detection method $\mathcal{A}$. This is just required to deal with (*static*) *simple graphs*. While in the exploration phase it directly applies to the snapshot graph G_t (Line 4), to handle the exploitation phase, the method should also be able to deal with *weighted* graphs: in this case, $\mathcal{A}$ is executed on the graph $G_{\mathbf{M}}$ built from the current DCM matrix in such a way that the edge weights in $G_{\mathbf{M}}$ correspond to the entries of $\mathbf{M}$ (Line 6). Next, from the obtained partitioning $\mathcal{C}_{\mathbf{M}}$ of $G_{\mathbf{M}}$, the knowledge about the community memberships of entity nodes in $\mathcal{C}_{\mathbf{M}}$ is used to infer a community structure $\mathcal{C}^{(t)}$ on the snapshot graph G_t (Line 7). Each community in $\mathcal{C}^{(t)}$ will have node set corresponding to exactly one community in $\mathcal{C}_{\mathbf{M}}$, and edge set consistent with the topology of G_t. Any entity v that newly appears in G_t (i.e., $v \in \mathcal{V}_t \wedge v \notin \mathcal{V}_{t'}$, $\forall t' < t$) and is disconnected will form a community in its own.

The dynamic consensus community structure $\mathcal{C}^*_{\leq t}$, for each t, is generated in two steps. The first step (Line 9) corresponds to a simple projection of the community memberships from $\mathcal{C}^{(t)}$ onto $\mathcal{G}_{\leq t}$. The second step (Line 10) corresponds to *stochastic refinement* of the candidate $\mathcal{C}^*_{\leq t}$ obtained at the previous step. This refinement can be performed through local search optimization, which will relocate some nodes from their assigned community in $\mathcal{C}^*_{\leq t}$ to a neighboring one by acting greedily w.r.t. a quality criterion, such as *modularity*.

Finally, we devise the phase of DCM update (Lines 11) following a standard principle in reinforcement learning, whereby as the agent explores further, it is capable of updating its current estimate according to a general scheme of the form $newEstimate \leftarrow oldEstimate + \alpha(target - oldEstimate)$, which intuitively consists in moving the current estimate in the direction of a "target" value, with slope α. In our setting, we want to control the update of co-associations by subtracting a quantity α of resource from the co-associations of each node, at time t, and redistributing this quantity among the nodes in $c_i^{(t)}$, for each v_i. This redistribution corresponds to the *reward* of a single co-association, i.e., given v_i, the reward of assigning any v_j to the same community of v_i.

V. Conclusion

In this paper, we originally brought the CMAB paradigm into the context of community detection in temporal networks. We formulated the novel problem of dynamic consensus community detection, and proposed a general algorithmic scheme to solve it, which is versatile to the bandit strategy and to the static community detection method.

Our ongoing research concerns the development of a fully defined algorithm for the above problem, which leverages on (multilayer) modularity optimization in the stochastic refinement of the dynamic consensus solution. Besides the aforementioned features of versatility, the algorithm should be able to deal with temporal networks that can have different structure and evolution rate.

More details on this research are available at http://people.dimes.unical.it/andreatagarelli/cmab-dccd.

Acknowledgements

This research work has been partly supported by the Start-(H)Open POR Grant No. J28C17000380006 funded by the Calabria Region Administration.

References

[1] P. Agarwal, R. Verma, A. Agarwal, and T. Chakraborty. DyPerm: Maximizing Permanence for Dynamic Community Detection. In *Proc. PAKDD*, pages 437–449, 2018.

[2] P. Brodka, S. Saganowski, and P. Kazienko. GED: the method for group evolution discovery in social networks. *Soc. Netw. Anal. Min.*, 3(1):1–14, 2013.

[3] W. Chen, Y. Wang, and Y. Yuan. Combinatorial Multi-Armed Bandit: General Framework and Applications. In *Proc. ICML*, pages 151–159, 2013.

[4] J. Crawford and T. Milenkovic. ClueNet: Clustering a temporal network based on topological similarity rather than denseness. *PLOS ONE*, 13(5):1–25, 2018.

[5] N. Dakiche, F. B.-S. Tayeb, Y. Slimani, and K. Benatchba. Tracking community evolution in social networks: A survey. *Inf. Process. Manag.*, 56(3):1084–1102, 2019.

[6] T. La Fond, G. Sanders, C. Klymko, and H. Van Emden. An ensemble framework for detecting community changes in dynamic networks. In *Proc. IEEE HPEC*, pages 1–6, 2017.

[7] Y. Gai, B. Krishnamachari, and R. Jain. Combinatorial Network Optimization With Unknown Variables: Multi-Armed Bandits With Linear Rewards and Individual Observations. *IEEE/ACM Trans. Netw.*, 20(5):1466–1478, 2012.

[8] W. Gao, W. Luo, and C. Bu. Adapting the TopLeaders algorithm for dynamic social networks. *The Journal of Supercomputing*, 2017.

[9] M. Giatsoglou and A. Vakali. Capturing Social Data Evolution Using Graph Clustering. *IEEE Internet Computing*, 17(1):74–79, 2013.

[10] D. Greene, D. Doyle, and P. Cunningham. Tracking the evolution of communities in dynamic social networks. In *Proc. IEEE/ACM ASONAM*, pages 176–183, 2010.

[11] Y. Gur, A. J. Zeevi, and O. Besbes. Stochastic Multi-Armed-Bandit Problem with Non-stationary Rewards. In *Proc. NIPS*, pages 199–207, 2014.

[12] J. He and D. Chen. A fast algorithm for community detection in temporal network. *Physica A: Stat. Mech. Appl.*, 429:87–94, 2015.

[13] P. Jiao, W. Wang, and D. Jin. Constrained common cluster based model for community detection in temporal and multiplex networks. *Neurocomputing*, 275:768–780, 2018.

[14] M. N. Katehakis and A. F. Veinott Jr. Multi-armed bandit problem: Decomposition and computation. *Math. Oper. Res.*, 12:262–268, 1987.

[15] A. Lancichinetti and S. Fortunato. Consensus clustering in complex networks. *Sci. Rep.*, 2:336, 2012.

[16] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *Science*, 328(5980):876–878, 2010.

[17] G. Rossetti, L. Pappalardo, D. Pedreschi, and F. Giannotti. Tiles: an online algorithm for community discovery in dynamic social networks. *Machine Learning*, 106(8):1213–1241, 2017.

[18] R. S. Sutton and A. G. Barto. *Introduction to reinforcement learning*. Vol. 135. Cambridge: MIT press, 1998.

[19] A. Tagarelli, A. Amelio, and F. Gullo. Ensemble-based community detection in multilayer networks. *Data Min. Knowl. Discov.*, 31(5):1506–1543, Sep 2017.

[20] M. Takaffoli, R. Rabbany, and O. R. Zaïane. Community evolution prediction in dynamic social networks. In *Proc. IEEE/ACM ASONAM*, pages 9–16, 2014.

[21] Z. Wang, Z. Li, G. Yuan, Y. Sun, X. Rui, and X. Xiang. Tracking the evolution of overlapping communities in dynamic social networks. *Knowl. Based Syst.*, 157:81–97, 2018.

[22] K. Yang, Q. Guo, and J.-G. Liu. Community detection via measuring the strength between nodes for dynamic networks. *Physica A: Stat. Mech. Appl.*, 509:256–264, 2018.

[23] A. Zakrzewska and D. A. Bader. Tracking local communities in streaming graphs with a dynamic algorithm. *Soc. Netw. Anal. Min.*, 6(1):65:1–65:16, 2016.