Polarization identification on multiple timescale using representation learning on temporal graphs in Eulerian description

Anonymous Author(s)
Affiliation
Address
email

Abstract

Social media is often described as both reflecting and distorting real-life debates. Indeed, social division occurs not only offline but also online on various political topics or scientific controversies. Several studies propose tools to identify and quantify online controversies through stance detection or polarization measures. While polarization is typically studied as a “snapshot" in time of a social network, we consider it as a temporal process. Moreover, the recent evolution in temporal graph representation learning provides new tools to directly combine time, content and graph topology. Current techniques that characterize polarization are beginning to use these tools, typically using a Lagrangian description which focuses on user trajectories. In this article, we make a case for approaching these problems with a Eulerian description, the concurrent description in fluid mechanics. In this description, the temporal evolution of nodes embeddings is represented with a deformation of velocity vector fields. Finally, we validate our method on a retweet graph from the last French presidential election campaign.

1 Introduction

The recent COVID-19 crisis shed light on scientific controversies in real-time. Among them were the effectiveness of masks and drugs such as hydroxychloroquine, or the risk-benefit balance of vaccination. Unsurprisingly, these controversies quickly became part of public discourse, sometimes leading to public health issues. As a result, the general public re-discovered the fact that research and politics or media discourse are on different time scales. Political and media expectations have contributed to a politicisation of some scientific controversies for which scientific consensus was being built at the time.

The natural consequence of this politicisation is a polarisation of the subject along pre-existing political dividing lines. For example, in the United States, Republicans and Democrats appear to greatly differ in their approaches towards vaccination against COVID-19 (Fridman et al. [2021]). Moreover, these dividing lines appear online where their evolutions may be measured. There exists a direct link between opinion dynamics and the study of social dividing lines, as a shift of these lines indicates that users switch sides.

Polarization is usually evaluated on static networks once communities are already divided. However, this approach does not characterize the process under which this polarization arises. Instead, using temporal graphs, we treat polarization as an evolution of social community boundaries inside the network.

This paper develops an approach based on representation learning of temporal graphs. Like recent works, the goal is to embed graph evolution inside a latent space to study community evolution over

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

time (Li et al. [2022]). Our work is based on an implementation of Temporal Graph Networks (Rossi et al. [2020]) to embed retweet graphs. However, instead of using a Lagrangian description, i.e. following trajectories of users, social dynamics such as polarization are then seen through a Eulerian description from fluid mechanics and represented with "velocity" vector fields, focusing on specific locations in the latent space.

2 Related works

2.1 Measure polarization

Many measures of polarization have been proposed, most of them coming from models of opinion dynamics (Gurukar et al. [2020], Baumann et al. [2020]) but not always suited for graph topology. As graph topology plays a major role in information propagation, we describe the literature based on graph topology, either coming from stance detection or from polarization metrics.

A large literature studies stance detection, which is related to polarization. It can be based on content (tweets or hashtags), interactions (follows, retweets, etc) or users directly (name, location, bio, etc). For Twitter, interaction-based solutions generally provide better results to identify polarization (Magdy et al. [2016]) so we will focus on these. In addition, supervised approaches use labeled users and propagation to classify other users stances (Borge-Holthoefer et al. [2014], Kutlu et al. [2018]). More recent approaches are unsupervised using UMAP (Uniform Manifold Approximation and Projection) and force-directed graph placement (Darwish et al. [2019]). Lai et al. [2019] work on dynamic stance detection approaches on different kinds of interaction graphs (retweets, quotes, etc). Retweets appear to be a good feature in their settings. While they also present work on content, a limitation of their work is that they don't bring together content, graph topology and time.

Polarization can also be measured directly on the graph. A first approach is based on boundaries between communities (Guerra et al. [2013]). The solution has two steps: (1) defining communities and their boundaries, and (2) using these boundaries to compute the polarization metric. Another solution from Morales et al. [2015] defines a metric based on the relative sizes of detected communities and the distance between their centroids. The authors use it to compare day to day political oppositions in Venezuela. To use existing solutions outside of graph toolkit and relying on distances, new metrics such as Social Network Distance (Amelkin et al. [2017]) have been introduced. Graph embeddings and especially nodes embeddings can also be used. For instance, Garimella et al. [2015] compares several controversies measures and reports the good results of Random Walk Controversies and Embedding Controversies. Solutions based on embeddings are now vastly used (Rashed et al. [2020]) but they are mostly limited to static graphs. Finally, Raghavendra et al. [2022] models the evolution of signed networks with a dynamic Graph Neural Network (GNN), these networks being well suited for societal applications such as polarization measurement (Bonchi et al. [2019]).

So far, some solutions have been proposed which focus on polarization evolution (Kearney [2019]) (opposed here to polarization as a static situation), do not fully benefit from more recent development in representation learning on dynamic graphs. This paper aims to reconcile the study of polarization evolution with recent representation learning techniques by creating dynamic embeddings and computing their deformation.

2.2 Eulerian approach

Each new interaction transforms nodes embeddings. By gathering lots of these transformation, we can model embeddings deformations. Velocity vector fields are used to represent them. Instead of following users or communities along the time and trying to predict their trajectories, we focus on areas of the latent space and computing the direction of users passing through areas without following them. This is directly inspired from the dichotomy between Lagrangian and Eulerian descriptions in fluid mechanics.

Eulerian descriptions of complex systems have been used in the study of road traffic or crowd flow (van Wageningen-Kessels et al. [2016], Ma et al. [2016]). To the best of our knowledge, the most recent Eulerian approaches for social network are from Mirtabatabaei et al. [2014], Canuto et al. [2012]. This approach attempts to overcome Lagrangian limits of opinion dynamics models such as the Hegselmann-Kraus model. Due to the Eulerian description, they manage to prove properties of

the models, reusing ideas from fluid mechanics with operators such as divergence for instance. Our objective is to combine this description with social network embeddings.

2.3 Representation learning on dynamic graphs

As presented by Kazemi et al. [2020] in their survey on representation learning for dynamic graphs, it is possible to analyze existing models via an encoder/decoder scheme. The encoder associates nodes from the graph to their embeddings, from which the decoder performs a specific task, such as link prediction or node classification. In general, as stated in Rossi et al. [2020], deep learning models on dynamic graphs can be separated according to the representation, discrete or continuous, that they adopt for the temporal dimension.

The discrete approach, based on modelling the evolution of the graph in the form of successive snapshots, has been the most developed. Static models are used on each snapshot, their outputs being then combined. Chen et al. [2018] and Taheri et al. [2019] use static GNNs to encode the structural information for each snapshot, which are then passed into Recurrent Neural Networks-type modules to extract the temporal information. Pareja et al. [2020] resorts to a Graph Convolutional Network (GCN) whose parameters are updated by a Recurrent Neural Network (RNN) to model the dynamics of the network. Sankar et al. [2020] uses a joint temporal and structural attention mechanism to learn node representations based on historical snapshots. Mahdavi et al. [2018] and Sajjad et al. [2019] propose to use random walks to produce nodes embeddings, extending DeepWalk (Perozzi et al. [2014]) and Node2Vec (Grover and Leskovec [2016]) to discrete time dynamic graphs. The discrete time setting poses several difficulties: limited temporal granularity, lost order of appearance of edges, and impossible identification of links that frequently appear together (Skarding et al. [2021]).

Different approaches have thus been proposed to model dynamic graphs in a continuous setting. Nguyen et al. [2018] and Bastas et al. [2019] build network representation learning methods using temporal random walks to generate structure and continuous time-aware embeddings. ATiSE (Xu et al. [2019]) leverages time series method for temporal knowledge graph embedding model. Specific dynamic graph neural networks architectures have also been developed, such as Dyrep (Trivedi et al. [2019]), which produces dynamic low-dimensional node embeddings via a learned set of functions, Jodie (Kumar et al. [2019]), which uses coupled RNNs to learn node embedding trajectories, or Temporal Graph Attention (TGAT) (Xu et al. [2020]), which introduces a new temporal graph attention layer allowing it to efficiently aggregate temporal and topological features. These three architectures can be seen as special cases of general framework, Temporal Graph Networks (Rossi et al. [2020]), that adds a memory module to TGAT. TGN's framework being the most generic, we choose to use it as a reference model for our study.

3 Methods

3.1 Problem formalization

As explained in the introduction, social phenomena such as polarization can be approached from the point of view of the dynamics of opinions associated with them. It thus seems relevant to try to analyze them by studying the evolution of online political behaviors. As these behaviors can naturally be described in the form of a dynamic social graph, the following problem can be considered: how to represent the evolution of a dynamic graph in a way that synthesizes temporal, topological and content-related information. The objective is thus to build a tool allowing a synthetic visualization of online opinion dynamics, understood as proxies of underlying social phenomena.

3.2 Method Description

As detailed in 2.3, the tool we have selected to model the evolution of online behaviors is the TGN framework [1], the goal being to analyze the observable dynamics in its latent space.We therefore start by presenting the general workflow of the algorithm.

The approach proposed by TGN to record the history of a node, i.e. the set of events in which it took part (interactions, change of features, etc.), relies on the use of a memory module. The latter, based on

[1] `https://github.com/twitter-research/tgn`, licensed under Apache-2.0 license.

a RNN, allows to update a memory vector encoding the past behaviors of the nodes. It is coupled to an embedding module allowing to aggregate the neighborhood information and which can be realized by different functions: identity, a sum, an attention mechanism (Veličković et al. [2018]), etc. These two modules, memory and embedding, constitute an encoder associating to a dynamic graph a vector representation of its nodes. This representation can then be used by a decoder to perform a specific task such as link prediction or node classification. In this study, we will use a decoder associated to the first of these tasks, the training being then self-supervised. Indeed, the model learns, for each batch, to predict the probability of next interactions. Moreover, the choice of the decoder used for learning impacts the meaning that can be given to the representations provided by the encoder. This aspect of the problem is discussed in part 5.

3.2.1 Memory

In practice, the memory of a node is the hidden vector, initially zero, of a RNN, which is updated after each event involving this node. More precisely, when a node interacts, a message storing information about this interaction is computed and used as the input vector of the RNN, with the previous memory of the node serving as the input hidden vector. The output hidden vector then becomes the new memory of the node. In fact, as interaction data are processed in batches, it is necessary to define an aggregation rule for the messages produced within the same batch. Here, only the last computed message is retained.

More formally, for an interaction involving nodes i and j, the messages will be the following concatenations:

$$\begin{cases} m_i(t) = s_i(t^-)||s_j(t^-)||e_{ij}(t)||\phi(\Delta t_i) \\ m_j(t) = s_j(t^-)||s_i(t^-)||e_{ij}(t)||\phi(\Delta t_j) \end{cases}$$

where $s_k(t^-)$ refers to the memory of node $k \in \{i, j\}$ before its update and Δt_k the time difference between the last update time of node $k \in \{i, j\}$ and that of the interaction at the origin of the message.

3.2.2 Embedding module

The general form adopted for this module in Rossi et al. [2020] is:

$$z_i(t) = \sum_{j \in \mathcal{N}_i^l([0,t])} h(s_i(t), s_j(t), e_{ij}, v_i(t), v_j(t)),$$

where $\mathcal{N}_i^l([0, t])$ denotes the time neighborhood at rank l of node i on period $[0, t]$ and h a learnable function. In the TGN configuration adopted for this study, the representation of a node $i \in [\![1, n(t)]\!]$ is obtained by applying the temporal graph attention layer developed in Xu et al. [2020].

3.2.3 Final MLP and Loss Function

The decoder is composed of an MLP followed by a sigmoid function, allowing the prediction of link probability between nodes $i, j \in [\![1, n(t)]\!]^2$ at time $t \geq 0$ via the following formula:

$$\hat{\mathbb{P}}_{ij}(t) = \sigma(MLP(z_i(t)||z_j(t))).$$

In training, the estimated probabilities of the edges of the batch being processed and of those obtained by negative sampling (noted $(p_i^+)_{i \in [\![1,M]\!]}$ and $(p_i^-)_{i \in [\![1,M]\!]}$, where M is the batch size) are injected into a Binary Cross Entropy loss function:

$$\ell = -\frac{1}{M} \sum_{i=0}^{M-1} [\log(p_i^+) + \log(1 - p_i^-)].$$

3.3 Visualization of deformations

Lastly, by removing the final MLP, it can be used to create graph embeddings dynamically. Depending on the wanted time scale, an appropriate time step has to be defined. It can be a minute, an hour, a day or a week for example. Once the time step is defined, embeddings can then be compared between two instants to understand the geometric evolution of the landscape.

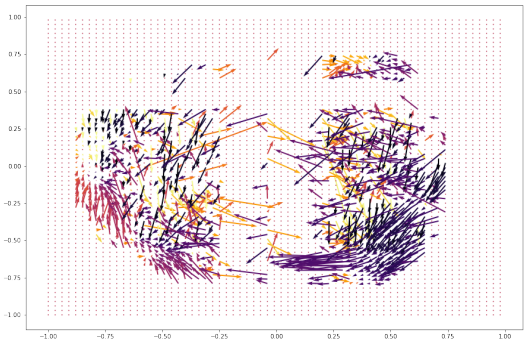

Figure 1: Velocity vector field obtained by comparing Figures 5 in Appendix. The axis are the same as the PCA used in Figure 2.

As stated before, a Lagrangian description with direct reading of trajectories is not the only way to describe polarization. What we want to do is to compute the evolution of nodes in areas with the best granularity. To do so, the solution adopted consists of defining a square mesh on the latent space [2]. For each square, the average move between two time steps is evaluated. To compute the average move of a square, the difference between positions of two time steps is computed and averaged over users located in the square at the beginning. This calculation is only performed for squares whose numbers of users exceed a certain threshold in order to ensure a certain significance. The move per square is represented with a vector leading to a vector field 1, where angles are associated with varying colors.

Two scales have to be chosen. The first one is spatial and depends on the volume of nodes. To get a good local representation, with a high resolution, large graphs are needed. The second one is the time scale. Depending on the addressed research question, the answer can differ. Working on a weekly level makes sense to consider long term polarisation for instance during a political election. Conversely, comparing embeddings between minutes can help to detect information campaigns such as astroturfing because they are going to change the structure locally, moving nodes in the same specific direction in the latent space. The question here is to measure a deformation of the latent space which could represent, for example, polarisation lines.

4 Experiments

4.1 Datasets

This dataset is collected through the Twitter API, by following 3000 personalities from the French political world. More precisely, all their immediate neighbors are considered, as well as the interactions that link them (*mentions*, *retweet*, *quote*, etc.). In addition, tweets and authors mentioning several *hashtags* related to the 2022 presidential elections are also collected. A more detailed description of this data collection is available as it has been used to work on the last two French presidential elections (Gaumont et al. [2018]). Any other graph with features could also be used.

From this dataset, a graph is built. It is obtained by selecting a period and by choosing the considered interaction (defining what an edge corresponds to). In our case, a dynamic graph is thus obtained, noted *Politics*, spanning 431999 seconds, including 30682 nodes and 1088815 interactions. Edges features are BERT embeddings of retweeted texts (dimension 300). Node features are zero vectors of dimension 172.

4.2 TGN settings and computing resources

The experiments presented in this section are performed with the TGN-attn configuration of Rossi et al. [2020], i.e. nodes memory is used, the memory updater is a GRU, the embedding module is an attention layer considering the 10 last neighbors, message aggregation is done by keeping only the most recent one and identity is used as message function. The hyperparameters are the same as those

[2] Instead of square, we should speak of a hypercube here. In practice, the solution developed is used after a dimension reduction to be visualized in two dimensions, hence the use of square.

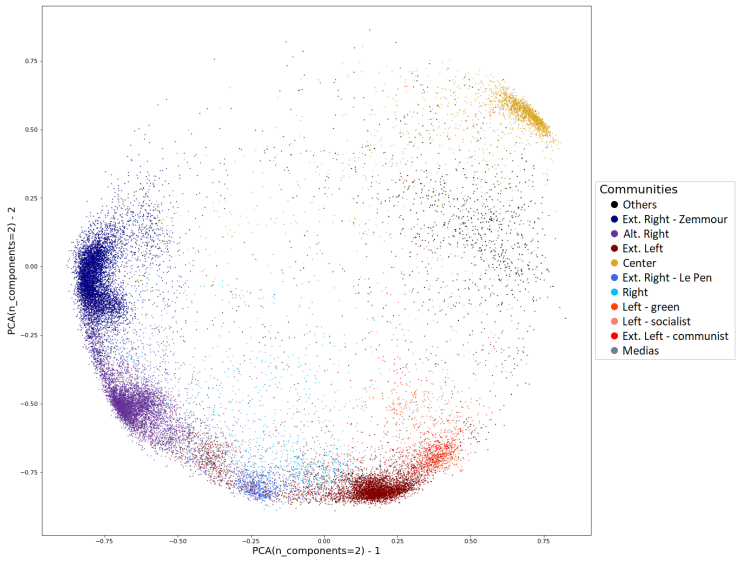

Figure 2: Embedding of the retweet graph with the French political communities (Left in red, Center in yellow and Right in blue). A PCA is used to reduce the dimensions.

used in the original paper. The code allowing the reproduction of these experiments[3] is available online [4].

Before considering embeddings deformation, one must ensure that these embeddings almost converged to their *true* position. As expected, during the first time steps, embeddings quickly evolve before stabilizing (or more exactly moving more slowly) and exhibiting interesting deformations.

4.3 Results

4.3.1 Preliminary analysis

A qualitative evaluation of our solution on a use case is proposed here. The first step is to provide meaningful interpretations for the embedding reduced dimensions.

Interestingly, the embeddings obtained provide similar results to other representations (Ramaciotti Morales et al. [2021], Chomel et al. [2022]) such as force directed algorithms leading to interpretable axis of the PCA. For instance, on Figure 2, the communities are computed using a Louvain algorithm on the static graph to exhibit the political colors of users. Extreme right parties appear on the left side (dark blue) and left parties appear in the bottom (red and pink). Lastly, the government party is isolated, on the top right (yellow). It seems that specific regions of the latent space can be linked to political communities and we will use this information to link deformations to political events.

Other information appear only in this representation such as the activity of each user. In a polar coordinate system, the angle is linked to political opinions, as we have seen. The radius is linked to activities: the higher the radius is, the higher the degree is. It is displayed in Figure 6 in Appendix.

Using this political affiliation with knowledge of impactful events that occurred during the last political election can help giving meaning to embedding deformations. For instance, both extreme parties criticized vaccination strategies leading to a polarization along the $y = x$ axis but bringing the two extreme parties together.

[3]Experiments were conducted on a laptop (Intel i5-8265U (1.60 GHz) CPU and 8 GB of RAM) and on a computer server (Nvidia Quadro P5000 GPU, 40*Xeon E5-2630 (2.20 GHz) CPUs and 128 GB of RAM).

[4]`https://github.com/TGL-sub/Polarization-identification-on-temporal-graphs-in-Eulerian-description`

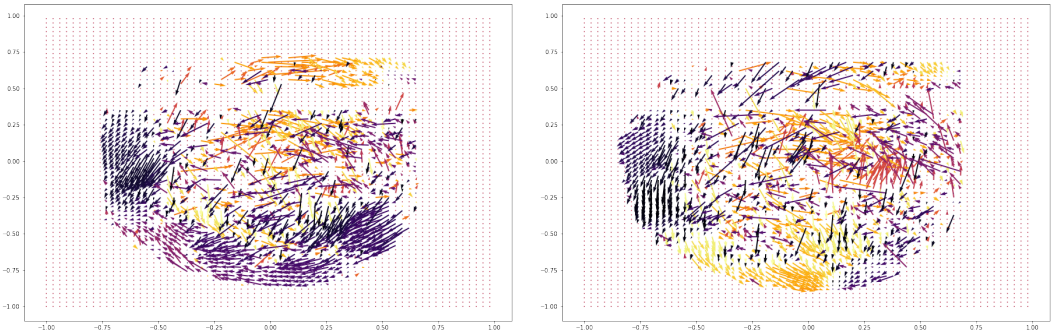

Figure 3: Example of two successive deformations, with a one hour time step, on April 5th 2022. Interesting patterns are in the lower left corner. The threshold is 10 nodes per tile. The axis are the same as the PCA used in Figure 2.

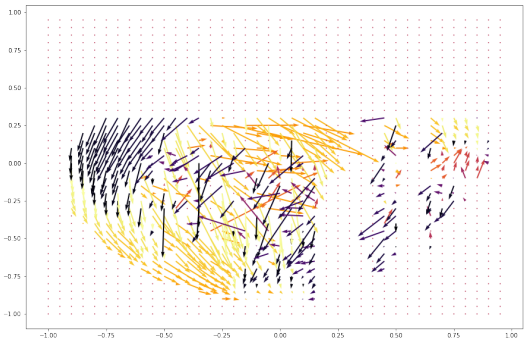

Figure 4: Embeddings deformation from April 7th to April 8th 2022. The threshold is 20 nodes per tile, explaining the large empty areas. The axis are the same as the PCA used in Figure 2.

4.3.2 Deformation examples

The first striking result is the apparent disorganization in regions with less nodes such as the center. In these regions, without strongly defined communities, there is no reason for close users to move in identical directions. We will focus the analysis on dense regions of the Figure 2 as they provide consistent and explainable evolutions.

A first example is studied with a time step of about an hour. It exhibits a pendulum swing that commonly occurs on the platform with an event bringing two groups closer before they take their distances again (Figure 3).

On the left panel, nodes at the bottom are going to the left. Politically speaking, it means that users from the extreme left are getting closer during this hour to alt right users. This matches perfectly the network activity as several major alt right accounts including the head of the movement Florian Philippot has been largely retweeted. In the meantime, extreme left figures have been less active during this time step, leading to the large departure from this area. During the same time interval, the extreme right around Zemmour promoted his role in the help of a victim of aggression, gathering users of the community.

During the next time step, leaders of the extreme left, such as Jean-Luc Mélenchon, were massively retweeted when they criticized the government. At the same time, the alt right was less active, pushing users from the bottom to the right of the plot.

A second example has been run with a one day time step. Instead of observing the impact on the network of a few tweets, this larger time step allows us to identify larger political movements. A consistent result across the last week before the first round is the exodus from one extreme right community (Zemmour) to the other one (Le Pen). It can be identified with the flow from the left

to the bottom of the space. This result is consistent with political analysis from media, polls and eventually outcomes from the French presidential election of 2022.

5 Discussion

It has been shown that the method proposed in this study, based on the visual analysis of the deformations of the latent space, can be used to identify social phenomena such as polarization[5]. Nevertheless, two questions appear naturally from the identification of such phenomena.

Actors/Users Which actors/users are involved and can explain these evolutions? An in-depth analysis of the role of leaders has been provided for the use case, but can it be automated?
The aforementioned deformations correspond to a synthetic representation of the dynamics of nodes embeddings, allowing an efficient visualisation and detection of social phenomena. Once detected, it would thus be interesting to perform a dynamic clustering on nodes embeddings instead of deformations, focused on the identified period and location. It should be noted that such a clustering would not be directly feasible, without a prior identification of the phenomenon through deformation analysis. Indeed, social phenomena are temporally and spatially localized. Consequently, looking at dynamic clusters of actors over the whole dataset will lead to hardly interpretable results.
One may compare this solution to the dynamic node classification configuration of TGN also introduced by Rossi et al. [2020] but not described here. However, such a dynamic clustering would not be redundant with this configuration of TGN as its object is different. Indeed, the classifier tries to sort the nodes based on labeled examples which do not exist in this case. On the contrary, the goal here is to identify nodes involved in the detected social phenomena, *a priori* independently of their features. Nevertheless, once identified, these nodes could effectively be used to train a classifier to detect users likely to be involved in such phenomena.

Content/Topics Is it possible to find which topics are covered by these deformation?
The focus is now on the semantic aspect of opinion dynamics, requiring to interpret the space of nodes representations in a meaningful way. Analogously to the opinion space used in Mirtabatabaei et al. [2014], the idea is thus to give an intrinsic meaning and not simply relative to the positions of nodes in this space. An approach would be to decode nodes embeddings via the decoder associated with the text encoder used to generate the node and edge features, with the hope that the information contained in these text embeddings has been preserved by TGN.

Choice of the decoder As previously mentioned, the interpretation of the nodes embeddings will depend on the choice of the decoder used for training as it impacts the latent space. In the case of link prediction, as used in this paper, the position in the latent space corresponds to a probability of interaction with the other points of this space. In particular, the use of a MLP as a decoder forbids to interpret the distance between two nodes as a proxy of their interaction probability. Indeed, the proximity of the embeddings reflects equivalent link probabilities with respect to the rest of the network, not a high chance of interacting with each other. An example would be users of low degree retweeting the same public personality. Their behavior with respect to probabilities of links to other nodes is similar, which places them next to each other. However, their mutual interaction probabilities are low.

Thus, the clusters observed in Figures 2 and 5 group types of users, corresponding to the different retweeting behaviors existing in the dataset. Observing their deformations with an Eulerian description seems therefore relevant, as positions in the opinion space appear to be key drivers of the network dynamics. Indeed, two close users are supposed to evolve in short term in similar ways, since their link probabilities to the rest of the network are analogous.

As stated before, instead of predicting links, a decoder could be associated with a node classification task during training. In this case, the interpretation of proximity of embeddings would be based on the similarity of their past according to features learned during the training to dissociate the nodes into different classes.

[5]These tools are meant to be used to measure and identify it in short term evolutions. Because of the complexity of social networks and their unpredictability on the long run, it cannot really be used to predict the impact of a few tweets directly, making it of no use to manipulate opinion.

Appendix

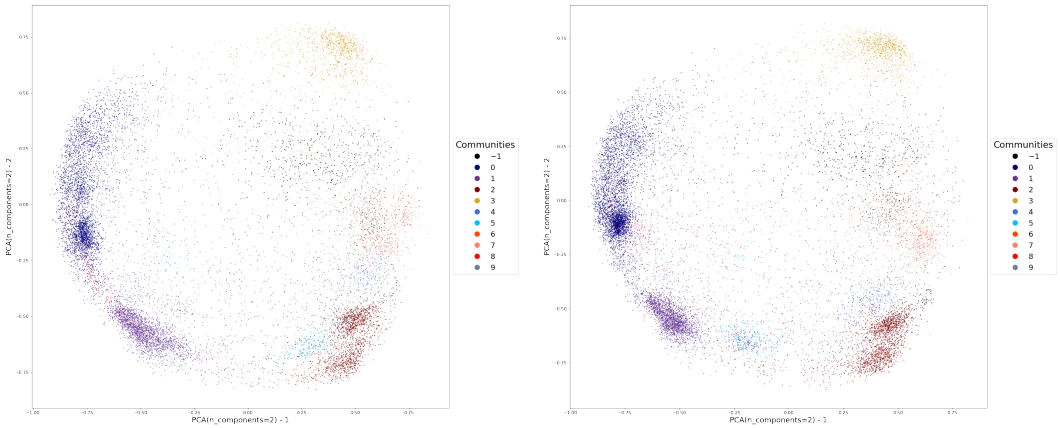

Figure 5: Node embeddings of a retweet graph from two successive time steps. Colors represent communities obtained using Louvain algorithm. A PCA is used to reduce the dimensions.

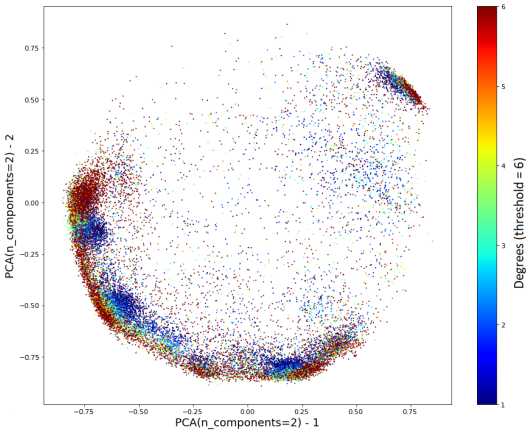

Figure 6: Embedding of the French political retweet graph. Nodes with high activity are in brown, nodes with only one interaction (low activity) are in blue. A PCA is used to reduce the dimensions.

References

Victor Amelkin, Petko Bogdanov, and Ambuj K. Singh. A distance measure for the analysis of polar opinion dynamics in social networks. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*. IEEE, apr 2017. doi: 10.1109/icde.2017.64. URL `https://doi.org/10.1109%2Ficde.2017.64`.

Nikolaos Bastas, Theodoros Semertzidis, Apostolos Axenopoulos, and Petros Daras. evolve2vec: Learning network representations using temporal unfolding. In *MMM*, 2019.

Fabian Baumann, Philipp Lorenz-Spreen, Igor M. Sokolov, and Michele Starnini. Modeling echo chambers and polarization dynamics in social networks. *Phys. Rev. Lett.*, 124(4):048301, January 2020. doi: 10.1103/PhysRevLett.124.048301. URL `https://link.aps.org/doi/10.1103/PhysRevLett.124.048301`.

Francesco Bonchi, Edoardo Galimberti, A. Gionis, Bruno Ordozgoiti, and Giancarlo Ruffo. Discovering polarized communities in signed networks. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019.

Javier Borge-Holthoefer, Walid Magdy, Kareem Darwish, and Ingmar Weber. Content and network dynamics behind egyptian political polarization on twitter, 2014. URL https://arxiv.org/abs/1410.3097.

C. Canuto, F. Fagnani, and P. Tilli. An eulerian approach to the analysis of krause's consensus models. *SIAM Journal on Control and Optimization*, 50(1):243–265, 2012. doi: 10.1137/100793177. URL https://doi.org/10.1137/100793177.

Jinyin Chen, Xueke Wang, and Xuanheng Xu. Gc-lstm: Graph convolution embedded lstm for dynamic link prediction, 2018. URL https://arxiv.org/abs/1812.04206.

Victor Chomel, Maziyar Panahi, and David Chavalarias. Manipulation during the french presidential campaign: coordinated inauthentic behaviors and astroturfing analysis on text and images. Proceedings of Complex Networks and their Applications, 2022.

Kareem Darwish, Peter Stefanov, Michaël Aupetit, and Preslav Nakov. Unsupervised user stance detection on twitter, 2019. URL https://arxiv.org/abs/1904.02000.

Ariel Fridman, Rachel Gershon, and Ayelet Gneezy. Covid-19 and vaccine hesitancy: A longitudinal study. *PloS one*, 16(4):e0250123, 2021.

Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. Quantifying controversy in social media, 2015. URL https://arxiv.org/abs/1507.05224.

Noe Gaumont, Maziyar Panahi, and David Chavalarias. Methods for the reconstruction of the socio-semantic dynamics of political activist Twitter networks. *PLoS ONE*, 13(9), September 2018. doi: 10.1371/journal.pone.0201879. URL https://hal.archives-ouvertes.fr/hal-01575456. La version française est disponible en V1 de cette archive.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

Pedro Henrique Calais Guerra, Wagner Meira Jr, Claire Cardie, and Robert D. Kleinberg. A measure of polarization on social media networks based on community boundaries. In *ICWSM*, 2013.

Saket Gurukar, Deepak Ajwani, Sourav Dutta, Juho Lauri, Srinivasan Parthasarathy, and Alessandra Sala. Towards quantifying the distance between opinions, 2020. URL https://arxiv.org/abs/2001.09879.

Seyed Mehran Kazemi, Rishab Goel, Kshitij Jain, Ivan Kobyzev, Akshay Sethi, Peter Forsyth, Pascal Poupart, and Karsten M. Borgwardt. Representation learning for dynamic graphs: A survey. *J. Mach. Learn. Res.*, 21:70:1–70:73, 2020.

Michael Wayne Kearney. Analyzing change in network polarization. *New Media & Society*, 21(6): 1380–1402, 2019. doi: 10.1177/1461444818822813. URL https://doi.org/10.1177/1461444818822813.

Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2019.

Mucahid Kutlu, Kareem Darwish, and Tamer Elsayed. Devam vs. tamam: 2018 turkish elections, 2018. URL https://arxiv.org/abs/1807.06655.

Mirko Lai, Marcella Tambuscio, Viviana Patti, Giancarlo Ruffo, and Paolo Rosso. Stance polarity in political debates: A diachronic perspective of network homophily and conversations on twitter. *Data & Knowledge Engineering*, 124:101738, 2019. ISSN 0169-023X. doi: https://doi.org/10.1016/j.datak.2019.101738. URL https://www.sciencedirect.com/science/article/pii/S0169023X19300187.

Jinning Li, Huajie Shao, Dachun Sun, Ruijie Wang, Yuchen Yan, Jinyang Li, Shengzhong Liu, Hanghang Tong, and Tarek Abdelzaher. Unsupervised belief representation learning with information-theoretic variational graph auto-encoders. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, jul 2022. doi: 10.1145/3477495.3532072. URL https://doi.org/10.1145%2F3477495.3532072.

Yuxin Ma, Tao Lin, Zhendong Cao, Chen Li, Fei Wang, and Wei Chen. Mobility viewer: An eulerian approach for studying urban crowd flow. *IEEE Transactions on Intelligent Transportation Systems*, 17(9):2627–2636, 2016. doi: 10.1109/TITS.2015.2498187.

Walid Magdy, Kareem Darwish, Norah Abokhodair, Afshin Rahimi, and Timothy Baldwin. Isisisnotislam or deportallmuslims? predicting unspoken views. In *Proceedings of the 8th ACM Conference on Web Science*, WebSci '16, page 95–106, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342087. doi: 10.1145/2908131.2908150. URL https://doi.org/10.1145/2908131.2908150.

Sedigheh Mahdavi, Shima Khoshraftar, and Aijun An. dynnode2vec: Scalable dynamic network embedding. *2018 IEEE International Conference on Big Data (Big Data)*, pages 3762–3765, 2018.

Anahita Mirtabatabaei, Peng Jia, and Francesco Bullo. Eulerian opinion dynamics with bounded confidence and exogenous inputs. *SIAM Journal on Applied Dynamical Systems*, 13(1):425–446, 2014. doi: 10.1137/130934040. URL https://doi.org/10.1137/130934040.

A. J. Morales, J. Borondo, J. C. Losada, and R. M. Benito. Measuring political polarization: Twitter shows the two sides of venezuela. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(3): 033114, March 2015. doi: 10.1063/1.4913758. URL https://doi.org/10.1063/1.4913758.

Giang Hoang Nguyen, John Boaz Lee, Ryan A. Rossi, Nesreen Ahmed, Eunyee Koh, and Sungchul Kim. Dynamic network embeddings: From random walks to temporal random walks. *2018 IEEE International Conference on Big Data (Big Data)*, pages 1085–1092, 2018.

Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao B. Schardl, and Charles E. Leiserson. EvolveGCN: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: online learning of social representations. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014.

Mr. Raghavendra, Kartik Sharma, Mariappan Anandkumar, and Srijan Kumar. Signed link representation in continuous-time dynamic signed networks. *ArXiv*, abs/2207.03408, 2022.

Pedro Ramaciotti Morales, Jean-Philippe Cointet, and Gabriel Muñoz Zolotoochin. Unfolding the dimensionality structure of social networks in ideological embeddings. In *2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, Amsterdam, Netherlands, November 2021. URL https://hal.archives-ouvertes.fr/hal-03315759.

Ammar Rashed, Mucahid Kutlu, Kareem Darwish, Tamer Elsayed, and Cansın Bayrak. Embeddings-based clustering for target specific stances: The case of a polarized turkey. 2020. doi: 10.48550/ARXIV.2005.09649. URL https://arxiv.org/abs/2005.09649.

Emanuele Rossi, Benjamin Paul Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael M. Bronstein. Temporal graph networks for deep learning on dynamic graphs. *ArXiv*, abs/2006.10637, 2020.

Hooman Peiro Sajjad, Andrew Docherty, and Yu. O. Tyshetskiy. Efficient representation learning using random walks for dynamic graphs. *ArXiv*, abs/1901.01346, 2019.

Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2020.

Joakim Skarding, Bogdan Gabrys, and Katarzyna Musial. Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey. *IEEE Access*, 9:79143–79168, 2021.

Aynaz Taheri, Kevin Gimpel, and T. Berger-Wolf. Learning to represent the evolution of dynamic graphs with recurrent models. *Companion Proceedings of The 2019 World Wide Web Conference*, 2019.

Rakshit S. Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. Dyrep: Learning representations over dynamic graphs. In *ICLR*, 2019.

Femke van Wageningen-Kessels, Ludovic Leclercq, Winnie Daamen, and Serge P. Hoogendoorn. The Lagrangian coordinate system and what it means for two-dimensional crowd flow models. *Physica A: Statistical Mechanics and its Applications*, 443:272–285, 2016. doi: 10.1016/j.physa.2015.09.048. URL `https://hal.archives-ouvertes.fr/hal-01215762`.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=rJXMpikCZ`.

Chengjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Hamed Shariat Yazdi, and Jens Lehmann. Temporal knowledge graph embedding model based on additive time series decomposition, 2019. URL `https://arxiv.org/abs/1911.07893`.

Da Xu, Chuanwei Ruan, Evren Körpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs. *ArXiv*, abs/2002.07962, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Section 4.3 for a use case based on the announced approach.
   (b) Did you describe the limitations of your work? [Yes] See Section 5 for a discussion of improvements that could be made.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See footnote 5 in Section 5.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See footnote 4 in Section 4.2.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Sections 4.1 for datasets description and 4.2 for TGN settings and hyperparameters.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See footnote 3.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] For TGN, see the end of Section 2.3.
   (b) Did you mention the license of the assets? [Yes] See footnote 1 in Section 3.2.

(c) Did you include any new assets either in the supplemental material or as a URL? [No]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] The data collection was described in a reference and collected using the Twitter API.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] The collected data are tweets and retweets from French politics. User ids are kept separately from users activities. The user decoding is only used on nodes with highest degrees as they are public personalities. It is used to make the political affiliation of communities.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]