Centrality informed embedding of networks for temporal feature extraction

Oggier, Frédérique; Datta, Anwitaman

2021

Oggier, F. & Datta, A. (2021). Centrality informed embedding of networks for temporal feature extraction. *Social Network Analysis and Mining*, 11, 12-.
<https://dx.doi.org/10.1007/s13278-021-00720-8>

<https://hdl.handle.net/10356/154815>

<https://doi.org/10.1007/s13278-021-00720-8>

© 2021 Springer. This is a post-peer-review, pre-copyedit version of an article published in *Social Network Analysis and Mining*. The final authenticated version is available online at:
<http://dx.doi.org/10.1007/s13278-021-00720-8>.

Downloaded on 26 Apr 2023 16:19:03 SGT

Centrality Informed Embedding of Networks for Temporal Feature Extraction

Frédérique Oggier¹ and Anwitaman Datta²

¹ Division of Mathematical Sciences, Nanyang Technological University, Singapore

² School of Computer Science and Engineering, Nanyang Technological University,
Singapore

Abstract. We propose a two-step methodology for exploring the temporal characteristics of a network. First, we construct a graph time series, where each snapshot is the result of a temporal whole-graph embedding. The embedding is carried out using the degree, Katz and betweenness centralities to characterize first and higher order proximities among vertices. Then a principal component analysis is performed over the collected temporal graph samples, which exhibits eigengraphs, graphs whose temporal weight variations model the sampled graph series. Analysis of the temporal timeline of each of the main eigengraphs reveals moments of importance in terms of structural graph changes. Parameters such as the dimension of the embeddings and the number of temporal samples are explored. Two case studies are presented: a Bitcoin subgraph, where findings are cross-checked by looking at the subgraph behavior itself, and the Enron email network, which allows us to compare our findings with prior studies. In both cases, the proposed methodology successfully identified temporal structural changes in the graph evolution.

Keywords: Social network dynamics · Time series of graphs · Dynamic network · Centrality · Graph embedding

1 Introduction

Given a (multi-)graph with time stamps on its edges, we study the problem of identifying timepoints where the graph undergoes significant structural changes, which we infer as signals indicating interesting or anomalous events for whichever real world activities induce the graph being studied.

A typical approach to temporal graph analysis consists of modeling the graph either as a contact sequence graph (when the interaction durations are short), or as an interval graph (when edges are active over intervals). See e.g. [10] for a survey of techniques used in this setting. The time stamps themselves may either be given, or aggregated from the available data. Different works that explore ways to represent the temporal information include, e.g. [14], which proposes a method to combine graph invariants over time, by assuming an underlying probability distribution on the edges over time; [15], which relies on Bayesian

hypothesis testing to quantitatively determine when a change occurred in the network timeline; [12], which uses principal component analysis (PCA) on the Blockchain network during 2012-2013 to estimate the price of the cryptocurrency Bitcoin. Recent works include an algebraic characterization of temporal networks using semirings [2] and an analysis of network motifs to study diffusion and inhibition [16].

In this work, we are interested in isolating relevant nodes, edges and time stamps. This is done by discarding information that is not relevant enough to our purpose. We realize this by combining the notion of graph embedding with centrality. The main reasons to wish to ignore some information are that the graph considered and all its user interactions could just be too large to handle, or that some interactions are just not relevant, e.g. in a social network, some users may have been active just once for a few minutes over several months or years of data, or because a subgraph is considered, and some boundary effects should be neglected. Centralities form a class of versatile measures that encompass diverse ways to capture node importance. Hence, they are naturally suited for determining which nodes and their edges are critical.

The proposed approach comprises two steps. (1) The first step includes a graph embedding (see Section 3), which is applied on temporal graph samples, that generates a graph time series $G(t) = (V, E(t)), t = 1, 2, \dots, T$, where the vertex set V is fixed but the edge set $E(t)$ varies over time. The proximity metric used is based on node centralities: we experiment with three types of centrality, the degree centrality that accounts for the local influence of a node, Katz centrality that incorporates the global influence of a node, and the betweenness centrality, which highlights “nodes in the middle”. (2) The second step is a PCA on the graph time series $G(t) = (V, E(t)), t = 1, 2, \dots, T$ (see Section 4). Relying on PCA analysis to sample the underlying edge distribution (we do not assume any knowledge on the actual distribution, unlike e.g. [14]) is a well-known technique, which has been popular in the context of eigenfaces (e.g. [17]) and was also used in [12] in the context of graph analysis. PCA enables both a temporal analysis, and as usual, provides a technique for dimensionality reduction. The result of the PCA analysis is a series of temporal eigengraphs, ranked by the importance of their corresponding eigenvalue. The dimensional reduction is obtained by ignoring small eigenvalues and focusing on the largest ones. The evolution of each eigengraph over time serves as a basis to identify anomalies and more generally time frames of interest in temporal networks where significant changes happen. Our approach is agnostic of the semantics of relationship, if any, among nodes. Changes in the timeline of an eigengraph, characterized by gradient changes (typically local maxima), are flagged as signals that something interesting, atypical or anomalous is happening in the real world system which induced the graph. However they do not in themselves identify the specific nature of such events, and are devoid of any semantics.

Graph embeddings (Step 1) and PCA (Step 2) achieve dimensionality reduction in a complementary manner - the former in the graph size dimension, by selecting the most relevant nodes over time, and the latter along the temporal

dimension, by projecting the graph samples over its main eigengraphs. While the principal component analysis over snapshots of the graphs is the key to determine the temporal changes, the role of the graph embedding is to filter out a small subset of the graph snapshots to reduce the computational costs, while ensuring that the embedded graph nevertheless carry most of the (or adequate) information about the full graph, as needed to infer important temporal changes in the full graph. Choosing different centralities as proximity metrics for embedding allows our approach the flexibility to differentiate the type of nodes as per their extent of their influence being localized, global or “in the middle”, which has the potential to capture events in the graph at different granularities. The method would be considered effective if it pinpoints relevant time frames, despite the effective data compression resulting from the graph embeddings. We demonstrate the results obtained on two data sets in Section 5, a subgraph of the Bitcoin network and the Enron email network (as used in respectively [12] and [14, 15, 8]). For the Bitcoin subgraph, for which no ground truth is available, we cross-check the findings by looking at the graph samples themselves.

2 Related works

One step of the proposed method is to perform a PCA on graph time series. PCA on vectorized entropy histograms, that is histograms created from node degree quantiles, was proposed in [18] to characterize node degree configurations and their distribution across edges. A notion of principal component for graphs was proposed in [6], where Fraiman et al. consider both labeled graphs and families $\{G_t\}_{t \geq 1}$ of labeled graphs evolving over time. They work under the assumptions that $\{G_t\}_{t \geq 1}$ forms a stationary and ergodic sequence of random graphs with a given law. They introduce a notion of principal component for graphs, which mimics over a graph space, the usual notion of principal component, defined as follows. Let $\mathcal{G}$ be a space of graphs with n nodes endowed with the metric $d(G, H) = \sum_{i < j} |A_{ij} - B_{ij}| c_{ij}$, where A, B are the adjacency matrices of G, H respectively, and $c_{ij} > 0$ is the cost of adding or deleting the link (i, j) , understood as an edit operation that takes G into H . Given G, H two graphs in $\mathcal{G}$, $G \wedge H$ is the graph with only the common links to both, and $|\mathcal{G}, G \wedge H|$ is the weighted inner product, given by the costs between A and B . Given a random network G , the first principal component is defined to be the set of networks that maximize the variance of the projection $|G \wedge H|/|H|$.

The graph times series that we consider is built using a centrality informed graph embedding. The term *graph embedding* is broadly used in two contexts: (1) to define a map from a graph G endowed with some distance to a metric space (possibly of higher dimension) with another distance, so that the distances are preserved in some sense (this is a particular case of *metric embeddings*, e.g [3]), alternatively, (2) “to represent a graph as low dimensional vectors while the graph structures are preserved” (quoted from [4]). We are interested in (2). In this context, many different notions of graph embeddings have been studied with different applications in mind. In [19], a dynamic graph $G = (G_1, \dots, G_t)$ is con-

sidered, whose graph embeddings $(Z_1, \dots, Z_t)$ is computed so as to minimize a loss function $\sum_{\tau=1}^n \|G_\tau - Z_\tau Z_\tau^T\|_F^2$ with temporal regularization. The purpose is link prediction, more precisely, $Z_1, \dots, Z_t$ is used to compute Z_{t+1} which in turn is used to predict G_{t+1} . In [13], graph embeddings are computed using learning methods. Mimicking document embedding models, graph structures are learned by considering rooted subgraphs of a fixed degree (given a node u , its rooted subgraph encompasses all the nodes and corresponding edges that are reachable in d hops from u). Applications considered are graph classification and clustering. For more kernel methods and neural networks based graph embeddings, we refer to the survey [4].

In [8], Goyal et al. look at embedding methods for dynamic graphs. A motivation for their work is that given a sequence $(G_1, \dots, G_T)$ of graphs, embedding each graph individually may not be the best approach, instead one may want a stable dynamic embedding, in the sense that two consecutive embeddings should differ by only a small amount (assuming the underlying graphs are themselves changing by a small amount). In order to do so, the algorithm proposed learns the embedding of G_t starting from that of G_{t-1} . An application is link prediction, but more relevant to the current work is its application to anomaly detection, since the Enron data set is also considered. A comparison between our findings and those of [8] in the context of the study of the Enron dataset is provided in Subsection 5.2.

3 Graph Embeddings

By graph embedding, we mean a map that sends a graph of interest to one low dimensional vector while preserving the dominant graph structures in terms of connectivity. The most common way to define a graph embedding [4] is to embed every node of the graph as a vector in a low dimensional space, but we are interested in a *whole-graph embedding*, where the whole graph G is sent to a vector in $\mathbb{R}^N$ where $N \ll |V|^2$, for V the set of vertices. The dimension $|V|^2$ is the dimension needed if the graph is general, but if it contains no self-loop, it reduces to $N \ll |V|^2 - |V|$, this quantity is divided by two if the graph is symmetric, namely $N \ll \frac{|V|^2 - |V|}{2}$.

Proximity Metrics. To choose the embedding, one needs to decide which graph structural properties to preserve. Typically, relations among neighbor nodes should be maintained. This is formally defined by the notion of *kth order proximity*: when $k = 1$, this defines the proximity between nodes u and v directly, when $k = 2$, the proximity takes into account the direct neighbors of u and v . Examples of metrics considered include cosine similarities, Rooted Page Rank to name a few (see [7] for more details). The choice of the metric depends on the nature of the graph. We will consider graphs modeling interactions among users, and choose to study centrality measures as *kth order proximities* (see e.g. [5] for a tutorial and survey about centrality measures):

1. *degree centrality* naturally takes into account nodes that are highly connected (it is defined as the node degree divided by $|V| - 1$),

2. *Katz centrality* singles out influential nodes, by measuring how nodes interact with other nodes (if A is the adjacency matrix of the graph, it is computed using $\sum_{i \geq 1} \alpha^i A^i$ where α is a weight that gives more importance to the close nodes and less to those far away),
3. *betweenness centrality* adopts a different view point and emphasizes nodes that are “in the middle of things” (the betweenness of u is the sum of the shortest paths between every pair of nodes that go through u divided by the total number of shortest paths between every pair).

There are many other centrality measures. The degree centrality is chosen because it is the most classical one, and nodes of high degree are natural candidates to incorporate in the graph embedding. Then Katz centrality is a representative of eigenvalue based centralities, and it includes nodes whose importance is seen by looking at a large horizon. Finally betweenness centrality is also a classical measure, with a different insight. Because of the size of graph considered, all centralities need to be easily computed or approximated.

Graph Embedding for Temporal Analysis. Consider a graph which evolves over time. The graph data is extracted for a time range $[1, T]$, and a graph G is formed which contains nodes that were active at least once during the time range, and edges with time stamps. The graph G will be sampled over time, using the edge time stamps, to create a discrete time series $G(t) = (V(t), E(t))$ for $t = 1, \dots, T$, where $V(t)$ is the set of vertices active at time t . If G is small enough, one could just obtain a time series by considering the whole set of vertices at all time, and look at the evolution of $E(t)$ over time. However, when the size of the graph grows, this may not be tractable. Considering all the vertices may also simply be not that meaningful, since there are many graph instances where most of the graph evolution is driven by the most influential nodes, a notion well captured by centrality measures. We use a graph embedding ϕ with degree/Katz/betweenness centrality as a proximity metric to map the graph time series $G(t) = (V(t), E(t))$ to $\phi(G(t)) = (\phi(V(t)), \phi(E(t)))$ where $\phi(V(t))$ becomes a constant (much smaller than the original vertex size) and does not depend on time anymore. This is a design choice, that enables the use of PCA: every graph embedding has the same size, so we can build the matrix on which the PCA will be performed. Each of the centralities characterizes which nodes should be included over time: those with high centralities in G are highly influential over the whole graph and therefore are kept as vertices of $\phi(G(t))$, then $\phi(V(t)) = V$ remains constant, and the temporal variation appears in the edge set $\phi(E(t))$.

4 Principal Component Analysis and Graph Time Series

Consider the embedded graph time series $\phi(G(t)) = (\phi(V(t)), \phi(E(t)))$, $t = 1, 2, \dots, T$ where $|\phi(V(t))| = n$. Let $A(t)$ be the corresponding adjacency matrix, whose (u, v) -entry takes values 1 if there is an edge between u and v , and 0 otherwise. If there are several edges between u and v , the entry is either 1 if the graph is unweighted, or it contains the number of edges if the graph is

weighted. We ignore self-loops, that is edges of the form (u, v) with $u = v$. Each entry of $A(t)$ may be interpreted as a random variable taking values 0 or 1 for an unweighted graph (or 0 to the maximum weight for a weighted graph). We need $N = n^2 - n$ entries to characterize a directed graph $A(t)$, and $N = (n^2 - n)/2$ for an undirected graph. The probability distribution we would be looking at is the edge distribution over the time period $[1, T]$. Since every adjacency matrix can be vectorized to form an $N = n^2 - n$ (respectively $N = (n^2 - 2)/2$) vector, we have a collection of T N -dimensional vectors, that can be put into a sample matrix $\tilde{S}$ of size $T \times N$. We next perform a principal component analysis (PCA) of this data.

Mean centering. To start with, we will assume that the PCA will be performed on a dataset which is centered, meaning that the data mean is 0. This removes bias that we do not wish to model. If the dataset is not centered, we proceed as follows. To obtain an average graph, we just sum the rows of $\tilde{S}$ and divide the obtained vector by T . We then subtract this average graph from every graph (every row $\tilde{S}_i$) of $\tilde{S}$, this gives a mean-centered matrix S . We notice that now every graph S_i becomes weighted in the sense that coefficients are not 0 and 1 anymore, and some even have negative weights. The mean-centering process is illustrated in Figure 1. For example, $A(1)$ depicts the adjacency matrix of $G(1)$. Since $G(1)$ is a star graph, its adjacency matrix contains one row of ones (the first row here, corresponding to node 1 to be the center of the graph), so there is an edge between node 1 and every other node, and since the graph is undirected, the matrix is symmetric, and there are ones in the first column, representing an edge between every node and node 1. The matrices $A(1), A(2), A(3)$ contain two colors, a dark color for the zeroes, and a light color for the ones. The color scheme at the bottom right represents the different values appearing in the matrices. While this process is reversible, we observe that the centered graph data cannot be readily interpreted anymore, e.g. in terms of degree distribution.

Fig. 1. $A(1), A(2), A(3)$ are the adjacency matrices for the graphs $G(1), G(2), G(3)$. Only $G(1)$ is shown. Then $\bar{A} = \frac{A(1)+A(2)+A(3)}{3}$ is the average adjacency matrix, and $A(i) - \bar{A}$, $i = 1, 2, 3$ are the corresponding mean-centered adjacency matrices.

Minimizing error during projection. If we wanted to project S over a line (a one-dimensional space), given by the unit vector $\mathbf{w}$, we could just compute the inner product $\langle S_i, \mathbf{w} \rangle$ for every row S_i , which is the distance of the projection from the origin, and S_i is projected onto $\langle S_i, \mathbf{w} \rangle \mathbf{w}$. This typically introduces an average error

$$\begin{aligned}
& \frac{1}{T} \sum_{i=1}^T \|S_i - \langle S_i, \mathbf{w} \rangle \mathbf{w}\|^2 \\
&= \frac{1}{T} \sum_{i=1}^T [\langle S_i, S_i \rangle - 2\langle S_i, \langle S_i, \mathbf{w} \rangle \mathbf{w} \rangle + \langle \langle S_i, \mathbf{w} \rangle \mathbf{w}, \langle S_i, \mathbf{w} \rangle \mathbf{w} \rangle] \\
&= \frac{1}{T} \sum_{i=1}^T [\langle S_i, S_i \rangle - 2\langle S_i, \mathbf{w} \rangle \langle S_i, \mathbf{w} \rangle + \langle S_i, \mathbf{w} \rangle^2 \|\mathbf{w}\|^2] \\
&= \frac{1}{T} \sum_{i=1}^T \|S_i\|^2 - \frac{1}{T} \sum_{i=1}^T \langle S_i, \mathbf{w} \rangle^2
\end{aligned}$$

since $\|\mathbf{w}\|^2 = 1$. Since the first sum depends only on the data, the average error is minimized by maximizing the term $\frac{1}{T} \sum_{i=1}^T \langle S_i, \mathbf{w} \rangle^2$, which is nothing else than $E[\langle S_i, \mathbf{w} \rangle^2]$, and

$$E[\langle S_i, \mathbf{w} \rangle^2] = \text{Var}[\langle S_i, \mathbf{w} \rangle] + E[\langle S_i, \mathbf{w} \rangle]^2 = \text{Var}[\langle S_i, \mathbf{w} \rangle]$$

since the mean of the projection is zero (the data is centered). More generally we want to project S over a subspace of dimension more than 1, and we may assume that this subspace is spanned by orthonormal vectors (any basis can be transferred into an orthonormal basis). The above computations can be repeated to show that we want to maximize the sum of the variances of the projections, which simplifies to maximize $\text{Var}[\langle S_i, \mathbf{w} \rangle]$ in one dimension. Since $\langle S_i, \mathbf{w} \rangle$ is the i th row of the column vector $S\mathbf{w}$, we can write the variance (recall, the mean is 0) as $\frac{1}{T} (S\mathbf{w})^T (S\mathbf{w}) = \mathbf{w}^T \frac{(S^T S)}{T} \mathbf{w}$ and the unit vector $\mathbf{w}$ which maximizes this quantity is the eigenvector of the covariance matrix $\frac{S^T S}{T}$ with largest eigenvalue. This covariance matrix is symmetric, so its eigenvectors will be real vectors orthogonal to each others, and since it is positive semi-definite, the eigenvalues will be non-negative reals. We refer the eigenvectors as *eigengraphs*, and the dimensionality reduction is obtained by treating the smallest eigenvalues as zero.

Singular value decomposition (SVD) of S . The matrix S can be decomposed using SVD into $S = UDV^T$ where U is an orthogonal $T \times T$ matrix, D is a diagonal $T \times N$ matrix containing the singular values of S , and V is an $N \times N$ orthogonal matrix. Now the covariance matrix $\frac{S^T S}{T}$ becomes $\frac{S^T S}{T} = \frac{1}{T} (UDV^T)^T (UDV^T) = \frac{1}{T} V D^2 V^T$. This provides a diagonalization of the covariance matrix, where each column of V is an eigengraph. The projections $SV = UD$ of the data on the principal eigengraphs are called principal components: the j th principal component is the j th column of SV .

Interpretation. Note that $S = UDV^T$ is equivalent to $U^T S = DV^T$. Thus the rows of $U^T S$ are linear combinations in u_{ti} of the rows of S , so the rows of DV^T are in the row space of S and in fact the column space of V is spanning the row space of S . It means that the columns of V , the eigengraphs, can be used as a basis out of which linear combinations of graphs can be obtained. Since V contains eigengraphs as its columns, V^T contains them as its rows. Multiplying by the diagonal matrix D means that $\tilde{V} = DV$ stores in each of its rows a scaled eigengraph. The matrix D thus models the importance of the i th eigengraph (the higher the eigenvalue, more important the eigengraph). Consider next the role of the i th column U_i of U in $U\tilde{V}$: it multiplies the i th scaled eigengraph, more precisely, since the matrix U has rows indexed by time, u_{ti} multiplies the i th scaled eigengraph at time t . The coefficient u_{ti} thus serves as temporal weight at time t of the i th eigengraph.

5 Case Studies

Before discussing the two case studies, we recall the parameters that will be explored.

T	the number of graph samples the larger T , the smaller the time window
$n_i, i \geq 1$	size (number of vertices) of each graph sample the smaller i , the smaller the graph size

5.1 A Bitcoin Subgraph

The first dataset we consider is a subgraph derived from 50 days of openly available Bitcoin transactions for the period of 8th December 2010 to 26 January 2011. The specific period was chosen to illustrate our techniques because it was from a time when Bitcoin had already bootstrapped and was in regular use, and yet mixing of transactions from multiple separate entities was rare/absent, which allows for effective aggregation of wallet addresses [9] (explained next).

Data Preprocessing. Bitcoin transactions are recorded as *inputs* and *outputs*. Outputs from previous transactions that have not yet been utilized are used as inputs for new transactions, leading to outputs that are assigned to certain wallet addresses. The quantum of coins consumed in a transaction needs to be equal to the quantum of outputs. By design, it may be difficult to determine which inputs are assigned to which outputs. Thus, typically [1] a directed (multi-)graph of wallet addresses is constructed by linking each of the input wallet addresses to each of the output wallet addresses. A common heuristic [9] is to consider all the wallet addresses contributing inputs to a transaction to belong to a single user, and furthermore, if there are two outputs, and the smaller output is smaller than all the inputs, and assigned to a yet unused wallet address, then to treat it as a “change” address, and consider it to belong to the same user who controls the input addresses. Such aggregated sets of wallet addresses obtained from different transactions are merged together provided

the sets intersect (based on a transitivity reasoning). Substituting the distinct wallet addresses with a representative per agglomerated wallet address group then yields a new ‘user’ level agglomerated wallet address graph. There can be (and in fact, are) multiple edges between a given representative wallet address pair, and individual edges are associated with time stamp corresponding to the transaction time, based on which the particular edge was created. For our study, we consider the undirected and unweighted but time-stamped edges version of the agglomerated wallet address (multi-)graph, which, given the semantics of the wallet address agglomeration, may be viewed as a user level social network.

Specifically, we had 55295 wallet addresses, 5576 agglomerated groups, which resulted in a graph G with 6286 nodes. To perform graph embeddings on G , we computed the degree/Katz/betweenness centrality of all its 6286 vertices. The nodes with highest centralities were selected. The dimensionality reduction in the graph embeddings depends on what “high” means. We chose three sizes: $n_1 \approx 0.2 - 0.3\%$, $n_2 \approx 0.6 - 0.8\%$, $n_3 \approx 0.8 - 0.9\%$. The values are approximative since they are decided by a centrality threshold: (1) we rank all nodes according to their centrality score, (2) nodes which have high centrality are the most influential and will be included in the embeddings, (3) we set three thresholds, nodes with a centrality higher than the threshold are chosen. In order to compare across centralities, we need to have more or less the same number of nodes, but since many nodes have the same centrality, we keep nodes with the same centrality together, leading to slightly different numbers of nodes depending on the centrality. Since we consider an undirected graph, this gives graphs of dimensions $\frac{n_i^2 - n_i}{2}$ for $i = 1, 2, 3$. The overall period is 50 days, we choose two time samplings: $T = 50$ means that we are looking at a daily sampling, while $T = 100$ zooms into half days.

Fig. 2. Eigenvalues of $\frac{S^T S}{T}$: Rows (first/second) are for $T = 50$ days and $T = 100$ half days. Columns (from left to right) are for degree, Katz and betweenness centralities.

Eigenvalues and Principal Directions. The principal directions (the most important eigengraphs) are those with the highest eigenvalues. The eigenvalues of $\frac{SS^T}{T}$ are shown on Figure 2 (the non-zero eigenvalues of $\frac{SS^T}{T}$ are also that of the covariance matrix $\frac{S^TS}{T}$). Each subplot is a histogram, where the x -axis contains the actual value of the eigenvalues, while the y -axis displays the frequency of each value. The frequency is given for each group size n_1, n_2, n_3 . The key observations are (1) the behavior for the degree and Katz centralities are similar, (2) the group size does not seem to play a prominent role in the distribution of the eigenvalues (6-7 of them are prominent), (3) using $T = 50$ or 100 does not change the overall distribution either.

Temporal Analysis: Parameter Comparisons. Figure 3 shows the temporal importance of the first two eigengraphs, for each centrality. The group size n_1, n_2, n_3 gives temporal timelines so close to each other that one barely notices that there are 6 time series, since the three lines for the first eigengraph (in darker color) are superimposed, and so are the three lines for the second eigengraph. The same holds whether $T = 50$ or 100. This observation suggests that (at least for this specific example graph) the results we obtain from the analysis of a highly compressed graph nevertheless provides similar result as compared to what would have been obtained by analyzing the relatively larger graphs. Having thus validated the consistency of the results, we report the rest of the experiment results based on n_1 .

Fig. 3. Temporal analysis: the temporal importance of the first two eigengraphs. The first row uses 50 days as temporal ranges, the second row uses finer intervals, 100 half-days. Columns (left to right) are for degree, Katz and betweenness centralities.

In Figure 4 (left) we superimpose the temporal importance time-series of the first two eigengraphs for $T = 50$ and $T = 100$ with n_1 . To fit $T = 50$ into the finer

timeline, we repeat each value twice, which results in the step-like time series. Both time series are consistent, with more “peaks” as expected for $T = 100$. We thus consider the finer time granularity $T = 100$ for the rest of our discussions.

Fig. 4. Left: Time series superimposition of 50 days and 100 half-days, for the two main eigengraphs. Right: Time series of the five most important eigengraphs (100 half-days). Rows (top to bottom) are for degree, Katz and betweenness centralities.

Temporal Analysis: Moments of Interest. In Figure 4 (right), the temporal weights for the five most important eigengraphs are shown (with $T = 100$ and n_1). The time points with sudden changes in the temporal weights of any of the eigengraphs suggest that for the corresponding (half day) time slot, the graph structure changed significantly with respect to its prior structure. This temporal characteristic thus can be used for any further analysis of the particular time frames. Alternatively, if there is a time series of graphs, the proposed technique helps identify a subset of these graphs, which can be treated as representatives for all graphs with similar temporal weights, and thus, instead of analyzing all the individual graph instances for their macroscopic properties, one could analyze the representatives. In Figure 5 we show up to top eight distinct centrality scores (if non-trivial ones exist) for half-day time slots during and just before/after which the temporal importance of the five most important eigengraphs eg_i^{type} were highest. We stack the three values together using color gradient, so that the highest values can be compared against the respective values in the adjacent time windows. The subscript index i represents the importance rank of the eigengraphs (eg). Rows (top to bottom) show degree, Katz and betweenness centrality ($type$ indicated with superscripts d , k and b). Temporal weights of several of the eigengraphs may witness the highest value during the same time slot, and these are accordingly indicated and plotted together. For

instance, this is the case for the eigengraphs eg_3^d and eg_4^d for degree centrality since the temporal weights are highest in the half day time slots 88, 66, 85, 85 and 79 respectively for the five eigengraphs. For degree and Katz centrality, we notice that the highest centrality score has significant changes, and the lower ones have varying degrees of changes too, with respect to the immediately preceding and following time slots. For betweenness centrality, the changes are in fact drastic.

Fig. 5. Top centrality scores (top to bottom): degree, Katz, betweenness.

Since there is no ground truth about this network, we next cross-check the results by going back to the graph samples (something which is feasible for small enough graph samples). In Figure 6 we show a snapshot of the graphs for the half-day just before (top row) and during the half-day (bottom row) where temporal importance was highest for eg_5^d , $eg_3^k - eg_5^k$ and $eg_1^b - eg_3^b$ respectively. These correspond to the half-day slots 78-79, 84-85 and 9-10 respectively for the three cases. The nodes with the highest centralities are highlighted (note that multiple nodes may have same high centrality). In each of these instances, the

visual inspection does suggest a significant change in the connectivities, with that for the betweenness indeed being drastic. The connectivities also explain why betweenness did not have eight non-trivial centrality scores for the ‘before’ time slot.

Fig. 6. Half-day time slot just before and during when temporal importance was highest (left to right) for: degree (eg_5^d), Katz ($eg_3^k - eg_5^k$) and betweenness ($eg_1^b - eg_3^b$).

Since nodes correspond to wallet addresses, one could try to see if any address of interest appears. An examination of the addresses with highest degree centralities reveals a wallet address belonging to one of the agglomerated groups, in fact the largest group, which contains 21093 addresses. Further macroscopic as well as microscopic investigations could be conducted over the network, where an analyst may focus on the times of interest which have been identified by the temporal analysis. This becomes a problem of Bitcoin forensics.

5.2 The Enron Email Graph

We next study the Enron corpus [11], which is frequently used for studying many machine learning and knowledge discovery algorithms, including prior work on changes and anomaly detection in time series graphs [15, 14, 8]. Specifically, we use a subset of a curated temporal graph (<http://konect.cc/networks/enron/>) for the time period [1998-01-05, 2002-12-21]. There were 87110 distinct nodes and 298691 edges in the resulting time stamped multi-graph. We use two time samplings: $T = 48$ and $T = 4 \times 48 = 192$, capturing roughly 38 and 10 days long samples respectively. We chose two node sample sizes: $n_1 \approx 0.085\%$, $n_2 \approx 0.17 - 0.18\%$. Similar to the discussion above on Bitcoin network, the exact values were determined as per centrality thresholds.

We next report our results based on graph embeddings using degree and Katz centrality. We note that since computing Katz centrality is much more compu-

Fig. 7. Eigenvalues for the Enron network: Their distributions based on degree and Katz centrality are shown respectively on the left and right. The x -axis contains the values taken by the eigenvalues, while the y -axis measures their frequency.

tationally expensive than degree centrality, we used a truncated Katz centrality to make the computations easily tractable. Figure 7 shows the distribution of the eigenvalues (left: degree, right: Katz) of the obtained graph time series. The distributions are mutually similar, furthermore, the values of the largest eigenvalues are very large (close or up to 400), as compared to the Bitcoin network, which had significantly smaller variation. Interpreting the eigenvalues as relative weights, this suggests that the corresponding top few eigengraphs are even better to approximate (summarize) the whole graph.

Fig. 8. Enron time series of the two most important eigengraphs: on the left for the degree centrality, on the right for Katz centrality. The top figures consider 48 time samples, the bottom figures have 192 times samples corresponding to 10 day periods.

Figure 8 shows the temporal weight of the top two eigengraphs, using different node sample sizes, for the two time sample instances (top: $T = 48$, bottom: $T = 192$), for the case of degree centrality on the right, and of Katz centrality

on the left. But for slight variations, both centralities yield the same behavior. The top figures illustrate that when considering a larger time frame, the main fluctuations are emphasized, while using a 10 day time frame makes evident smaller fluctuations. Both view points are consistent.

Fig. 9. Consistency of Enron time series resolution with respect to the sampling size: time superimposition (left, degree centrality, right Katz centrality): the lower sampling resolution is repeated 4 times to compare with the higher sampling resolution.

Figure 9 shows the superimposition of the two temporal importance for the two time samples (lower sampling resolution value is repeated 4 times) for the most important eigengraph corresponding to n_1 , for degree and Katz centrality respectively on left and right. From these plots we see, similar to our observation from the experiments with the Bitcoin data set, that the analysis is robust across node sample sizes, and the temporal sample resolutions.

Figure 10 shows the temporal importance weights for the top four eigengraphs with respect to degree centrality. We chose degree centrality since it is computationally cheaper to compute than Katz centrality, and previously reported results show no significant difference in the temporal evolution of the respective eigengraphs. The choice of the 4 first eigengraphs is motivated by the finding of Figure 7.

Inflection points (sharp rise/fall of the temporal importance) suggest that the graph structure underwent several marked changes over the span of the five years studied. These peaks within each of the eigenvalue time series reflect changes in the email exchange pattern, which in turn arguably carry signals to detect real world events.

To validate this idea, and to compare our approach, we use the study in [15] as a baseline (we note that their study was based on a set of 151 email users). We first report and analyze our findings visually (see Figure 10). Specifically, we started with the set of events for Enron scandal which [15] enumerates, these are represented by shaded regions (bars) on Figure 10: for each event listed in [15], we found a date corresponding to the event description (some events are correlated and happen on the same day, e.g., 28 November 2001 event of Dynegy calling off its \$8.4B merger with Enron, and consequently Enron stock price plunging below \$1 valuation). Then for each such day highlighted as an event, the 10 day period including this day, followed by another 10 day period are considered as

a period of interest. Concurrent events lead to larger shadowed regions (some events treated as single event in [15] were in fact spread over a period of time, e.g., Skilling testifying on 7th February, Watkins testifying on 14 February). Then in each temporal eigengraph evolution, we set a threshold of 0.02, and for any increase ≥ 0.02 on the y -axis, we identify the following local maximum as a point of interest, represented by dots on the timelines. We then compare the events of the Enron timeline with the points of interest in the eigengraph temporal timelines. Red dots and pink regions mean a match, blue dots mean trigger for events that are detected by our approach which uses eigengraphs, but were not reported in [15], while blue shaded regions mean events detected in [15] but not in the corresponding eigengraph based analysis.

The first observation is that taking into account the eigengraphs corresponding to the four highest eigenvalues lead a match for all but six events: among those, two early events at the end of 1999 and early 2000, and one late event in February 2002 seem really unnoticed in the current analysis, at least using the four first eigengraphs. The other three events are close to detected peaks, whether they are noticed in the current analysis is a matter of choice in tuning the parameters (the threshold of 0.02, and the time window considered around an event, set up to be around 20 days).

Conversely, our approach indicates events at time points where the approach of [15] did not identify any events. Tracing back to historical records of events around the Enron scandal and the associated broader electricity crisis in California during the period, we identified ten further relevant events, some mutually close, which were not accounted for in [15]. The corresponding blue dots in the eigengraph temporal evolutions are circled.

In [8], Goyal et al. looked at embedding methods for dynamic graphs, and at its application to anomaly detection. The Enron data set was also considered in that study. An anomaly was defined as a change in embeddings F between time $t - 1$ and t , defined by $\|F_{t+1}(G_t) - F_t(G_t)\|_F$. Three major anomalies were reported, around Feb 2001, June 2001 and Jan 2002. Goyal et al. associated these to Skilling taking over as CEO in Feb 2001, Rove divesting his stocks in June 2001 and the CEO resignation in January 2002. All these events are also detected by our approach. The study in [15] detected only one of these three events, namely the January 2002 change of CEO.

In the Appendix of this paper, we enumerate the list of events from [15] along with the precise date(s) of those events. Additionally, we also list and highlight (using a distinctive color, and [†] mark) some other significant events that we identified while interpreting the results based on our approach. This list comprises events across 34 distinct dates, and it was created using data obtained from multiple sources, including <https://www.timelines.ws/subjects/Enron.HTML> and <https://www.nytimes.com/2006/01/18/business/worldbusiness/timeline-a-chronology-of-enron-corp.html>. Note that even those sources had many other seemingly less significant events which we have not enumerated here, and each of those individual sources were also incomplete. Thus the list of events presented in the Appendix is not exhaustive.

Fig. 10. Enron time line: from top to bottom, the temporal evolution of the 1st, 2nd, 3rd and 4th eigengraph. Peaks in the timeline are highlighted by dots. Red dots in pink regions represent a match with previous analyses. We provide quantified interpretation of this result in terms of precision and recall metrics respectively in Tables 1 and 2.

Eigengraph	Whole period	Aug 2000 onwards
1	0.81	0.81
2	0.66	0.66
3	0.35	0.5
4	0.6	0.85
Cumulative	0.57	0.70

Table 1. Temporal feature extraction *precision* based on the Enron scandal associated events tabulated in the Appendix

Eigengraph	Whole period	Aug 2000 onwards
1	0.61	0.65
2	0.35	0.37
3	0.26	0.28
4	0.52	0.56
Cumulative	0.79	0.84

Table 2. Temporal feature extraction *recall* based on the Enron scandal associated events tabulated in the Appendix

We report the precision and recall (in Tables 1 and 2 respectively, showing the first two significant digits without rounding) achieved by the proposed temporal feature extraction mechanism for the same set of parameter choices as previously mentioned, namely: graph snapshots of 10 day periods, and a threshold of 0.02 increase in the eigengraph weight to identify a local maximum. We show these results per eigengraph, as well as the one obtained when combining the event triggers or signals obtained from them together (cumulative). The number of triggers per eigengraph correspond to the number of dots (red and blue together) in the corresponding plots, i.e., 16, 12, 14 and 20 respectively. Several eigengraphs provided common triggers. Hence, the cumulative measures are based on 38 unique triggers.

These results need to be interpreted subject to several caveats. Foremost, as mentioned above, the ‘ground-truth’ is not complete, and we evaluate the mechanism based on the 34 event dates tabulated in the Appendix. Furthermore, several of those events were occasionally close to each other. However, our temporal analysis considers graph snapshots for larger (ten days) windows of time. Consequently, multiple real events may fit within a single window. The email network graph analysis does not carry the semantics to discern what was being discussed in the emails, and whether that had any relation with the events that were reported prominently and considered in our list of events. Finally, we also notice that there were hardly much report on events prior to August 2000. Yet, our analysis of the eigengraphs corresponding to the third and fourth eigenvalues suggest several events. Since information for this period is sparse, it is not possible to validate whether something interesting actually was happening or not. Naturally, considering these signals result in poorer precision. We thus

report the precision value considering not only if our analysis is conducted for the whole period, but also for the period from only August 2000 onwards. For consistency, we also report the recall for both the whole period, and for this latter more interesting time period.

6 Concluding Remarks and Implications

In this paper we have proposed an approach to study the temporal characteristics of graphs, combining the ideas of graph embedding, centralities, and principal component analysis (PCA) for dimensionality reduction. This reduction is realized in the graph size dimension from the use of a centrality measure driven k th order proximity, while we derive the dimensionality reduction in the temporal dimension using PCA. It is a macroscopic analysis which helps identify time periods which are similar to each other and time points of interests/changes.

This is useful, because, for the periods where no large changes are occurring, studies of other macroscopic graph properties of a representative sample would suffice to understand the big picture, instead of having to replicate the analysis for each instance in the graph time series.

Likewise, points of inflections suggest significant structural changes in the graph, and help pinpoint specific time instances, where it would make sense to carry out further analyses of the relevant graph samples. For instance, our technique identifies several time points of interest (with significant changes) in the Enron data set, matching in part with the findings of prior studies [15, 8] but also detecting other real world events the previous techniques did not identify.

While we instantiate the embedding with certain centrality measures for our case studies, the approach we propose can readily accommodate other centrality measures. This modularity in turn gives the approach flexibility to determine node importance based on the extent of influence of nodes (localized, e.g., using degree centrality; global, e.g., Katz centrality.) . Our proposed framework, by decoupling the dimensionality reduction of the graph size (using embedding) from the temporal dimension (using PCA), provides a simple yet flexible approach, which gave robust results in our case studies. This suggests that our approach is practical for analyzing dynamic graphs.

Given a dynamic graph to be analyzed, the parameters T and n_i may be tuned accordingly: (1) T decides the level of time granularity, fine if many events are expected, coarse otherwise (it may also be dynamically adjusted), (2) the sizes n_i , $i \geq 1$ depend on the graph structures. If a graph is sparse, these values may be chosen fairly small, as is the case for the Bitcoin networks in our first case study. Alternatively, if the graph structure depends on a few influential nodes, n_i , $i \geq 1$ may also be chosen fairly small, as was the case for the second case study, the Enron network. The extent of compression through embedding which would still carry useful information naturally depends on the overall graph structure, and thus it needs to be determined in a per case basis.

Further experiments with weighted and directed graphs, and applications of our techniques, for example to Bitcoin forensics, comprise one direction for future work. Another direction could be to explore the use of other centralities.

Appendix: Enron scandal - Chronology of events

Timeline	Event (non-exhaustive) summary
1999	
November	Launch of EnronOnline, a global commodity trading web site
2000	
January	Launch of Enron Broadband Service (EBS)
19 July	Announcement of EBS joining forces with Blockbuster
23 August	Enron stock reached all time high price of \$90.75
3 October	Enron attorney discussed Timothy Belden's strategies
1 November	FERC investigation exonerated Enron
December	At end of 2000, Enron declared \$53 million earnings for Broadband
13 December	Announcement that Jeffrey Skilling would take over as CEO
2001	
17 January	Rolling blackouts in California
1 February [†]	State lawmakers legislate to spend up to \$10 billion for power
12 February [†]	Skilling was named CEO of Enron, replacing Lay
23 March	Enron conference call with analysts to boost stock
17 April	The 'asshole' call: Jeffrey Skilling response to an analyst query
15 May [†]	California energy regulators adopted the highest rate increase in the state's history
17 May [†]	California energy regulators uncovered evidence that some electrical power companies repeatedly shut down generating plants for unnecessary maintenance
26 May	Schwarzenegger, Lay, Milken meeting
5 June [†]	Karl Rove divested his stocks in energy, defense and pharmaceutical companies (including Enron).
11/12 July	Quarterly conference call
24-25 July	Skilling met analysts and investors in NY
14 August [†]	Skilling resigned; Lay named CEO again
22 August	Sherron Watkins met Lay to discuss accounting irregularities
16 October [†]	Enron announces \$638 million in third-quarter losses
19 October [†]	Securities and Exchange Commission launches inquiry
23 October [†]	Lay professes confidence in Fastow to analysts
24 October [†]	Fastow ousted
9 November	Dynegy Inc. announced an agreement to buy Enron
19 November	Enron restated its 3rd-quarter earnings disclosing \$690M debt
28 November	Dynegy called off its \$8.4B merger with Enron Enron stock plunged below \$1
2 December	Enron Corp. under CEO Kenneth Lay filed for bankruptcy
2002	
29 January	Stephen Cooper took over as interim Enron CEO
5 February	Lay cancelled senate committee appearance invoking the 5th Fastow, Kopper, Lay invoked the 5th
7 February	Skilling testified Fastow and Kopper invoked the 5th
14 February	Sharon Watkins testified
14 March	Former Enron auditor Arthur Andersen LLP indicted

[†] Events not considered in [15]

References

1. Akcora, C., Kantarcioglu, M., Gel, Y.: Blockchain data analytics. *IEEE International Conference on Data Mining (ICDM)* (2018)
2. Batagelj, V., Praprotnik, S.: An algebraic approach to temporal network analysis based on temporal quantities. *Social Network Analysis and Mining* **6**(28) (2016)
3. Bourgain, J.: On lipschitz embedding of finite metric spaces in hilbert space. *Israel Journal of Mathematics* **52**(1-2), 46–52 (1985)
4. Cai, H., Zheng, V.W., Kevin Chen-Chuan Chang: A comprehensive survey of graph embedding: Problems, techniques and applications. *IEEE Transactions on Knowledge and Data Engineering* **30**, 1616–1637 (2018)
5. Estrada, E.: *The Structure of Complex Networks: Theory and Applications*. Oxford University Press, Inc., USA (2011)
6. Fraiman, D., Fraiman, N., Fraiman, R.: Nonparametric statistics of dynamic networks with distinguishable nodes. *Test* **26**(3), 546–573 (2017)
7. Goyal, P., Ferrara, E.: Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems* **151**, 78–94 (July 2018)
8. Goyal, P., Kamra, N., He, X., Liu, Y.: Dyngem: Deep embedding method for dynamic graphs. *arXiv preprint arXiv:1805.11273* (2018)
9. Harrigan, M., Fretter, C.: The unreasonable effectiveness of address clustering. *Tech. Rep. arXiv:1605.06369, arXiv* (2016)
10. Holme, P., Saramäki, J. (eds.): *Temporal Networks*. Springer (2013)
11. Klimt, B., Yang, Y.: The enron corpus: A new dataset for email classification research. In: *European Conference on Machine Learning* (2004)
12. Kondor, D., Csabai, I., Szüle, J., Pósfai, M., Vattay, G.: Inferring the interplay between network structure and market effects in bitcoin. *New J. Phys.* **16** 125003 (2014)
13. Narayanan, A., Chandramohan, M., Venkatesan, R., Chen, L., Liu, Y., Jaiswal, S.: graph2vec: Learning distributed representations of graphs. *Workshop on mining and learning with graphs* (2017)
14. Park, Y., Priebe, C.E., Youssef, A.: Anomaly detection in time series of graphs using fusion of graph invariants. *IEEE Journal of Selected Topics in Signal Processing* **7**(1) (February 2013)
15. Peel, L., Clauset, A.: Detecting change points in the large-scale structure of evolving networks. In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (2015)
16. Sarkar, S., Guo, R., Shakarian, P.: Using network motifs to characterize temporal network evolution leading to diffusion inhibition. *Social Network Analysis and Mining* **9**(14) (2019)
17. Sirovich, L., Kirby, M.: Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A.* **4**(3), 519–524 (1987)
18. Ye, C., Wilson, R., Hancock, E.: Network analysis using entrop component analysis. *Journal of Complex Networks* (2017)
19. Zhu, L., Guo, D., Yin, J., Ver Steeg, G., Galstyan, A.: Scalable temporal latent space inference for link prediction in dynamic social networks. *IEEE Transactions on Knowledge and Data Engineering* **28**(10), 2765–2777 (2016)