Change Point Detection in Weighted and Directed Random Dot Product Graphs

Federico Larroca*, Paola Bermolen*, Marcelo Fiori*, and Gonzalo Mateos†

*Facultad de Ingeniería, Universidad de la República, Uruguay
Email: {flarroca,paola,mfiori}@fing.edu.uy
†Dept. of Electrical and Computer Engineering, University of Rochester, Rochester, NY, USA.
Email: gmateosb@ur.rochester.edu

Abstract—Given a sequence of possibly correlated randomly generated graphs, we address the problem of detecting changes on their underlying distribution. To this end, we will consider Random Dot Product Graphs (RDPGs), a simple yet rich family of random graphs that subsume Erdös-Rényi and Stochastic Block Model ensembles as particular cases. In RDPGs each node has an associated latent vector and inner products between these vectors dictate the edge existence probabilities. Previous works have mostly focused on the undirected and unweighted graph case, a gap we aim to close here. We first extend the RDPG model to accommodate directed and weighted graphs, a contribution whose interest transcends change-point detection (CPD). A statistic derived from the nodes' estimated latent vectors (i.e., embeddings) facilitates adoption of scalable geometric CPD techniques. The resulting algorithm yields interpretable results and facilitates pinpointing which (and when) nodes are acting differently. Numerical tests on simulated data as well as on a real dataset of graphs stemming from a Wi-Fi network corroborate the effectiveness of the proposed CPD method.

Index Terms—Change-point detection, graph representation learning, node embeddings, wireless networks.

I. Introduction

Consider the problem of remotely managing a Wi-Fi network. Once the network is installed, it is not uncommon to experience unforeseen interference – Access Points (APs) are moved without authorization or even that the surroundings change. Instead of waiting for the users to report problems, a proactive monitoring approach would analyze the time-series of acquired power measurement between APs (typically available in all large-scale deployments) looking for persistent changes in the mean. However, for a network with n APs there may be up to $n(n-1)$ time-series to analyze, which may not be practical even for modest values of n [1]. Furthermore, analyzing each time-series separately does not take into account the underlying structure of the data, namely spatial correlations naturally modeled via a weighted and directed graph.

This wireless network monitoring problem exemplifies the usefulness of statistical analysis of dynamic graphs, in particular detecting changes on the underlying distribution generating said networks. Further applications include the analysis of functional brain networks [2], social networks [3], and neuronal activity [4], just to name a few. There exists basically two approaches to addressing change-point detection (CPD)

in dynamic graphs. The first one is to extract a vector representation from each graph and apply standard (geometric) CPD algorithms on the resulting time-series [5]. However, applicability of these techniques may be hindered by the often limited interpretability and potential loss of information of the embedding, as well as due to a lack of theoretical guarantees.

The second approach to CPD in graphs, and the one we explore here, is to resort to probabilistic generative models, for which theoretically-sound results may be derived. Although classic ensembles such as Erdös-Rényi (ER) or Stochastic Block Models (SBM) have been used in the past [6], we resort to the more general Random Dot Product Graph (RDPG) model [7], [8]. In RDPGs each node has an associated latent vector in $\mathbb{R}^d$, and the probability of a pair of nodes having an edge is simply the inner product between the corresponding vectors. Even though these vectors may be interpreted as node embeddings, they are directly related to the generative process, and are not an (arbitrary) summary of graph structure (as in traditional graph representation learning). As we discuss in the next section, RDPGs capture phenomena commonly encountered with real-world graphs (e.g., statistical dependencies among edges) and subsume both ER and SBM as special cases, while still being amenable to analysis. Moreover, RDPGs offer interpretability, an attractive feature that simplifies the explanation of the detected change-points.

Our main contribution is to extend the vanilla RDPG model for undirected and unweighted graphs [4], [7] and adapt it to perform CPD on directed and weighted networks (Sec. III). Extensions to directed graphs are relatively straightforward (Sec. II-B), but we carefully study those ambiguities inherent to the model (not discussed in previous work) which may challenge downstream CPD methods. Our extension of RDPGs to the weighted case (Sec. II-C) is totally new, and unlike previous efforts in this direction [9], [10], our non-parametric approach does not require *a priori* specification of the weights' distribution to perform inference and estimation. We believe this contribution has value on its own, and beyond CPD it can e.g., impact graph classification and visualization of networks.

II. Random Dot Product Graphs

A. Vanilla RDPG

Let $G = (V, E)$ denote an unweighted and undirected graph, where $V = \{1, \dots, n\}$ are the nodes and $E \subseteq V \times V$ are

the edges. In the RDPG model each node $i \in V$ has an associated column vector $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d$, and edge (i, j) exists with probability $\mathbf{x}_i^T \mathbf{x}_j$ (a particular case of the latent space model [11]). The geometric interpretation is that nodes with large $\|\mathbf{x}_i\|$ tend to have higher connectivity, whereas a small angle between $\mathbf{x}_i$ and $\mathbf{x}_j$ indicates higher "affinity". Note that the set $\mathcal{X}$ of possible $\mathbf{x}_i$ is such that $\mathbf{x}^T \mathbf{y} \in [0, 1] \, \forall \, \mathbf{x}, \mathbf{y} \in \mathcal{X}$. In turn, vectors $\mathbf{x}_i$ may be random with distribution $F(\mathbf{x})$, where $F(\mathbf{x}) = 0 \, \forall \, \mathbf{x} \notin \mathcal{X}$.

Thus, letting $\mathbf{A} \in \{0, 1\}^{n \times n}$ be the symmetric adjacency matrix of G and $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$, the RDPG model specifies

$$P\left(\mathbf{A} \mid \mathbf{X}\right) = \prod_{i < j} (\mathbf{x}_i^T \mathbf{x}_j)^{A_{ij}} (1 - \mathbf{x}_i^T \mathbf{x}_j)^{1 - A_{ij}}. \quad (1)$$

That is, *given* $\mathbf{X}$, edges are conditionally independent with $A_{ij} \sim \text{Ber}(\mathbf{x}_i^T \mathbf{x}_j)$. For instance, if $\mathbf{x}_i = \sqrt{(p/d)}\mathbf{1} \; \forall i$, we obtain an ER graph with edge probability p. Correlation between edges is induced by the choice of the possible vectors $\mathbf{x}_i$ and the hierarchical nature of the model. For instance, an SBM with M communities may be generated by restricting $\mathbf{X}$ to having only (at most) M different columns (i.e. $|\mathcal{X}| = M$). Several additional examples are discussed in [8].

Given a graph stemming from an RDPG with adjacency matrix $\mathbf{A}$, we now discuss how to estimate the matrix $\mathbf{X}$. The key intuition is that $\mathbf{A}$ is a noisy observation of

$$\mathbf{P} = \mathbf{X}\mathbf{X}^T, \quad (2)$$

the matrix of connections probabilities, since $\mathbb{E}\left[\mathbf{A}\right] = \mathbf{P}$. We thus adopt the estimator $\hat{\mathbf{X}} = \text{argmin}_{\mathbf{X}} \|\mathbf{A} - \mathbf{X}\mathbf{X}^T\|_F^2$, s. to $\text{rank}(\mathbf{X}) = d$. The solution is readily given by

$$\hat{\mathbf{X}} = \hat{\mathbf{Q}}\hat{\mathbf{\Lambda}}^{1/2}, \quad (3)$$

where $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$ is the spectral decomposition of $\mathbf{A}$, $\hat{\mathbf{\Lambda}} \in \mathbb{R}^{d \times d}$ is a diagonal matrix with the d largest eigenvalues, and $\hat{\mathbf{Q}} \in \mathbb{R}^{n \times d}$ are the corresponding d dominant eigenvectors. We are assuming that $\hat{\mathbf{\Lambda}}$ has only non-negative values, a limitation that may be easily circumvented [12]. In practice, d is likely unknown but can be estimated by looking for "elbows" on the so-termed scree plot (an ordered plot of the eigenvalues) [13]. Estimator (3) defines the so-called Adjacency Spectral Embedding (ASE), which approaches the actual $\mathbf{X}$ as $n \to \infty$ provided the true d is chosen [8].

An important aspect of the RDPG model is that, by definition, it is invariant to rotations of $\mathbf{X}$. To see this, consider an orthogonal matrix $\mathbf{W} \in \mathbb{R}^{d \times d}$, and note that the rotated vectors $\mathbf{X}\mathbf{W}$ will produce the same probability matrix as in (2) because $\mathbf{X}\mathbf{W}(\mathbf{X}\mathbf{W})^T = \mathbf{X}\mathbf{W}\mathbf{W}^T\mathbf{X} = \mathbf{X}\mathbf{X}^T = \mathbf{P}$. This implies the estimator (3) is unbiased up to an unknown rotation matrix $\mathbf{W}$, and the ambiguity should be accounted for when detecting changes on the graph's distribution.

B. Directed RDPG

As defined before, the RDPG model is only suitable for undirected graphs. Indeed, $\mathbf{X}\mathbf{X}^T = \mathbf{P}$ is always symmetric. Directed graphs require an adaptation to the model [14], where each node $i \in V$ has an associated column vector $\mathbf{x}_i$ – now in $\mathbb{R}^{2d}$. Let us denote by $\mathbf{x}_i^l$ and $\mathbf{x}_i^r$ the first and last d coordinates of $\mathbf{x}_i$ respectively, and by $\mathbf{X}^l, \mathbf{X}^r \in \mathbb{R}^{n \times d}$ the matrices stacking the transposed nodal vectors as their rows. Analogously to the vanilla case we define the directed RDPG (D-RDPG) model as $P\left(\mathbf{A} \mid \mathbf{X}\right) = \prod_{i \neq j} [(\mathbf{x}_i^l)^T \mathbf{x}_j^r]^{A_{ij}} [1 - (\mathbf{x}_i^l)^T \mathbf{x}_j^r]^{1 - A_{ij}}$ [cf. the product over all $i \neq j$ here versus $i < j$ in (1)], and the probability matrix is now

$$\mathbf{P} = \mathbf{X}^l (\mathbf{X}^r)^T. \quad (4)$$

We thus basically have two vectors per node, where $\mathbf{x}_i^l$ models node i's outgoing connectivity and $\mathbf{x}_i^r$ its incoming one [as the probability of existence of the directed link (i, j) is given by $(\mathbf{x}_i^l)^T \mathbf{x}_j^r$].

Note that the rotational ambiguity is still present; i.e., given an orthogonal matrix $\mathbf{W} \in \mathbb{R}^{d \times d}$, the rotated vectors $\mathbf{X}^l \mathbf{W}$ and $\mathbf{X}^r \mathbf{W}$ produce the same $\mathbf{P}$. However, the ambiguity is exacerbated in this case, and *any invertible* matrix $\mathbf{W}$ will produce the same $\mathbf{P}$. To see this, consider $\mathbf{X}^l \mathbf{W}$ and $\mathbf{X}^r \mathbf{W}^{-T}$ and note that $\mathbf{X}^l \mathbf{W}(\mathbf{X}^r \mathbf{W}^{-T})^T = \mathbf{X}^l \mathbf{W}\mathbf{W}^{-1}(\mathbf{X}^r)^T = \mathbf{X}^l (\mathbf{X}^r)^T = \mathbf{P}$.

Thus, as stated the D-RDPG model in (4) will be extremely difficult to interpret, particularly when comparing two graphs and their corresponding embeddings. In order to have roughly the same level of ambiguity as in the vanilla RDPG case, we will require that the d columns of both $\mathbf{X}^l$ and $\mathbf{X}^r$ are orthogonal vectors (i.e. $(\mathbf{X}^l)^T \mathbf{X}^l$ and $(\mathbf{X}^r)^T \mathbf{X}^r$ are $d \times d$ diagonal matrices). This extra requirement does not constrain the expressiveness of the model (matrix $\mathbf{P}$ is still of rank d), but it does limit the ambiguity introduced by $\mathbf{W}$.

All in all, we are left with the same rotation ambiguity as in the vanilla RDPG, in addition to a scaling one. To see this, consider a diagonal matrix $\text{diag}(\boldsymbol{\alpha})$ with non-zero entries and $\mathbf{W}$ an orthogonal matrix. Then it follows that $\mathbf{X}^l \mathbf{W}\text{diag}(\boldsymbol{\alpha})$ and $\mathbf{X}^r \mathbf{W}\text{diag}(\boldsymbol{\alpha})^{-1}$ (which still have orthogonal columns) will produce the same $\mathbf{P}$ as (4). Consequently, comparing the magnitude of $\mathbf{x}_i^l$ with that of $\mathbf{x}_i^r$ is meaningless. This scaling ambiguity, which to the best of our knowledge was overlooked before, will challenge CPD if one is interested in the behavior in a single direction (either incoming or outgoing). This is an interesting extension we will leave for future work.

Let us now discuss how to estimate the matrices $\mathbf{X}^l$ and $\mathbf{X}^r$. Since $\mathbf{P} = \mathbb{E}\left[\mathbf{A}\right]$ still holds, we will again look for the pair $\{\hat{\mathbf{X}}^l, \hat{\mathbf{X}}^r\}$ having orthogonal columns such that $\hat{\mathbf{X}}^l (\hat{\mathbf{X}}^r)^T$ is the best rank-d approximant of $\mathbf{A}$. Letting $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ be the singular-value decomposition of $\mathbf{A}$, we set

$$\hat{\mathbf{X}}^l = \hat{\mathbf{U}}\hat{\mathbf{D}}^{1/2} \text{ and } \hat{\mathbf{X}}^r = \hat{\mathbf{V}}\hat{\mathbf{D}}^{1/2}. \quad (5)$$

Note that (5) satisfies the required orthogonality constraint. The choice in terms of scaling and counterscaling of columns is clearly arbitrary. Using $\hat{\mathbf{D}}^{1/2}$ simply assumes that edges are equally generated by the outgoing and incoming connectivity of the nodes. We may have multiplied and divided each entry in $\hat{\mathbf{D}}$ by the same non-zero value and used the resulting decomposition instead.

C. Weighted RDPG

We now discuss how to extend the RDPG to the weighted case. Let us then define a positive weight for each edge (i.e., a map $w : E \mapsto \mathbb{R}^+$), where the absence of an edge is encoded as 0. Naturally, an unweighted graph is a particular case of a weighted graph where weights are either 0 or 1.

Two previous works have proposed adaptations of the RDPG to the weighted case [9], [10], both of which are basically identical and we outline now. Assume that the weights are generated from a given parametric distribution $G_\theta(w)$ (with $\boldsymbol{\theta} \in \mathbb{R}^M$, for instance $\theta = \lambda$ for a Poisson distribution). Each node i now has M vectors $\mathbf{x}_i[m] \in \mathbb{R}^{d_m}$ ($m = 1, \dots, M$) such that the weight w_{ij} between nodes i and j is random with parametric distribution $G_{(\mathbf{x}_i^T[1]\mathbf{x}_j[1],\dots,\mathbf{x}_i^T[M]\mathbf{x}_j[M])}(w_{ij})$, independently of all other edges. One recovers the vanilla RDPG by considering a Ber(θ) distribution.

This extension has several drawbacks. For starters, all edges are required to have the same weight distribution, albeit with different parameters. This limitation may be partially overcome by considering a mixture distribution. However, and limiting even more its applicability, $G_\theta(w)$ has to be chosen *a priori*. So if edges have different weight distributions, we would have to know how many of them have each distribution before proceeding to inference.

We propose instead that the sequence of vectors associated to each node is related to the moment generating function of the weight distribution. In particular, each node has a sequence of column vectors $\mathbf{x}_i[m] \in \mathbb{R}^{d_m}$ (for $m \in \mathbb{N}^+$), and the (now non-parametric) weight distribution $G(w_{ij})$ is such that

$$\mathbb{E}\left[e^{tw_{ij}}\right] = \sum_{m=0}^{\infty} \frac{t^m \mathbb{E}\left[w_{ij}^m\right]}{m!} = 1 + \sum_{m=1}^{\infty} \frac{t^m \mathbf{x}_i^T[m]\mathbf{x}_j[m]}{m!}. \tag{6}$$

In words, we model the m-th order statistic of the weight w_{ij} between nodes i and j via the inner product $\mathbf{x}_i[m]^T \mathbf{x}_j[m]$. One can recover the vanilla RDPG by setting $\mathbf{x}_i[m] = \mathbf{x}_i \, \forall m$, where $\mathbf{x}_i$ is the vector associated to node i on the vanilla case. Moreover, vectors $\mathbf{x}_i[m]$ are estimable as in the vanilla case, as an ASE of the matrix $\mathbf{A}^{(m)} = [w_{ij}^m]$ provides an unbiased estimator of the matrix $\mathbf{X}[m]$.

To illustrate the discriminative power of this novel embedding, we consider a weighted SBM graph, where edges exist with fixed probability $p = 0.5$, but weights are Gaussian with mean $\mu = 5$ and standard deviation $\sigma = 0.1$ for all edges except between a group of nodes where the distribution is Poisson with parameter $\lambda = 5$ (meaning that weights have the same mean). The vectors $\hat{\mathbf{x}}[m]$ corresponding to the ASE for $m = 1, 2, 3$ and $d_m = 2$ are shown in Fig. 1. Note how the nodes are indistinguishable for $m = 1$. Indeed, the $\hat{\mathbf{x}}_i[1]$ vectors are, as expected, centered around $(\sqrt{\mu p}, 0) = (\sqrt{\lambda p}, 0) \approx (1.58, 0)$ corresponding to the mean weight. For $m = 2$ the vectors are now centered around $(\sqrt{p(\mu^2 + \sigma^2)}, 0) \approx (3.5, 0)$ and $(\sqrt{p(\lambda^2 + \lambda)}, 0) \approx (3.9, 0)$. However, the noise corrupting the estimates hinders the ability
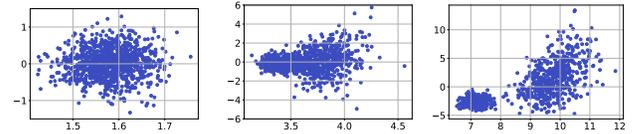
Fig. 1. ASE embedding of $\mathbf{A}^{(m)}$ for the case with Gaussian and Poisson-distributed weights for $d_m = 2$ and $m = 1$ (left), 2 (center), and 3 (right). Nodes with different weight distributions are only revealed for $m = 3$.

to distinguish both distributions. For $m = 3$, where the skewness of the distribution comes into play, nodes are clearly separated into two groups.

III. Change-Point Detection

A. CPD for Vanilla RDPG

Let us now consider a sequence of adjacency matrices $\mathbf{A}_t \in \{0,1\}^{n \times n}$, $t = 1, \dots, T$, generated by a vanilla RDPG. The goal is to detect changes on the underlying distribution of $\mathbf{A}_t$. In this context, and under mild conditions, this is equivalent to detecting changes on the distribution of the vectors $(\mathbf{x}_i)_t$ [4]. We have then that for a given t, each vector is random with a certain distribution $(F(\mathbf{x}))_t$, where $(F(\mathbf{x}))_t = (F(\mathbf{x}))_{t+1}$ except when $t+1$ constitutes a change-point. Furthermore, vectors are not necessarily re-drawn at each time-step (unless, naturally, it is a change-point), thus inducing temporal correlation.

A first CPD approach would be to estimate the matrices $\hat{\mathbf{X}}_t$, and try to detect changes on the estimated vectors' distribution. However, the rotational ambiguity we have discussed in Sec. II implies the embeddings are not necessarily aligned for different values of t. As a workaround one could attempt to solve a so-called Procrustes problem, but as T increases this alignment method may represent an insurmountable computational burden.

The alternative is to consider a proxy to the vectors $(\mathbf{x}_i)_t$ that is unambiguous. In particular, the matrix $\hat{\mathbf{Y}}_t = \hat{\mathbf{X}}_t \hat{\mathbf{X}}_t^T$ (i.e., the estimated expected value of the adjacency matrix) is clearly invariant to rotations of the vectors. Furthermore, and as discussed in [4], it is not necessary to consider the whole matrix, but only entries that do not share nodes (for instance, those corresponding to indices $(i,j) \in \mathcal{O} = \{(i, i+n/2) \, \forall i = 1, \dots, n/2\}$), thus resulting in $n/2$ independent observations for each t. This simplifies proving consistency of the resulting CPD algorithm without sacrificing statistical accuracy, all while avoiding unnecessary computational burden.

All in all, for each $t = 1, \dots, T$ we now have a set of $n/2$ i.i.d. random variables (the entries i,j of $\hat{\mathbf{Y}}_t$ for $(i,j) \in \mathcal{O}$), and we would like to detect at what values of t (if any) does the underlying distribution change. We are thus facing a standard CPD problem, for which several algorithms exist [15]. In particular, for the experiments of Sec. IV we will use a variation of the CUSUM algorithm of [4].

B. CPD for Directed RDPG

In the D-RDPG case we have that $\mathbf{P}_t = \mathbf{X}_t^l (\mathbf{X}_t^r)^T$ for $t = 1, \dots, T$. Recall that on top of the rotational ambiguity,

we also have a column scaling-counterscaling indeterminacy on the factor matrices. Moreover, we have to estimate two matrices per time-step so if aligning $\hat{\mathbf{X}}_t$ was challenging in the vanilla RDPG case, in the directive case the challenges are compounded.

We thus proceed as before and consider $\hat{\mathbf{Y}}_t = \hat{\mathbf{X}}_t^l(\hat{\mathbf{X}}_t^r)^T$. Assuming the outgoing behavior of nodes is independent of the incoming one (i.e., the first d entries of $(\mathbf{x}_i)_t \in \mathbb{R}^{2 \times d}$ are independent of the last d ones), we may also sample entries $(i,j) \in \mathcal{O}' = \{(j+n/2, j) \, \forall j = 1, \ldots, n/2\}$ (in addition to those in $\mathcal{O}$).

C. CPD for Weighted RDPG

In the weighted case, a naive idea would be to consider the matrices $\hat{\mathbf{Y}}_t[m] = \hat{\mathbf{X}}_t[m]\hat{\mathbf{X}}_t^T[m]$ (or the directed counterpart if needed) and run a standard CPD algorithm for each m separately. However, recall from the example in Sec. II-C that as m increases, $\hat{\mathbf{Y}}[m]$ incorporates more features from the underlying weights' distribution. In general, the particular application will dictate which values of m would be most informative. For instance, if we only care about connectivity and mean weights, then $m = 1$ suffices. If we are interested in detecting more fine-grained changes in the distribution, then we would consider higher values of m.

Considering several values of m would only be useful to detect particular changes that produce the same m-th moment. Going back to the example in Sec. II-C, we would not detect any changes by considering only $\hat{\mathbf{Y}}[1]$ if p is halved while λ and μ are doubled at the same t, which may be remedied by additionally considering $\hat{\mathbf{Y}}[2]$. In our simulations we will consider a single m (thus ignoring this kind of situations), but it is an interesting problem to characterize for which scenarios it is necessary to consider several values of m and how to efficiently combine the resulting matrices $\hat{\mathbf{Y}}[m]$.

IV. NUMERICAL TESTS

Let us now present some numerical examples. In particular, we have adapted the CPD algorithm described in [4]. Our code is available at https://github.com/git-artes/cpd_rdpg.
Simulated data. We begin by considering a sequence of $T = 110$ simulated graphs generated by a weighted and directed SBM. In particular, assume two blocks (each of size 50 nodes) with community connection probabilities equal to $\mathbf{Q} = \left(\begin{smallmatrix} 0.5 & 0.3 \\ 0.3 & 0.03 \end{smallmatrix} \right)$. The weights are Gaussian until $t = 80$, when the weights from community 2 to community 1 become Poisson. The mean weight is 5.0 until $t = 30$, when the mean weight from community 1 to 2 becomes 3.0. Finally, the standard deviation is equal to 0.1, until $t = 50$ when it becomes $\sqrt{5}$ for the weights from community 2 to 1. Note that this last change is such that at $t = 80$ the mean and the standard deviation remain constant.

Fig. 2 shows the corresponding $\hat{\mathbf{Y}}_t[4]$ for the entries $(i,j) \in \mathcal{O}$. As expected, the most challenging change-point is at $t = 80$, which the algorithm detects but with an offset in the localization (see the vertical dashed line at $t = 76$). All other change points are perfectly detected and localized.
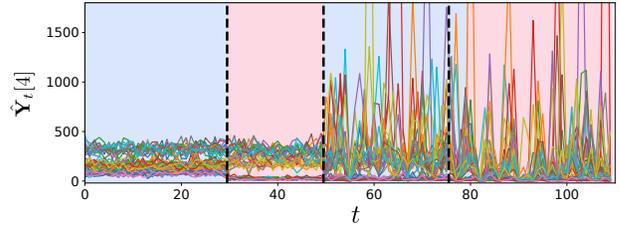
Fig. 2. The evolution of $\hat{\mathbf{Y}}_t[4]$ for entries $(i,j) \in \mathcal{O}$ in the directed and weighted SBM example. The background color indicates the real change-points and the vertical lines the estimation.

Given the scarcity of generative models for weighted graphs, the most common alternative when using probabilistic methods to perform CPD is to apply a threshold th to the weights and fall back to the unweighted case. This threshold may be simply $th = 0$ as in [6], which would detect changes on the connection probability only (none in this example). Positive values of this threshold are naturally possible, but would detect changes on the weight's CDF evaluated at th (i.e. on $G(th)$) which is not very informative, and adds the problem of choosing the value(s) of th (more on this below).

On the other hand, our approach detects changes on several moments of the weight's distribution depending on the choice of m. The next example illustrates how lower values of m may be interesting for certain applications and how the interpretability of the RDPG models (as opposed to methods such as those proposed in [5]) may be further leveraged.
Real data. Received Signal Strength Indicator (RSSI) measurements between APs on a school are obtained from the dataset described in [16]. In this particular example we considered a network consisting of $n = 6$ APs, with measurements collected hourly during almost four weeks, spanning from 10/17/2018 to 11/13/2018 (corresponding to $T = 655$ graphs) where the AP corresponding to node 3 was moved on 10/30/2018. As RSSI is measured in dBm (and are negative), we have first added an offset of 91 to all weights so that they become positive (as -90 dBm is the smallest RSSI measurement in this case) and that larger values still mean "stronger" edges. We have thus a directed (as power measurements between APs are not necessarily symmetric) and weighted graph.

We are interested in connectivity and mean values, so we focus on $m = 1$. The values of $\hat{\mathbf{Y}}_t[1]$ for entries $(i,j) \in \mathcal{O}$ are shown in Fig. 3. Note that we are using only samples in $\mathcal{O}$ since for this application it is not prudent to assume independence among the incoming and outgoing behaviours. Fig. 3 shows a noticeable change-point at $t = 310$, which is correctly detected and localized by the algorithm. This point corresponds to around noon of 10/30/2018, which verifies the correctness of our method. As we discussed before, using a threshold to perform CPD in this case only yields reasonable results for specific values of th; see Fig. 3.

In addition to CPD, a valuable feature of RDPG and its variants is their easy interpretability. To illustrate this
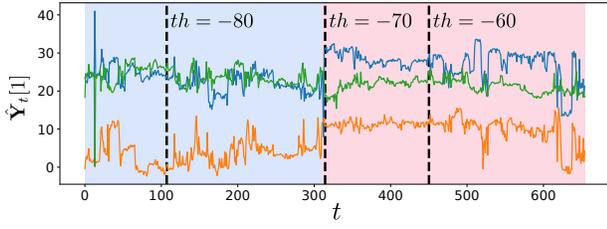
Fig. 3. The evolution of $\hat{\mathbf{Y}}_t[1]$ for entries $(i,j) \in \mathcal{O}$ in the RSSI graph. The background color indicates the change-point estimated through our embedding and the vertical lines by applying different thresholds th to the graph.
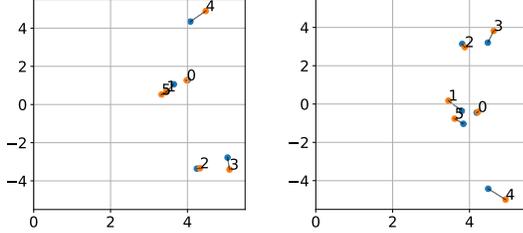
Fig. 4. $\hat{\mathbf{X}}_1^l$ and $\hat{\mathbf{X}}_2^l$ on the left and $\hat{\mathbf{X}}_1^r$ and $\hat{\mathbf{X}}_2^r$ on the right (for $d = 2$, latent vectors corresponding to $\bar{\mathbf{A}}_1$ and $\bar{\mathbf{A}}_2$ respectively). Vectors corresponding to the same node are joined by an arrow. $|\mathbf{x}_3^l|$, $|\mathbf{x}_4^l|$, $|\mathbf{x}_3^r|$ and $|\mathbf{x}_4^r|$ increase after the change-point.

attribute, let us average all adjacency matrices for $t \in [0, 309]$ and $t \in [310, 654]$, resulting in matrices $\bar{\mathbf{A}}_1$ and $\bar{\mathbf{A}}_2$, and analyze the resulting latent positions. In order to avoid the rotation ambiguities, we have used the so-called Omnibus Embedding [17], which in this case amounts to performing ASE to $\mathbf{M} = \begin{pmatrix} \bar{\mathbf{A}}_1 & (\bar{\mathbf{A}}_1 + \bar{\mathbf{A}}_2)/2 \\ (\bar{\mathbf{A}}_1 + \bar{\mathbf{A}}_2)/2 & \bar{\mathbf{A}}_2 \end{pmatrix}$. This approach is only practical when jointly embedding a few adjacency matrices (two here), as the size of $\mathbf{M}$ increases rapidly with the number of matrices considered. The resulting vectors ($d = 2$) are depicted in Fig. 4, where an arrow shows the changes between the embeddings of $\bar{\mathbf{A}}_1$ and $\bar{\mathbf{A}}_2$. Notice how the largest changes, particularly on the outgoing behaviour, correspond to nodes 4 and (to a lesser extent) 3. Since $\mathbf{x}_3^l$ and $\mathbf{x}_4^r$ (as well as $\mathbf{x}_4^l$ and $\mathbf{x}_3^r$) are aligned and both increase their magnitude (differently to node 2), an increase in the mean weight between these two nodes is the main reason behind the change-point. The scaling ambiguity we discussed in Sec. II-C obscures which of the two APs was actually moved.

V. CONCLUSIONS AND FUTURE WORK

We have presented a generative model based CPD algorithm for weighted and directed graphs. In particular, we have considered the Random Dot Product Graph model, which was originally proposed for unweighted and undirected graphs. We have thus presented how to extend the model to directed and weighted graphs. Although the extension to directed graphs was studied before (here we take a closer look at model identifiability), our generalization to weighted graphs is more general and useful than previous proposals, and thus it represents a significant contribution beyond CPD.

There are several important extensions still worth investigating. For instance, it would be interesting to consider the case when the number of nodes changes over time. Since the distribution of the vectors is affected by n (larger values of n produce estimates $\hat{\mathbf{x}}_i$ with less variance), the techniques we presented here may not be applied as is [18]. Another interesting extension would be to detect changes only on a certain direction (outgoing or incoming connections) while taking into account the scaling ambiguity we presented.

REFERENCES

[1] G. Mateos and K. Rajawat, "Dynamic network cartography: Advances in network health monitoring," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 129–143, 2013.

[2] A. N. Khambhati, K. A. Davis, T. H. Lucas, B. Litt, and D. S. Bassett, "Virtual cortical resection reveals push-pull network control preceding seizure evolution," *Neuron*, vol. 91, no. 5, pp. 1170 – 1182, 2016.

[3] P. Sarkar and A. W. Moore, "Dynamic social network analysis using latent space models," in *Advances in Neural Information Processing Systems 18*, Y. Weiss, B. Schölkopf, and J. C. Platt, Eds., pp. 1145–1152. MIT Press, 2006.

[4] O. H. M. Padilla, Y. Yu, and C. E. Priebe, "Change point localization in dependent dynamic nonparametric random dot product graphs," *arXiv:1911.07494 [stat.ME]*, 2019.

[5] D. Zambon, C. Alippi, and L. Livi, "Change-point methods on a sequence of graphs," *IEEE Trans. Signal Process*, vol. 67, no. 24, pp. 6327–6341, 2019.

[6] L. Peel and A. Clauset, "Detecting change points in the large-scale structure of evolving networks," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. 2015, AAAI'15, p. 2914–2920, AAAI Press.

[7] S. J. Young and E. R. Scheinerman, "Random dot product graph models for social networks," in *Algorithms and Models for the Web-Graph*, Anthony Bonato and Fan R. K. Chung, Eds., Berlin, Heidelberg, 2007, pp. 138–149, Springer Berlin Heidelberg.

[8] A. Athreya, D. E. Fishkind, M. Tang, C. E. Priebe, Y. Park, J. T. Vogelstein, K. Levin, V. Lyzinski, and Y. Qin, "Statistical inference on random dot product graphs: A survey," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 8393–8484, Jan. 2017.

[9] R. Tang, M. Tang, J. T. Vogelstein, and C. E. Priebe, "Robust estimation from multiple graphs under gross error contamination," *arXiv:1707.03487 [stat.ME]*, 2017.

[10] D. R. DeFord and D. N. Rockmore, "A random dot product model for weighted networks," *arXiv:1611.02530 [stat.AP]*, 2016.

[11] P. D. Hoff, A. E. Raftery, and M. S. Handcock, "Latent space approaches to social network analysis," *J. Am. Stat. Assoc.*, vol. 97, no. 460, pp. 1090–1098, 2002.

[12] P. Rubin-Delanchy, J. Cape, M. Tang, and C. E. Priebe, "A statistical interpretation of spectral embedding: the generalised random dot product graph," *arXiv:1709.05506 [stat.ML]*, 2017.

[13] Mu Zhu and Ali Ghodsi, "Automatic dimensionality selection from the scree plot via the use of profile likelihood," *Comput Stat Data Anal*, vol. 51, no. 2, pp. 918 – 930, 2006.

[14] C. E. Priebe, Y. Park, M. Tang, A. Athreya, V. Lyzinski, J. T. Vogelstein, Y. Qin, B. Cocanougher, K. Eichler, M. Zlatic, and A. Cardona, "Semiparametric spectral modeling of the drosophila connectome," *arXiv:1705.03297 [stat.ML]*, 2017.

[15] C. Truong, L. Oudre, and N. Vayatis, "Selective review of offline change point detection methods," *Signal Process.*, vol. 167, pp. 107299, 2020.

[16] G. Capdehourat, F. Larroca, and G. Morales, "A nation-wide Wi-Fi RSSI dataset: Statistical analysis and resulting insights," in *2020 IFIP Networking Conference (Networking)*, 2020, pp. 370–378.

[17] K. Levin, A. Athreya, M. Tang, V. Lyzinski, and C. E. Priebe, "A central limit theorem for an omnibus embedding of multiple random dot product graphs," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2017, pp. 964–967.

[18] A. A. Alyakin, J. Agterberg, H. S. Helm, and C. E. Priebe, "Correcting a nonparametric two-sample graph hypothesis test for graphs with different numbers of vertices," *arXiv:2008.09434 [stat.ME]*, 2020.