Algorithmic Advances for the Adjacency Spectral Embedding

Marcelo Fiori*, Bernardo Marenco*, Federico Larroca*, Paola Bermolen*, and Gonzalo Mateos†
*Universidad de la República, Uruguay. Email: {mfiori,bmarenco,flarroca,paola}@fing.edu.uy
†University of Rochester, Rochester, NY, USA. Email: gmateosb@ur.rochester.edu

***Abstract*—The Random Dot Product Graph (RDPG) is a popular generative graph model for relational data. RDPGs postulate there exist latent positions for each node, and specifies the edge formation probabilities via the inner product of the corresponding latent vectors. The embedding task of estimating these latent positions from observed graphs is usually posed as a non-convex matrix factorization problem. The workhorse Adjacency Spectral Embedding offers an approximate solution obtained via the eigendecomposition of the adjacency matrix, which enjoys solid statistical guarantees but can be computationally intensive and is formally solving a surrogate problem. In this paper, we bring to bear recent non-convex optimization advances and demonstrate their impact to RDPG inference. We develop first-order gradient descent methods to better solve the original optimization problem, and to accommodate broader network embedding applications in an organic way. The effectiveness of the resulting graph representation learning framework is demonstrated on both synthetic and real data. We show the algorithms are scalable, robust to missing network data, and can track the latent positions over time when the graphs are acquired in a streaming fashion.**

***Index Terms*—Graph Representation Learning; Gradient Descent; Non-convex Optimization; Random Dot Product Graphs**

I. Introduction

One of the most popular generative models for random graphs is the Random Dot Product Graph (RDPG). Under this model each node $i \in \{1, \dots, N\}$ in a simple, undirected graph G has an associated latent position vector $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d$, and edge (i, j) exists with probability $P_{ij} = \mathbf{x}_i^\top \mathbf{x}_j$, independent of all other edges. In other words, letting $\mathbf{A} \in \{0, 1\}^{N \times N}$ be the random symmetric adjacency matrix of G and $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]^\top \in \mathbb{R}^{N \times d}$ the matrix of latent vertex positions, the RDPG model specifies that *given* $\mathbf{X}$, edges are conditionally independent with $A_{ij} \sim \text{Bernoulli}(\mathbf{x}_i^\top \mathbf{x}_j)$. The model's popularity stems from its simplicity and expressiveness; e.g. the Erdös-Rényi and Stochastic Block Model (SBM) families are included as particular cases [1]. Furthermore, the resulting embeddings are easy to interpret: nodes with large $\|\mathbf{x}_i\|_2$ tend to exhibit higher connectivity, whereas a small angle between $\mathbf{x}_i$ and $\mathbf{x}_j$ indicates higher "affinity" among i and j.

Background on RDPG inference. Let us now discuss the associated inference (or node embedding) problem, which is the focus of the methods presented here. Given a realization of a graph (or a sequence of graphs), we look for the latent position matrix $\hat{\mathbf{X}}$ which best explains the given adjacency matrix $\mathbf{A}$ under the RDPG model. Since the maximum-likelihood estimator is intractable beyond toy graphs [2], moving forward we note that $\mathbf{A}$ is a noisy observation of $\mathbf{P} = \mathbf{X}\mathbf{X}^\top$, the rank-$d$ matrix of edge probabilities P_{ij}, since $\mathbb{E}\left[\mathbf{A} \,\middle|\, \mathbf{X}\right] = \mathbf{P}$. Therefore, and remembering that the diagonal entries of $\mathbf{P}$ are zero, we want to solve the following problem [2]:

$$\hat{\mathbf{X}} \in \underset{\mathbf{X} \in \mathbb{R}^{N \times d}}{\operatorname{argmin}} \|\mathbf{M} \circ (\mathbf{A} - \mathbf{X}\mathbf{X}^\top)\|_F^2, \tag{1}$$

where $\circ$ is the Hadamard product, and $\mathbf{M} = \mathbf{1}\mathbf{1}^\top - \mathbf{I}$ is a mask matrix, with zero-diagonal and ones everywhere else.

In the RDPG framework, the usual approach to obtain an approximate solution of (1) is to slightly modify the problem in order to avoid the zero-diagonal constraint (either by replacing the main diagonal of $\mathbf{A}$, or simply ignoring the constraint) [1]:

$$\hat{\mathbf{X}} \in \underset{\mathbf{X}}{\operatorname{argmin}} \|\mathbf{A} - \mathbf{X}\mathbf{X}^\top\|_F^2, \text{ s. to } \operatorname{rank}(\mathbf{X}) = d. \tag{2}$$

Its solution can be computed as $\hat{\mathbf{X}} = \hat{\mathbf{V}}\hat{\mathbf{\Lambda}}^{1/2}$, where $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$ is the eigendecomposition of $\mathbf{A}$, $\hat{\mathbf{\Lambda}} \in \mathbb{R}^{d \times d}$ is a diagonal matrix with the d largest eigenvalues of $\mathbf{A}$, and $\hat{\mathbf{V}} \in \mathbb{R}^{N \times d}$ are the corresponding d eigenvectors. This estimator is known as the Adjacency Spectral Embedding (ASE).

Contributions and paper outline. Inspired by related matrix-factorization problems, we propose to tackle the non-convex problem (1) via gradient descent (GD). As we show in Section II-A, our method scales better than the spectral-based ASE, and therefore may be used for graphs with several tens of thousands of vertices. Very recent papers [3] explicitly comment on the difficulty of scaling these RDPG approaches for large graphs and streaming settings. In this context, ours is the first work to develop scalable algorithms to compute RDPG embeddings, by proposing a proper formulation and bringing to bear recent advances in first-order non-convex optimization.

Furthermore, our framework allows to solve exactly the more appropriate problem formulation (1) [instead of (2)]. This limitation was recognized more than a decade ago [2], yet to the best of our knowledge it has not been satisfactorily addressed in the recent RDPG literature. The existing alternative [2], where the ASE is repeatedly computed and the diagonal entries of $\mathbf{A}$ are completed with the diagonal of $\hat{\mathbf{X}}\hat{\mathbf{X}}^\top$, lacks convergence guarantees and multiplies the ASE complexity by the number of iterations. Moreover, as we discuss in Section III, the proposed algorithmic framework

This work was partially funded by the NSF (awards CCF-1750428 and ECCS-1809356).

can seamlessly accommodate missing or unobserved data. All in all, our approach offers a *better* representation at a *lower* computational cost, in *more general* settings.

The last contribution is an online, lightweight method for tracking and visualizing ASEs of dynamic networks. Observe that the RDPG model (and the solutions of (2) and (1)), are invariant to orthogonal transformations. This poses a challenge when the goal is to embed a sequence of graphs and subsequently compare the estimated vectors. Existing alternatives to align the resulting estimates rely on eigendecomposition of a matrix whose size increases linearly with the number of graphs in the sequence [3]–[5]. In Section II-B we propose to use our first-order method to embed sequences of graphs, maintaining a certain alignment between the latent positions by virtue of warm restarts, while scaling favorably with the number of graphs involved. Tracking the latent nodal positions of dynamic graph streams is presented in Section IV.

II. Estimation via gradient descent

We propose to solve the smooth problem (1), and embedding tasks for other RDPG generalizations, using a GD approach. Although the formulation is not convex with respect to $\mathbf{X}$, there exist recent results showing convergence under very reasonable assumptions. Note that the objective function is indeed convex with respect to $\mathbf{Z} = \mathbf{X}\mathbf{X}^\top$, since it becomes $\|\mathbf{M} \circ (\mathbf{A} - \mathbf{Z})\|_F^2$.

Let us denote by $f : \mathbb{R}^{n\times d} \to \mathbb{R}$ the objective function $f(\mathbf{X}) = \|\mathbf{M} \circ (\mathbf{A} - \mathbf{X}\mathbf{X}^\top)\|_F^2$. The GD algorithm is

$$\mathbf{X}_{t+1} = \mathbf{X}_t - \alpha \nabla f(\mathbf{X}_t), \quad t = 0, 1, 2, \dots \tag{3}$$

where $\alpha > 0$ is the step size and $\nabla f(\mathbf{X}) = 4\left[\mathbf{M} \circ (\mathbf{X}\mathbf{X}^\top - \mathbf{A})\right]\mathbf{X}$, for symmetric $\mathbf{A}$ and $\mathbf{M}$. When applied to this class of problems where the objective function depends on the product $\mathbf{X}\mathbf{X}^\top$, and in general is convex with respect to $\mathbf{Z} = \mathbf{X}\mathbf{X}^\top$, this approach is sometimes called *factorized GD* [6], or *Procrustes flow* [7]. There have been several noteworthy advances in the study of its convergence, rate of convergence, and accelerated variants [6]–[10].

For the RDPG embedding problem dealt with here, the main result states that if the initial condition is close to the solution, the iteration (3) converges with linear rate to $\hat{\mathbf{X}}$ [6], [11].

Proposition 1: Let $\hat{\mathbf{X}}$ be a solution of (1). Then there exist $\delta > 0$ and $0 < \kappa < 1$ such that, if $\|\mathbf{X}_0 - \hat{\mathbf{X}}\|_F \leq \delta$, we have

$$d(\mathbf{X}_t, \hat{\mathbf{X}}) \leq \kappa^t d(\mathbf{X}_0, \hat{\mathbf{X}}), \quad \forall t > 0 \tag{4}$$

where $\mathbf{X}_t$ is the sequence of GD iterates (3) with an appropriate constant stepsize, and $d(\mathbf{X}, \hat{\mathbf{X}}) := \min_{\mathbf{W}\in\mathcal{O}^{d\times d}} \|\mathbf{X}\mathbf{W} - \hat{\mathbf{X}}\|_F^2$ accounts for the orthogonality invariance.

See [11] and references therein for a similar version of this proposition. Although there are some very specific initializations which correspond to stationary points (and therefore do not lead to global convergence), in our experience the method converges to the global optimum when initalizated randomly.

Generalizations. The RDPG model can be extended to describe more general graph families. For instance, the Generalized RDPG model [12] can capture disassortative connectivity patterns by including a diagonal matrix $\mathbf{D}_{p,q}$, with p elements with value $+1$ and q elements with value -1. The formulation (1) can be then readily adapted for this model, resulting in

$$\hat{\mathbf{X}} \in \operatorname*{argmin}_{\mathbf{X}\in\mathbb{R}^{n\times d}} \|\mathbf{M} \circ (\mathbf{A} - \mathbf{X}\mathbf{D}_{p,q}\mathbf{X}^\top)\|_F^2, \tag{5}$$

which also allows to compute embeddings with missing data. Indeed, via the binary mask $\mathbf{M}$ one can encode which entries of $\mathbf{A}$ are observed and which are missing. The gradient step in (3) only differs in the computation of the gradient itself.

The RDPG model can only represent undirected graphs, since the product $\mathbf{X}\mathbf{X}^\top$ is always symmetric. Extensions to accommodate directed graphs (digraphs) are possible [1], [13], whereby each node i of the digraph has two vectors associated, denoted by $\mathbf{x}_i^l$ and $\mathbf{x}_i^r$. This way, the Directed RDPG (D-RDPG) model is defined as

$$\mathrm{P}\left(\mathbf{A} \,\middle|\, \mathbf{X}\right) = \prod_{i\neq j} [(\mathbf{x}_i^l)^\top \mathbf{x}_j^r]^{A_{ij}} [1 - (\mathbf{x}_i^l)^\top \mathbf{x}_j^r]^{1-A_{ij}}. \tag{6}$$

For this case, the corresponding embedding problem (which also allows to work with missing data) becomes

$$\{\hat{\mathbf{X}}^l, \hat{\mathbf{X}}^r\} = \operatorname*{argmin}_{\mathbf{X}^l\in\mathbb{R}^{n\times d}, \mathbf{X}^r\in\mathbb{R}^{n\times d}} \|\mathbf{M} \circ (\mathbf{A} - \mathbf{X}^l(\mathbf{X}^r)^\top)\|_F^2.$$

Here, GD can be thought as comprising two gradient steps, one with respect to $\mathbf{X}^l$ and the other with respect to $\mathbf{X}^r$.

A. Complexity and execution time analyses

The per-iteration computational cost incurred to evaluate $\nabla f(\mathbf{X})$ is dominated by the matrix multiplication, which is $\Theta(N^2 d)$ for a naïve implementation. The number of iterations depends on $\mathbf{X}_0$, but in our experience even with random initializations the runtime is still markedly lower than the SVD-based ASE. Further time complexity improvements may be obtained by using Nesterov-type acceleration methods [14]. In Figure 1 we compare the execution times of GD and the ASE as a function of N. For the latter, we use the `SciPY` implementation of the eigendecomposition in Python, as in state-of-the-art RDPG inference packages such as `Graspologic` [15]. For GD, we test two different initializations: a uniform random matrix $\mathbf{X}$, and one based on a fast eigendecomposition computation using the randomized-SVD (RSVD) [16] (we account for the RSVD in the overall execution time). In all

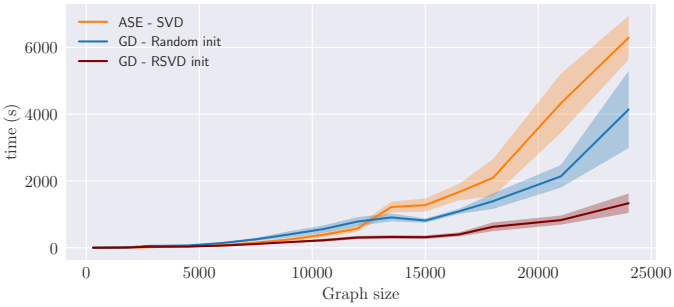

Fig. 1. Execution time for SBM graphs up to $N = 24000$ nodes. As N grows, GD exhibits markedly better scaling than the state-of-the-art ASE algorithm.

cases, the methods converge to a solution of the inference problem. The obtained cost function is very similar for each run, with slightly lower values for the GD method because it is solving the problem with the zero-diagonal restriction.

For each N, we sampled several 2-block SBM graphs, with connection probabilities of $p = 0.5$ (within community) and $q = 0.2$ (across communities). Community sizes are $N/3$ and $2N/3$. The embedding dimension was set to $d = 2$ in all cases. Results are averaged over 10 Monte Carlo replicates, and corresponding standard deviations are depicted in Figure 1. As N grows, GD attains significant reductions in wall-clock time relative to the state-of-the-art ASE implementation, especially when GD is initialized using the RSVD.

B. Warm restart for embedding graph sequences

As mentioned in Section I, the embedding matrix $\hat{\mathbf{X}}$ in the RDPG model can only be determined up to rotations. This challenges network inference tasks where the data comes from multiple graphs, e.g., sequential graphs over time. For instance, problems like hypothesis testing to determine whether two graphs are drawn from the same RDPG model, or tracking nodal embeddings over time, heavily rely on the correspondence of nodes across different networks.

For the hypothesis testing problem, a test is put forth in [17] which involves solving a Procrustes problem to align the embeddings. A joint so-termed *Omnibus* embedding of m graphs is proposed in [4], by forming an $mN \times mN$ matrix from the adjacency matrices and computing its ASE latent positions. More recently, the Unfolded Adjacency Spectral Embedding (UASE) was proposed in [3], [5]. The UASE also relies on an auxiliary matrix, this time by horizontally stacking the adjacency matrices of all graphs, and then computes its SVD to obtain the joint embeddings. Both these approaches come with asymptotic statistical guarantees under some technical assumptions. However, from a computational standpoint they do not scale well with the number of nodes N of each graph, or the total number of graphs m.

The GD algorithm of this paper can be initialized using the latent positions of another related graph. This so-termed warm restart will not only decrease processing time, but is also likely to yield embeddings that are closely aligned with those of previous graphs in the squence.

Numerical example. To illustrate this desirable behaviour, we borrow the motivating experiment and code from [3]. The goal is to compute the embedding of two SBMs with four communities, that change their connection probabilities. Two graphs are respectively generated according to the following SBMs, whose inter-community probabilities are given by

$$\mathbf{B}_1 = \begin{pmatrix} 0.08 & 0.02 & 0.18 & 0.10 \\ 0.02 & 0.20 & 0.04 & 0.10 \\ 0.18 & 0.04 & 0.02 & 0.02 \\ 0.10 & 0.10 & 0.02 & 0.06 \end{pmatrix}, \mathbf{B}_2 = \begin{pmatrix} 0.16 & 0.16 & 0.04 & 0.10 \\ 0.16 & 0.16 & 0.04 & 0.10 \\ 0.04 & 0.04 & 0.09 & 0.02 \\ 0.10 & 0.10 & 0.02 & 0.06 \end{pmatrix}.$$

Notice how communities 1 and 2 merge in the second model, while community 4 retains the same probabilities with the other three groups. In [3], the authors comment how their UASE approach manages to capture the merger of communities $1-2$, while keeping the latent positions of nodes in community 4 largely unchanged. The *Omnibus* approach, or independent embedding of both graphs, fails in accomplishing at least one of these objectives; see Figure 2.

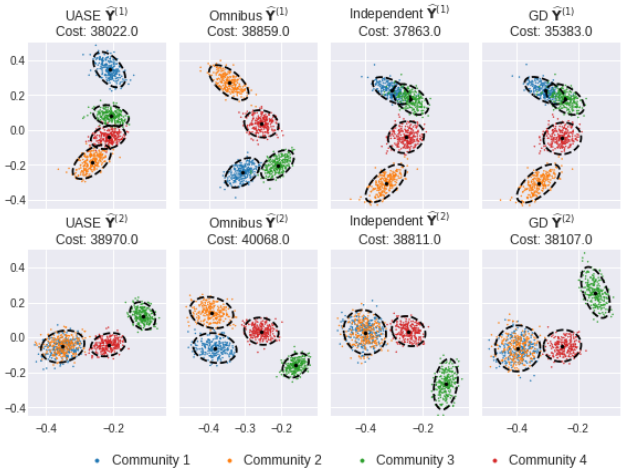

Fig. 2. Embeddings of two SBM graph realizations, where communities 1 and 2 merge, while community 4 keeps the connection probabilities with other groups. Observe how the GD approach (far right) manages to capture this behaviour, while providing the best representation for each graph individually (quantified by the smallest cost function values). Code adapted from [3].

We tested the GD algorithm in this numerical example, and confirmed that the obtained embeddings also reflect the desired behaviour regarding communities 1, 2, and 4. Moreover, GD provides a better overall representation for each graph (similar to independent embeddings of the graph). We quantify this via the cost function of (1) evaluated at each solution. These values are also presented in Fig. 2, and show the favorable representation quality obtained by the GD algorithm.

Another example of this alignment method via warm restarts is outlined in Section IV, using a real-world application involving model tracking for Wi-Fi network monitoring.

III. Inference with missing data

We now show an example of how GD-based inference can be useful for problems with missing data. In this setup we have a bipartite graph that simulates a two-party senate. Nodes correspond to senators and laws, and the fact that senator i has voted affirmatively for law j is indicated by the edge (i, j).

Each of the two parties can submit a law for voting, with affirmative votes being more likely for senators from the party that introduced the law, and less likely for senators from the opposing party. There are also bipartisan laws, for which affirmative votes are more similar across parties. Furthermore, for each law, there is a subset of senators that have a 30% chance of being absent from the voting. So in our formulation the mask matrix $\mathbf{M}$ in (1) will encode whether those senators were present in the voting of each law.

We simulated such a graph with 50 senators of each party and a total of 230 proposed laws, and compared the embeddings given by the ASE and by GD. Results are shown in Fig. 3. Both methods result in a clear alignment between

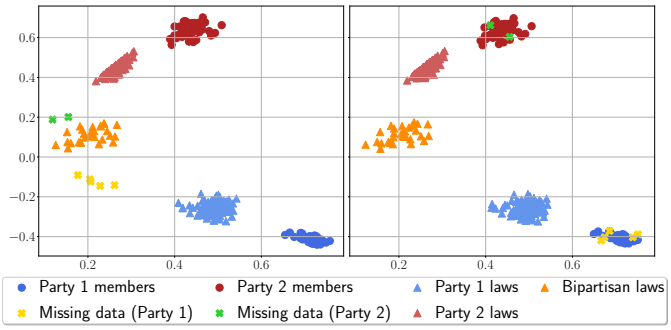

Fig. 3. Embeddings with missing data. Naive ASE (left) and GD with mask matrix encoding present and absent voters (right). Our approach is able to assign the absent voters to the correct group.

the embeddings of laws and senators from the same party, and opposing groups are embedded as almost orthogonal clusters (which is consistent with a small inner product, i.e. small connection probability). The difference is that GD assigns the senators that are absent to the same cluster as the rest of the senators in their party, while the ASE assigns them to a smaller, independent cluster. GD is able to correctly handle the missing information in this setup, producing a representation that is faithful to the underlying structure of the data.

IV. Model tracking for dynamic graphs

Consider now a monitoring scenario, where we observe a stream of graphs G_t (where t denotes time), and the goal is to track the underlying model. We will assume that there exists a correspondence between nodes at different times, and that all these graphs stem from an RDPG with latent positions given by $\mathbf{X}_t \in \mathbb{R}^{N\times d}$. We focus on the vanilla model for ease of exposition, and consider other variants in the numerical tests.

There is a growing interest in the above setting, with several applications arising with the online change-point detection problem; i.e., flagging if and when the underlying generative model changed [13], [18]–[20]. Here we take a step further, and strive at actually tracking the model. Applications may include recommender systems (where rankings are revealed or even change over time) [21] or, as we discuss below, monitoring wireless networks [22].

Suppose then that at time t we observe a window of length m of the past graphs in the sequence, which we may safely assume stem from the same RDPG model; i.e., $\mathbf{X}_{t-m} = \ldots = \mathbf{X}_t = \mathbf{X}$. In this case the best estimate of $\mathbf{X}$ is the ASE of the averaged adjacency matrix $\bar{\mathbf{A}}_t = 1/m \sum_{k=t-m}^{t} \mathbf{A}_k$ [23]. Applying GD iterations (3) when $\mathbf{A} \leftarrow \bar{\mathbf{A}}_t$ yields updates:

$$\hat{\mathbf{X}}_t = \hat{\mathbf{X}}_{t-1} - \alpha \nabla_{\hat{\mathbf{X}}} \|\mathbf{M} \circ (\bar{\mathbf{A}}_t - \hat{\mathbf{X}}_{t-1}\hat{\mathbf{X}}_{t-1}^\top)\|_F^2. \quad (7)$$

The algorithm (7) suggests a tracking system as the one depicted in Figure 4, where we have substituted $\bar{\mathbf{A}}_t$ with the output of an entry-wise filter $\mathbf{F}(z)$ applied to the stream of incoming adjacency matrices $\mathbf{A}_t$. A moving average as the one discussed before is readily recovered by considering a FIR filter with all its m taps set to $1/m$, although a single-pole IIR filter would be preferable in terms of memory and

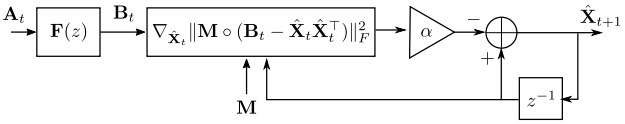

Fig. 4. A diagram of the proposed RDPG tracking system. If the entry-wise filter $\mathbf{F}(z)$ is a moving average, we would obtain $\mathbf{B}_t = \bar{\mathbf{A}}_t$, resulting in embedding the average adjacency matrix.

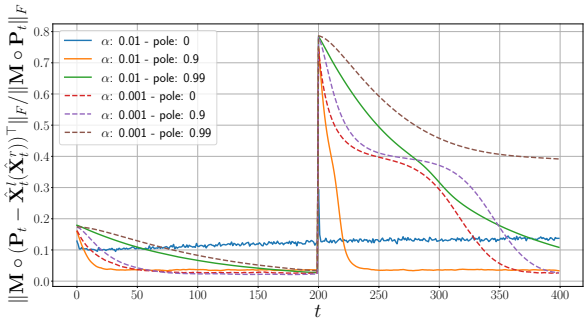

Fig. 5. Evolution of $\|\mathbf{M} \circ (\mathbf{P}_t - \hat{\mathbf{X}}_t^l(\hat{\mathbf{X}}_t^r)^\top\|_F)/\|\mathbf{M} \circ \mathbf{P}_t\|_F$ as a function of t. Different curves correspond to different step sizes and poles. A judicious choice of these values yields both estimation accuracy and tracking speed.

processing. We may even drop the filtering stage altogether (setting $m = 1$) to yield a least mean squares (LMS)-type stochastic gradient algorithm .

Simulated numerical results. Let us consider a synthetic example, where a stream G_t of SBMs digraphs with two communities of $N/2 = 100$ nodes each changes from an assortative to a disassortative behaviour. That is to say, until $t = 200$ nodes belonging to the same community would connect with probability $p = 0.5$ and with members of the other community with probability $q = 0.2$. This is reversed so that $p = 0.2$ and $q = 0.5$ after $t = 200$, meaning that nodes now tend to connect more to nodes of the other community.

The temporal evolution of the relative error between the true $\mathbf{P}_t$ and the estimated $\hat{\mathbf{X}}_t^l(\hat{\mathbf{X}}_t^r)^\top$ obtained via our tracking system is shown in Fig. 5. Different curves correspond to different values of α and the pole's value of $\mathbf{F}(z)$. In all cases, the ASE of the first graph is used as the initial estimate of $\hat{\mathbf{X}}_0^l$ and $\hat{\mathbf{X}}_0^r$. Firstly, note how when no filter is applied (i.e. a pole at 0) excellent estimates of the latent positions are found, provided that a small enough step size is used. Else, the resulting estimate is no better than the single graph one. Note however that what is lost in precision is gained in speed, as after $t = 200$ the error is the same as before in very few iterations for $\alpha = 0.01$, whereas the more precise $\alpha = 0.001$ takes almost 150 iterations to find the new embeddings.

Secondly, note how including the filter and a judicious choice of α may result in better accuracy-speed tradeoffs. When the incoming graphs' generative model do not change, the averaging performed by the filter allows to use larger values of α without compromising precision. This in turn may result in faster convergence after the change (see the curve corresponding to $\alpha = 0.01$ and a pole at 0.9).

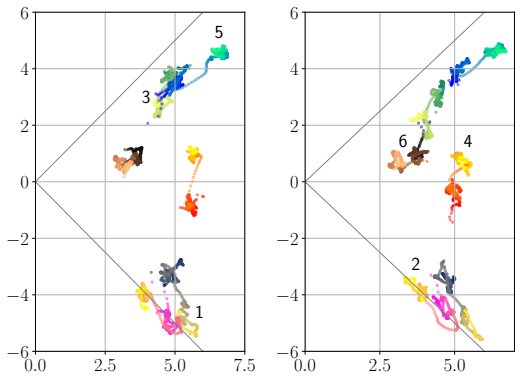

Fig. 6. $\hat{\mathbf{X}}_t^l$ (left) and $\hat{\mathbf{X}}_t^r$ (right) using $d = 2$ for the RSSI example. Color palettes identify the nodes and a lighter tone indicate larger values of t. Best viewed in a color display. The network's change at $t \approx 310$ is clearly visible: the AP that was moved ($i = 4$) is now closer to the upper cluster of nodes.

Real-world dataset. We also study the dataset described in [24], which includes the Received Signal Strength Indicator (RSSI) measurements between Wi-Fi access points (APs) in a Uruguayan school. In particular we considered a network consisting of $N = 6$ APs, with measurements collected hourly during almost four weeks, corresponding to $m = 655$ graphs. The AP corresponding to $i = 4$ was moved at $t \approx 310$. As RSSI is measured in dBm (and are negative), we have first added an offset to all weights so that they become positive and larger values still mean "stronger" edges. We thus have a directed (as power measurements between APs are not necessarily symmetric) and weighted graph sequence.

The GD estimates $\hat{\mathbf{X}}_t^l$ and $\hat{\mathbf{X}}_t^r$ for $d = 2$ are shown in Fig. 6. We have used $\alpha = 0.01$ and $\mathbf{F}(z)$ with a pole at 0.9. Different color palettes identify the nodes and as t increases the color becomes lighter. Note how at first all the embeddings are relatively static (cf. darker colors), with two almost orthogonal clusters of nodes, in addition to node 4 in a somewhat intermediate position. When the network is modified, this node approaches the upper cluster. The node is apparently moved nearer node 5 than 3, and not so far away from node 1. Note how the movement of nodes in the clusters are radial and result in an unchanged inner product between them.

V. Concluding Summary

In this paper we proposed a new algorithmic framework to estimate latent positions of RDPG models, by bringing to bear non-convex optimization techniques that have been recently developed for low-rank matrix factorization problems. This allows to solve the associated nodal embedding problem by taking into account the (often overlooked) zero-diagonal constraint, which is relevant for graphs with no self loops. The general formulation also offers the possibility to compute the RDPG embeddings in the presence of unobserved data. We tested the computational complexity of the method, and provided examples of applications with missing data, and to track latent positions of nodes in graph sequences, both with synthetic and real data.

References

[1] 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, January 2017.

[2] E.R. Scheinerman and K. Tucker, "Modeling graphs using dot product representations," *Comput. Stat*, vol. 25, pp. 1–16, 2010.

[3] I. Gallagher, A. Jones, and P. Rubin-Delanchy, "Spectral embedding for dynamic networks with stability guarantees," *Advances in Neural Information Processing Systems 34, NeurIPS*, 2021.

[4] 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 *Int. Conf. on Data Mining Workshops*, 2017, pp. 964–967.

[5] A. Jones and P. Rubin-Delanchy, "The multilayer random dot product graph," *arXiv preprint arXiv:2007.10455*, 2020.

[6] S. Bhojanapalli, A. Kyrillidis, and S. Sanghavi, "Dropping convexity for faster semi-definite optimization," in *Conference on Learning Theory*. PMLR, 2016, pp. 530–582.

[7] S. Tu, R. Boczar, M. Simchowitz, M. Soltanolkotabi, and B. Recht, "Low-rank solutions of linear matrix equations via procrustes flow," in *International Conference on Machine Learning*. PMLR, 2016.

[8] Y. Chen and M. Wainwright, "Fast low-rank estimation by projected gradient descent: General statistical and algorithmic guarantees," 2015.

[9] R. Sun and Z. Luo, "Guaranteed matrix completion via non-convex factorization," *IEEE Trans. on Information Theory*, vol. 62, 2016.

[10] D. Zhou, Y. Cao, and Q. Gu, "Accelerated factored gradient descent for low-rank matrix factorization," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 4430–4440.

[11] Y. Chi, Y. M Lu, and Y. Chen, "Nonconvex optimization meets low-rank matrix factorization: An overview," *IEEE Trans. on Signal Processing*, vol. 67, no. 20, pp. 5239–5269, 2019.

[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] Bernardo Marenco, Paola Bermolen, Marcelo Fiori, Federico Larroca, and Gonzalo Mateos, "Online change point detection for weighted and directed random dot product graphs," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 8, pp. 144–159, 2022.

[14] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $o(1/k^2)$," in *Sov. Math. Dokl*, 1983, vol. 27.

[15] J. Chung, B. D. Pedigo, E. W. Bridgeford, B. K. Varjavand, H. S. Helm, and J. T. Vogelstein, "Graspy: Graph statistics in Python.," *J. Mach. Learn. Res.*, vol. 20, no. 158, pp. 1–7, 2019.

[16] N. Halko, P. Martinsson, and J. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM review*, vol. 53, no. 2, pp. 217–288, 2011.

[17] M. Tang, A. Athreya, D. Sussman, V. Lyzinski, Y. Park, and C.E. Priebe, "A semiparametric two-sample hypothesis testing problem for random graphs," *J. of Comput. and Graphical Statistics*, vol. 26, no. 2, 2017.

[18] Y. Yu, O. H. M. Padilla, D. Wang, and A. Rinaldo, "Optimal network online change point localisation," *arXiv:2101.05477 [math.ST]*, 2021.

[19] H. Chen, "Sequential change-point detection based on nearest neighbors," *Ann. Stat*, vol. 47, no. 3, pp. 1381 – 1407, 2019.

[20] M. Zhang, L. Xie, and Y. Xie, "Online community detection by spectral cusum," in *Proc. Int. Conf. Acoustics, Speech, Signal Process.*, 2020.

[21] P. Campos, F. Díez, and I. Cantador, "Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols," *User Modeling and User-Adapted Interaction*, vol. 24, no. 1, pp. 67–119, 2014.

[22] 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.

[23] 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.

[24] 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.