Approximate High Dimensional Graph Mining With Matrix Polar Factorization: A Twitter Application

Georgios Drakopoulos
Ionian University
0000-0002-0975-1877

Eleanna Kafeza
Zayed University
0000-0001-9565-2375

Phivos Mylonas
Ionian University
0000-0002-6916-3129

Spyros Sioutas
University of Patras
sioutas@ceid.upatras.gr

Abstract—**At the dawn of the Internet era graph analytics play an important role in high- and low-level network policymaking across a wide array of fields so diverse as transportation network design, supply chain engineering and logistics, social media analysis, and computer communication networks, to name just a few. This can be attributed not only to the size of the original graph but also to the nature of the problem parameters. For instance, algorithmic solutions depend heavily on the approximation criterion selection. Moreover, iterative or heuristic solutions are often sought as it is a high dimensional problem given the high number of vertices and edges involved as well as their complex interaction. Replacing under constraints a directed graph with an undirected one having the same vertex set is often sought in applications such as data visualization, community structure discovery, and connection-based vertex centrality metrics. Polar decomposition is a key matrix factorization which represents a matrix as a product of a symmetric positive (semi)definite factor and an orthogonal one. The former can be an undirected approximation of the original adjacency matrix. The proposed graph approximation has been tested with three Twitter graphs with encouraging results with respect to density, Fiedler number, and certain vertex centrality metrics based on matrix power series. The dataset was hosted in an online MongoDB instance.**

Index Terms—**polar factorization, graph approximation, algebraic graph mining, graph signal processing, MongoDB**

1. Introduction

With the advent of the Internet era physical and virtual networks have been proliferated while their importance in almost every financial and technological sector has been multiplied. Approximating large graphs is among the major algorithmic cornerstones of a plethora of graph analytics. Although directed graphs constitute a more general algorithmic or data flow model and certain problems in fields such as electrical current computation, maximum profit routes in currency exchange networks, computational neuroscience, or capacity and flow prediction can be accurately modeled only with them, in many engineering applications approximating the former with the latter is often necessary if meaningful results are to obtained under tight time constraints. Prominent examples include event discovery in social media [1], high abstraction views [2], and graph resilience [3].

The above raise two questions, namely what is the approximation criterion and how to actually compute an undirected graph which is optimal under said criterion. The answer to both questions is further complicated from the inherent high graph dimensionality, which ultimately translates to the large cardinality of the vertex set as well as to the complexity of their interaction as cofidied in the edge set or equivalently in the graph adjacency matrix. Empirical evidence tends to corroborate that ignoring edge direction results in a rather structurally distorted graph. The main motivation behind this work is to address both questions as well as the dimensionality issue through the matrix polar factorization. An iterative scheme [4] can be the building block for new approaches, whereas approximation goodness can be assessed from graph properties.

The primary contribution of this conference paper is a higher dimensionality methodology for approximating directed unweighted graphs with undirected ones while preserving up to a degree key structural properties such as density and community structure. The algorithmic cornerstone for the this is the polar decomposition, which differentiates this work from previous ones. The proposed scheme was applied to three Twitter graphs obtained with topic sampling.

The structure of this conference paper follows. Section 2 overviews relevant scientific literature regarding graph approximation, graph signal processing, and polar factorization. In section 3 the main properties of the matrix polar factorization are examined, whereas in section 4 its application to directed graph approximation is described. The results are summarized and interpreted in section 5 and this work is concluded with section 6. Vectors are symbolized with lowcase boldface and matrices with capital boldface. Acronyms are explained the first time they are encountered in the text. Finally, notation is summarized in table 1.

2. Previous Work

Graph signal processing has recently emerged as an interdisciplinary field with numerous applications [5] [6]. Therein are defined operations on irregular domains [7] including graph sampling [8], graph Laplace transforms [9], vertex frequency operations [10], and dimensionality

978-1-xxxx-xxxx-x/21/$31.00 ©2021 IEEE

TABLE 1. NOTATION OF THIS CONFERENCE PAPER

Symbol	Meaning	First in
$\triangleq$	Definition or equality by definition	Eq. (2)
$\mathbf{M}^{[k]}$	k-th iteration of an iterative process	Eq. (20)
$\mathrm{radius}\left(\mathbf{M}\right)$	Spectral radius of matrix $\mathbf{M}$	Sec. 3
$\mathrm{sp}\left(\mathbf{M}\right)$	Spectrum of matrix $\mathbf{M}$	Eq. (6)
$\mathrm{diag}\left[d_1 \dots d_n\right]$	Diagonal matrix with $d_1 \dots d_n$	Eq. (30)
$\lVert\mathbf{M}\rVert$	Matrix or vector norm	Eq. (5)
$\{s_1, \dots, s_n\}$	Set with elements $s_1, \dots, s_n$	Eq. (6)
$\lvert S\rvert$	Set or sequence cardinality functional	Eq. (6)
$\deg\left(u\right)$	Degree of vertex u	Eq. (30)
$\langle p_1 \,\Vert\, p_2\rangle$	Kullback-Leibler divergence	Eq. (35)

reduction with discrete cosine transform [11]. Graph neural networks (GNNs) are based on message passing protocols which contribute to the evolution of local information stored in vertices [12] and thus rely heavily on topological properties [13] [14]. Techniques in the face of topological uncertainties are proposed in [15].

Matrix polar factorization [4] has been proposed as a method on its own right [16] and as an alternative to singular value decomposition (SVD) [17], which is been frequntly used in methods assessing vertex importance as a hub or as an authority as for instance in the tensor harmonic centrality [18], the Gel-point centrality [19], and the HITS algorithm in information retrieval (IR) [20]. The polar decomposition is also related to the Cholesky factorization [21]. Incomplete forms of the latter are the computational kenrels for an entire class of machine learning (ML) methods [22] [23]. Also it is the building block for symmetric matrix interpolation [24]. A distributed implementation is described in [25].

Approximation is a major challenge in graph mining and various definitions can be found in the literature depending on the task at hand such as flow reliability [26] and structural resilience [27]. Proposed frameworks for replacing the entire combinatorial structure range from Fourier transform based methods [28], Markov chain approximation based on graph spectra [29], low cost subgraphs of bidirected Steiner networks [30], and abstractions based on neighborhood systems [2]. Another class of methodologies focuses on estimating graph properties such as diameter and eccentricity [31] and vertex set cardinality [32]. A space efficient data structure supporting persistency in Scala is discussed in [33].

3. Polar Factorization

In this section the definition and the properties of the polar factorization are described and intuition about them is given. In the scalar case the polar factorization is tantamount to casting a complex number z to the form of equation (1):

$$z \;=\; x + iy \;=\; re^{i\theta} \;=\; r(\cos\theta + i\sin\theta) \qquad (1)$$

In (1) the magnitude r and the angle θ are given from non-linear transforms as shown in equation (2). Since the inversion of both sine and cosine yields each two possible solutions, their intersection gives the correct angle.

$$r \triangleq \sqrt{x^2 + y^2}$$

$$\theta \triangleq \arcsin\left(\frac{y}{r}\right) \cap \arccos\left(\frac{x}{r}\right) \qquad (2)$$

Definition 1 shows the fundamental properties of the right polar factorization. There is also an alternative form known as the left polar factorization, but it is less common that the form used and explained in this work.

Definition 1 (Polar factorization). The polar factorization of $\mathbf{A} \in \mathbb{R}^{n\times n}$ is the product of a symmetric positive (semi)definite matrix $\mathbf{P}$ with an orthogonal matrix $\mathbf{Q}$:

$$\mathbf{A} \;=\; \mathbf{QP} \qquad (3)$$

It should be noted that $\mathbf{A}^{-1}$ exists if and only if $\mathbf{P}$ is positive definite. The orthogonality of $\mathbf{Q}$ implies (4):

$$\mathbf{Q}^T\mathbf{Q} \;=\; \mathbf{Q}\mathbf{Q}^T \;=\; \mathbf{I} \qquad (4)$$

Geometrically this means that $\mathbf{Q}$ is a product of rotations and reflections. As a result $\mathbf{Q}$ maintains the length of any vector is multiplied with and hence it is invariant under the Euclidean norm. Also, since $\mathbf{Q}$ is of full rank by definition any zero eigenvalues of $\mathbf{A}$ will be part of the spectrum of $\mathbf{P}$. Equation (4) also directly implies the fundamental identity (5) for any orthogonal matrix $\mathbf{U}$ and any vector $\mathbf{s}$:

$$\lVert\mathbf{Us}\rVert_2 \triangleq \sqrt{(\mathbf{s}^T\mathbf{U}^T)\mathbf{Us}} = \sqrt{\mathbf{s}^T\mathbf{U}^T\mathbf{Us}} = \sqrt{\mathbf{s}^T\mathbf{s}} \triangleq \lVert\mathbf{s}\rVert_2 \quad (5)$$

Equation (5) means that the length measured in the Euclidean norm of any vector $\mathbf{s}$ remains invariant after the multuplication with any orthogonal matrix $\mathbf{U}$ or by induction with any sequence of orthogonal matrices for that matter. This is paramout in the transformation of linear systems to equivalent ones which are easier solved as for instance in the case of linear least squares with QR [34].

In the remaining of this conference paper the spectra of $\mathbf{A}$ and $\mathbf{P}$ will be respectively denoted as $\mathrm{sp}\left(\mathbf{A}\right)$ and $\mathrm{sp}\left(\mathbf{P}\right)$ as shown in equation (6). Both sets consist of n elements including the algebraic multiplicity of each eigenvalue.

$$\begin{aligned} \mathrm{sp}\left(\mathbf{A}\right) &\triangleq \{\lambda_1, \dots, \lambda_n\}, \quad \lvert\mathrm{sp}\left(\mathbf{A}\right)\rvert \;=\; n \\ \mathrm{sp}\left(\mathbf{P}\right) &\triangleq \{\mu_1, \dots, \mu_n\}, \quad \lvert\mathrm{sp}\left(\mathbf{P}\right)\rvert \;=\; n \end{aligned} \qquad (6)$$

By definition $\mathrm{sp}\left(\mathbf{P}\right)$ contains real and nonnegative values. Therefore without loss of generality it holds that:

$$\mu_1 > \mu_2 \geq \ldots \geq \mu_n \geq 0 \qquad (7)$$

Since $\mathbf{A}$ is at least of rank one, as it is a non-zero matrix, so is $\mathbf{P}$ since $\mathbf{Q}$ is of full rank. Thus, it always holds that $\mu_1 > 0$ and $\lambda_1 \neq 0$. The corresponding eigenvectors of $\mathbf{A}$ and $\mathbf{P}$ are denoted by $\mathbf{g}_k$ and $\mathbf{u}_k$ where $1 \leq k \leq n$.

Some of the major properties of the polar factorization follow. They establish upper and lower approximation bounds as well as properties of spectrum $\mathrm{sp}\left(\mathbf{P}\right)$.

Property 1. For any orthogonal matrix $\mathbf{U} \in \mathbb{R}^{n\times n}$ and for any invariant under orthogonal transforms norm $\lVert\cdot\rVert$ such as the Euclidean $\lVert\cdot\rVert_2$ and the Frobenius $\lVert\cdot\rVert_F$ hold:

$$\begin{aligned} \lVert\mathbf{A} - \mathbf{Q}\rVert \;\leq\; \lVert\mathbf{A} - \mathbf{U}\rVert \;\leq\; \lVert\mathbf{A} + \mathbf{Q}\rVert \quad &\Leftrightarrow \\ \lVert\mathbf{P} - \mathbf{I}\rVert \;\leq\; \lVert\mathbf{A} - \mathbf{U}\rVert \;\leq\; \lVert\mathbf{P} + \mathbf{I}\rVert & \end{aligned} \qquad (8)$$

Property 1 means that under $\|\cdot\|$ the closest orthogonal matrix to $\mathbf{A}$ is $\mathbf{Q}$ and the furthest is $-\mathbf{Q}$. This establishes the arguments for the bounds for a broad class of matrix norms. In certain applications the actual bounds, namely the values of the norms, are also of interest. The latter depend heavily on the norm utilized. In the general case this is not always straightforward as the computation of certain norms may well be an NP-hard problem. However, in certain special cases analytical bounds do exist. In the case of the Frobenius norm $\|\cdot\|_F$ the lower and upper bounds can be found as follows. A formula for the lower bound is given in (9):

$$\begin{aligned}\|\mathbf{P}-\mathbf{I}\|_F^2 &\triangleq \operatorname{tr}\left((\mathbf{P}-\mathbf{I})^T(\mathbf{P}-\mathbf{I})\right)\\ &= \operatorname{tr}\left(\mathbf{P}^T\mathbf{P}\right) - 2\operatorname{tr}(\mathbf{P}) + \operatorname{tr}(\mathbf{I})\\ &= \|\mathbf{P}\|_F^2 - 2\operatorname{tr}(\mathbf{P}) + n\\ &= \|\mathbf{A}\|_F^2 - 2\sum_{k=1}^{n}\mu_k + n \qquad (9)\end{aligned}$$

Along a similar line of reasoning the upper bound of property 1 for the Frobenius norm is computed as in (10):

$$\begin{aligned}\|\mathbf{P}+\mathbf{I}\|_F^2 &= \|\mathbf{A}\|_F^2 + 2\sum_{k=1}^{n}\mu_k + n\\ &\leq \|\mathbf{A}\|_F^2 + n(1+2\mu_1) \qquad (10)\end{aligned}$$

The lower and upper bounds in the case of the Euclidean $\|\cdot\|_2$ can be computed as follows. Both $\mathbf{P}\pm\mathbf{I}$ are symmetric and hence with real spectrum. Thus, their respective maximum singular values σ_{max} equal $\operatorname{radius}(\mathbf{P}\pm\mathbf{I})$ and so:

$$\begin{aligned}\|\mathbf{P}+\mathbf{I}\|_2 &= \mu_1 + 1\\ \|\mathbf{P}-\mathbf{I}\|_2 &= \max_{1\leq k\leq n}|\mu_k - 1| \qquad (11)\end{aligned}$$

Recall that for any $\mathbf{M}\in\mathbb{R}^{m\times n}$ the Frobenius norm is defined as the square root of the sum of the squared matrix entries as shown in equation (12). One of its major advantages is that it can be immediately and naturally extended to tensors of literally any order, making it thus quite useful among others in advanced social network analysis [35] or in adaptive tensor-based non-linear system identification [36].

$$\|\mathbf{M}\|_F \triangleq \sqrt{\operatorname{tr}(\mathbf{M}^T\mathbf{M})} = \left(\sum_{i=1}^{m}\sum_{j=1}^{n}\mathbf{M}[i,j]^2\right)^{\frac{1}{2}} \qquad (12)$$

Observe that the Frobenius norm is invariant under linear orthogonal transforms as for any square matrices $\mathbf{M}$ and $\mathbf{U}$ where the latter is orthogonal equation (13) yields:

$$\|\mathbf{UM}\|_F \triangleq \sqrt{\operatorname{tr}(\mathbf{M}^T\mathbf{U}^T\mathbf{UM})} = \sqrt{\operatorname{tr}(\mathbf{M}^T\mathbf{M})} \triangleq \|\mathbf{M}\|_F \qquad (13)$$

Intuitively the Frobenius norm is a measure of the total energy of the elements of a matrix. However, in the context of other applications this norm may have additional interpretations as well. For instance from a graph mining perspective the Frobenius norm of an adjacency matrix is a measure of the graph density as it shows the vertex set cardinality. Frobenius and Euclidean (or ℓ_2) norms have the additional advantage of being differentiable, allowing therefore gradient-based optimization techniques. In contrast the Chebyshev (or ℓ_∞) and Manhattan (or ℓ_1) norms are not.

Another impoartant property relates $\mathbf{P}$ with $\mathbf{A}^T\mathbf{A}$.

Property 2. $\mathbf{P}$ is the unique Cholesky factor of $\mathbf{A}^T\mathbf{A}$.

Proof: Applying the right polar factorization to both matrices $\mathbf{A}$ and $\mathbf{A}^T$ the result follows immediately considering also he uniqueness of the Cholesky factorization [23].

$$\mathbf{A}^T\mathbf{A} = \mathbf{P}^T\mathbf{Q}^T\mathbf{Q}\mathbf{P} = \mathbf{P}^T\mathbf{P} = \mathbf{P}^2 \qquad (14)$$

□

As a consequence of (14) the factor $\mathbf{P}$ can be computed from readily available algorithms for factoring $\mathbf{A}^T\mathbf{A}$. This may be desirable when one of more of the following hold:

- Matrix $\mathbf{A}$ is large and sparse [37], as is frequently the case with social network adjacency matrices.
- Matrices $\mathbf{A}$ and $\mathbf{A}^T$ can be computed from their effects on the output, leading to martrix free methods.
- Efficient kernels for the incomplete Cholesky factorization of $\mathbf{A}^T\mathbf{A}$ are available [23] [38].

However, working directly with $\mathbf{A}^T\mathbf{A}$ leads to a more difficult problem from a numerical viewpoint as it has roughly the double condition number than $\mathbf{A}$, meaning that less digits of the result are to be considered as accurate. Therefore, when numerical accuracy matters, especially about the spectrum of $\mathbf{P}$, other strategies should be explored.

Matrix $\mathbf{A}^T\mathbf{A}$ appears in applications such as the normal equations in least squares fitting [39], regularization [40] and inverse problems [41], or in ML algorithms which are robust against noisy or even poisoned training datasets [42]. Moreover, in graph mining this matrix appears often in higher order vertex centrality metrics which take into consideration graph connectivity patterns. At the very core of these metrics there are linear algebraic kernels of the respective adjacency matrix. For instance the eigenvectors of $\mathbf{A}^T\mathbf{A}$ play a central role in some graph spectral partitioning algorithms [43]. Also a broad class of centrality metrics relies on a matrix power series of the form of (15):

$$\mathbf{W} = \sum_{k=0}^{+\infty}\sum_{j=1}^{2^k}\gamma_{k,j}\left[\mathbf{A},\mathbf{A}^T\right]_{k,j} = g\big(\mathbf{A},\mathbf{A}^T;\{\gamma_{k,j}\}\big) \qquad (15)$$

In (15) $\left[\mathbf{A},\mathbf{A}^T\right]_{k,j}$ denotes the j-th element of the set of products containing k terms of either $\mathbf{A}$ or $\mathbf{A}^T$. For instance when k equals two the corresponding set is that of (16):

$$\pi\left[\mathbf{A},\mathbf{A}^T\right] \triangleq \left\{\mathbf{A}^2, \mathbf{A}\mathbf{A}^T, \mathbf{A}^T\mathbf{A}, \left(\mathbf{A}^T\right)^2\right\} \qquad (16)$$

Evaluating the difference between such metrics applied on the original and the new adjacency matrix leads directly to a way to assess approximation goodness.

The spectra of $\mathbf{A}$ and $\mathbf{P}$ are connected in property 3.

Property 3. $\mathbf{A}$ and $\mathbf{P}$ have the same determinant up to sign.

Proof: Applying the polar factorization yields:

$$\begin{aligned}\det\left(\mathbf{P}^2\right) &= \det\left(\mathbf{A}^T\mathbf{A}\right) \Leftrightarrow \\ \det\left(\mathbf{P}\right)^2 &= \det\left(\mathbf{A}^T\right)\det\left(\mathbf{A}\right) \Rightarrow \\ \prod_{k=1}^{n}\mu_k^2 &= \prod_{k=1}^{n}\lambda_k^2 \Rightarrow \prod_{k=1}^{n}\mu_k = \prod_{k=1}^{n}\left|\lambda_k\right|\end{aligned}$$

□

Another way is to directly compute the determinant of the two sides of the polar factorization as shown below:

$$\begin{aligned}\mathbf{Q}^T\mathbf{Q} &= \mathbf{I} \Rightarrow \det\left(\mathbf{Q}\right)^2 = 1 \\ \det\left(\mathbf{A}\right) &= \det\left(\mathbf{QP}\right) = \det\left(\mathbf{Q}\right)\det\left(\mathbf{P}\right) \pm \det\left(\mathbf{P}\right)\end{aligned} \quad (17)$$

The preceding property is a much weaker condition to $\mathbf{A}$ and $\mathbf{P}$ representing two isospectral graphs. This can be a termination criterion for iterative schemes.

Property 4. Matrices $\mathbf{A}$ and $\mathbf{P}$ have the same number of zero eigenvalues. Also, their respective nullspaces coincide.

Proof: Let $\mathbf{g}_k$ be an eigenvector of $\mathbf{A}$ corresponding to a zero eigenvalue. Then by definition it holds that:

$$\mathbf{A}\mathbf{g}_k = \mathbf{QP}\mathbf{g}_k = \mathbf{0} \quad (18)$$

Since $\mathbf{Q}$ is by definition invertible, it follows that $\mathbf{g}_k$ belongs to the nullspace of $\mathbf{P}$. Conversely, for every eigenvector $\mathbf{u}_k$ for which $\mathbf{P}\mathbf{u}_k = \mathbf{0}$, it holds also that $\mathbf{QP}\mathbf{u}_k = \mathbf{0}$. □

4. Proposed Methodology

In this section the basic concepts underlying the proposed graph approximation method will be explained. The graph adjacency matrix is fundamental as the approximation process operates on this matrix rather than the on the combinatorial graph structure. This allows for viewing both global and local connectivity patterns and creating a balance between them as needed, perhaps even adaptively as the approximation process progresses. This is especially appealing when multiple graph abstraction levels are shought.

The proposed methodology differs from most existing approximation techniques in the following directions:

- Directed graphs are algebraically approximated instead of in a combinatorial fashion.
- It relies on high dimensional operations codified in the graph adjacency matrix.

Definition 2 (Graph adjacency matrix). The adjacency matrix $\mathbf{A}$ of an unweighted graph $G = (V, E)$, whether directed or undirected, is elementwise defined as in (19).

$$\mathbf{A}[i,j] \triangleq \begin{cases}1, & (i,j) \in E \\ 0, & (i,j) \notin E\end{cases} \in \{0,1\}^{|V|\times|V|} \quad (19)$$

Definition 3 (Graph spectrum). The eigenvalues of the respective adjacency matrix is the graph spectrum.

Graph spectra $\mathrm{sp}\left(\mathbf{A}\right)$ have the following properties [35]:

- There is one large eigenvalue followed by a second one, while the others have lower magnitudes according to a power law decay. Thus there is a heavy tail compared to the Gaussian or the Poisson kernels.
- Smaller eigenvalues tend to alternate in sign around zero, canceling thus each other. Thus $\mathrm{tr}\left(\mathbf{A}\right)$ and a number of functions dependent on $\mathrm{sp}\left(\mathbf{A}\right)$ can be approximated using only the few largest ones.
- With the exception of the few larger ones, the eigenvalues tend to be clustered. This can be contribute considerably to the acceleration of iterative methods such as the proposed one or the conjugate gradient.
- The scree plot of the eigenvalues, namely the plot of their (groupped) frequency vs their rank, also has a power law decay. This facilitates spectrum approximation as only a few ones are important.

By definition the adjacency matrix of an undirected graph $\mathbf{M}$ is symmetric and so $\mathrm{sp}\left(\mathbf{M}\right)$ is real. This is crucial in a graph mining context as $\mathrm{sp}\left(\mathbf{M}\right)$ is linked to properties such as community structure through the Fiedler number and expansion potential through the Cheeger number.

At the core of the proposed methodology is the iterative scheme for finding the unitary factor $\mathbf{Q}$ shown in (20). As stated in [4], it represents coupled Newton methods for the square roots of one starting from the singular values of $\mathbf{A}$.

$$\mathbf{Q}^{[k+1]} = \frac{1}{2}\left(\mathbf{Q}^{[k]} + \mathbf{Q}^{[k]^{-T}}\right), \quad \mathbf{Q}^{[0]} = \mathbf{A} \quad (20)$$

As an early $\mathbf{Q}^{[k]}$ can be close to singular, the Moore-Penrose inverse can replace the second term of the right hand side of (20). The latter is a working solution for many applications and it is acceptable since $\mathbf{Q}$ is invertible. The iteration (20) can be accelerated by multiplying $\mathbf{Q}^{[k]}$ by a scalar $\gamma^{[k]}$. A popular choice for $\gamma^{[k]}$ is given in [4]. It is shown in (21) and it is a tight estimation of $\left\|\mathbf{Q}\right\|_2$.

$$\gamma^{[k]} = \left(\frac{\left\|\mathbf{Q}^{[k]}\right\|_1\left\|\mathbf{Q}^{[k]}\right\|_\infty}{\left\|\mathbf{Q}^{[k]}\right\|_1\left\|\mathbf{Q}^{[k]}\right\|_\infty}\right)^{\frac{1}{4}} \quad (21)$$

Another scheme for selecting $\gamma^{[k]}$ is shown in (22). It moves along a similar line of reasoning with (21).

$$\gamma^{[k]} = \left(\frac{\left\|\mathbf{Q}^{[k]^{-1}}\right\|_F}{\left\|\mathbf{Q}^{[k]}\right\|_F}\right)^{\frac{1}{2}} \quad (22)$$

Regardless of the particular selection of the acceleration scalar, the general iteration form is now that of (23) which includes the previous one as a special case. As the iteration progresses the two terms of (23) converge to each other as a direct consequence of the orthogonalty of factor $\mathbf{Q}$.

$$\mathbf{Q}^{[k+1]} = \frac{1}{2}\left(\gamma^{[k]}\mathbf{Q}^{[k]} + \frac{1}{\gamma^{[k]}}\mathbf{Q}^{[k]^{-T}}\right) \quad (23)$$

The iteration terminates when the relative error of (24) drops below a prespecified threshold τ_0. This criterion also relies heavily on the orthogonality of polar factor $\mathbf{Q}$.

$$\left\|\mathbf{Q}^{[k]} - \mathbf{Q}^{[k]^{-T}}\right\|_F / \left\|\mathbf{Q}^{[k]}\right\|_F \leq \tau_0 \quad (24)$$

In algorithm 1 the proposed algorithm is described. Notice that the inversion process is not specified. Therefore, it can be the Newton inversion algorithm 2, the power method, or any other suitable scheme for that matter such as the Moore-Penrose inverse, the Drazin inverse, and the pseudoinverse. Once $\mathbf{Q}$ is obtained, then $\mathbf{P}$ can be found.

Algorithm 1 Polar factorization [4] [17] - Outer iteration

```
Require: Matrix $\mathbf{A}$ and termination criterion $\tau_0$
Ensure: Polar factorization $\mathbf{A} = \mathbf{QP}$
set $\mathbf{Q}^{[0]} \leftarrow \mathbf{A}$
repeat
    compute $\gamma^{[k]}$ as in (21) or (22)
    obtain ${\mathbf{Q}^{[k]}}^{-1}$ from algorithm 2
    compute $\mathbf{Q}^{[k+1]}$ as in (23)
until $\tau_0$ is true
return
```

The Newton method of algorithm 2 is an efficient method for computing the inverse $\mathbf{M}^{-1}$ of any invertible matrix $\mathbf{M}$. Moreover, when $\mathbf{M}$ is not invertible, then it converges to the Moore-Penrose inverse [44], which may be useful in the early outer iterations of algorithm 1 when $\mathbf{Q}^{[k]}$ may be close to singular. The starting matrix in each execution of algorithm 2 is $\mathbf{Q}^{[k]}$. The inner inversion iteration terminates when the absolute error drops under threshold τ_1.

$$\left\|\mathbf{Q}^{[k]}\mathbf{V}^{[j]} - \mathbf{I}_n\right\|_F \leq \tau_1 \qquad (25)$$

Algorithm 2 Newton matrix inversion [45] - Inner iteration

```
Require: Matrix $\mathbf{Q}^{[k]}$ and termination criterion $\tau_1$
Ensure: Inverse ${\mathbf{Q}^{[k]}}^{-1}$ or Moore-Penrose inverse
set $\mathbf{V}^{[0]} \leftarrow \mathbf{Q}^{[k]}$
repeat
    set $\mathbf{V}^{[j+1]} \leftarrow 2\mathbf{Q}^{[j]} - \mathbf{V}^{[j]}\mathbf{Q}^{[k]}\mathbf{V}^{[j]}$
until $\tau_1$ is true
return $\mathbf{V}^{[j]}$ as ${\mathbf{Q}^{[k]}}^{-1}$
```

Given the above algorithm 1 in conjunction with 2 essentially constitute a pair of nested loops. In the experiments of section 5 the iterations for each loop are reported separately.

5. Results

5.1. Data Synopsis

The proposed methodology was applied to three benchmark social graphs constructed with a topic sampling from Twitter [18] which was selected for the following reasons:

- The *follow* relationship is inherently directed, which forms the basis of many Twitter influence metrics.
- Twitter graphs, and the majority of social graphs for that matter, have a recursive community structure.
- Twitter API provides access to a vast number of data allowing their study as well as experiments on them.

The three benchmark graphs were created by collecting tweets from the US Twitter using the hashtags #*Julia*, #*Windows11*, and #*BlackList* for a period of two months, namely during June and July of 2021. The *follow* relationships were codified in directed unweighted social adjacency matrices. The structural and functional properties of the three Twitter benchmark social graphs are shown in table 2.

Observe that except from the tweet polarity profile the graphs have similar characteristics, facilitating conclusion drawing. Each topic was selected on the following grounds:

- Julia is a computing language for data intensive computations. Its official conference JuliaCon[1] 2021 in July attracted a high number of participations. It is a mostly neutral and positive tweets since they are generated by closely interlinked community members or enthusiasts.
- The release of Windows 11 scheduled for October 2021[2] has already created considerable dispute among its user base because of features such as UEFI secure boot and memory requirements. This can explain the large number of neutral tweets as well as the loose interconnection.
- *The Blacklist*[3] is a series known for the non-binary moral code of the characters and plot twists. The final season is expected in October 2021 among rampant fanbase arguments. The latter may be the cause of the emotional polarization and the high number of replies and mentions.

The tri-state, namely positive, negative, or neutral, emotional polarity of each tweet was derived from Python module *textblob*. Specifically, said polarity was calculated based on the majority of the polarity of their words.

The abovementioned graphs we stored in a free online MongoDB instance in the Atlas cloud[4], the mainstay of the MongoDB digital ecosystem offering Database-as-a-Service. Specifically Atlas offers among others visual dataset overview, access to a number of analytics as well as to monitoring tools, and connections to the stored collections based on the MongoDB shell. The proposed methodology was coded in Python 3.9 with the help of *numpy* module.

In table 3 the results from the various evaluation metrics of the proposed technique are shown and are marked with P. These metrics are described in detail in the remainder of this section. Moreover, the corresponding results of approximating the original graph by ignoring edge direction are also shown for comparison purposes marked with F.

From the entries of table 3 the following can be said:

- The scaled scheme of (21) terminates quicker systematically, even by a few iterations. However, in large scale applications even that may well be a significant performance advantage. The most likely ex-

[1] https://juliacon.org

[2] https://blogs.windows.com/windowsexperience/2021/08/31/windows-11-available-on-october-5/

[3] https://www.imdb.com/title/tt2741602

[4] https://www.mongodb.com/cloud/atlas

TABLE 2. TWITTER SOCIAL GRAPH PROPERTIES.

Graph	**#Julia**	**#Win11**	**#BlackList**	**Graph**	**#Julia**	**#Win11**	**#BlackList**
Property	**Value**	**Value**	**Value**	**Tweet property**	**Value**	**Value**	**Value**
Vertices	143019	152231	122535	Polarity% (pos/neg)	45.11/2.67	27.25/29.13	45.67/52.77
Edges	9232117	8536771	8425224	Length (mean/std)	167.33/45.12	145.17/37.83	154.86/41.84
Mean in-degree	66.21	61.89	72.43	Distinct hashtags	1182	1263	1314
Mean out-degree	71.36	63.18	76.08	Hashtags (mean/std)	5.13/0.89	8.42/1.17	7.18/1.01
Triangles	2458114	2282375	2946268	Replies (mean/std)	14.22/5.17	11.22/3.76	19.46/6.22
Squares	1034216	100736	117874	Mentions (mean/std)	17.63/4.38	13.38/3.29	15.49/5.34
Diameter	17	21	16	Density (linear/log)	64.55/1.35	56.08/1.38	68.76/1.36

TABLE 3. RESULTS FOR THE TWITTER GRAPHS.

Metric	**Def.**	**#Julia**	**#Win11**	**#BlackList**
Iterations (basic)	Eq. (20)	15	16	18
Iterations (ℓ_2)	Eq. (21)	10	11	12
Iterations (Frob.)	Eq. (22)	10	12	13
Density error P	Eq. (28)	0.1531	0.1835	0.2346
Density error F		0.3103	0.3562	0.4737
Logdensity error P		0.1833	0.2093	0.2274
Logdensity error F		0.2145	0.2566	0.2736
Fiedler error P	Eq. (31)	0.1744	0.1902	0.2016
Fiedler error F		0.1953	0.2353	0.2687
Hub Odd P	Eq. (35)	4.3418	5.0402	7.7519
Hub Odd F		6.8117	8.5562	11.8516
Auth. Odd P		4.1154	5.4572	8.4776
Auth. Odd F		6.2442	9.1374	12.0033
Hub Mercator P		5.7216	5.0333	7.3653
Hub Mercator F		6.5574	7.6470	10.4988
Auth. Mercator P		5.254	6.2221	7.9832
Auth. Mercator F		6.5519	7.4266	11.3321

planation is that this scheme often generates matches closer to the true orthogonal factor.

- Additionally the scaled scheme of (22) is a close second to the abovementioned one. Therefore, it may be advisable to seek a scaling scheme in order to accelerate the convergence of (20) as in the Richardson method. This may be a line search, a heuristic, a prediction filter, or an adaptive one.
- Based on the total number of outer iterations, the convergence speed of algorithm 1 seems to depend at least in part on the structural complexity of the approximated graph. More complex vertex interaction patterns take longer to be discovered and encoded in the polar factors in addition to their own.
- For the *#Blacklist* graph in spite of the different number of iterations there is basic pattern of an initial large error followed by a quick drop in all three variations of the iterative scheme as seen in figure 2. This can be attributed to the quadratic convergence rate of the Newton method.
- Ignoring edge direction introduces significant spurious graph structure in the form of additional edges. In turn, this translates to distorted community structure and density. On the contrary, the proposed methodology may delete some directed edges to preserve a structural balance up to a point.
- Vertex centrality, which is a global property, is also heavily influenced by this addition of spurious edges and it changes considerably. On the other hand, the proposed technique respects centrality up to a certain degree resulting in a much better approximation.

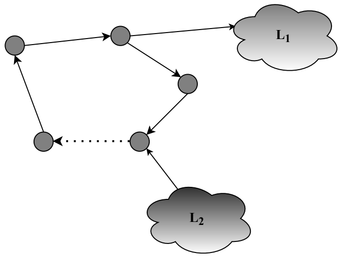

Figure 1. Removing an edge from a directed cycle.

In order to understand the distortion caused by obliterating edge direction consider the case of a directed graph bridge. Almost by definition is a weak link, especially if there is no way to return to one of the two partitions it connects. Removing the direction information results in a much reinforced link which furthermore may not even have a meaning in the underlying domain. Thus, in such a case not only graph structure but also semantics are strongly violated. A similar scenario is that of a directed cycle which may represent a set of interdependent restrictions which are essentially lifted in the resulting undirected graph. The proposed technique may break this cycle by removing one or more of its edges, essentially respecting at least the condition that the cycle cannot be traversed from the proper direction. Of course the cycle remains inaccessible from the other side, but the original restriction is still in place. However, if said cycle is part of the critical graph paths as for instance in the scenario of figure 1, then both the proposed and the baseline approximation methodologies severely undermine the graph overall structural integrity.

Finally, regarding the parameters used in the experiments of this work thresholds τ_1, τ_2, and τ_3 were set to 5%. Notice that $\mathbf{P}$ cannot be readily used since its entries do not in the general case are not either 0 or 1. In this work a threshold of 0.5 is used to determine the final values. Also let $\mathbf{A}_F$ denote the matrix obtained from removing edge direction information in the original adjacency graph. The Kullback-Leibler logarithm base b in equation (35) is two.

5.2. Number Of Iterations

In figure 2 is shown the graph reconstruction error $E_r^{[k]}$ of (26) in terms of the Frobenius norm vs the iterations

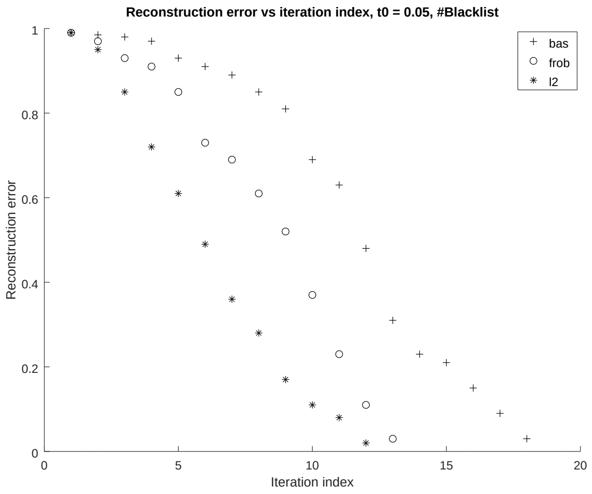

Figure 2. Number of outer iterations for the #Blacklist graph.

is shown. It represents the root mean square (rms) error between the original adjacency matrix and its factored form.

$$E_r^{[k]} \triangleq \frac{1}{n} \left\| \mathbf{A} - \mathbf{Q}^{[k]} \mathbf{P}^{[k]} \right\|_F \tag{26}$$

Because of the square root the Frobenius norm is divided by n instead of n^2, namely the total number of elements of $\mathbf{A}$. The interpretation of (26) is that it represents the residual energy between the original adjacency matrix and its approximation during the k-th iteration of algorithm 1.

5.3. Sparsity

In order to evaluate the performance of the approximation process a number of benchmarks will be used. One of the most indicative is the pair of density and logdensity which reveal up to an extent the graph expension potential as graphs with low both values can be easily enlarged through strategic edge addition. Additionally, density is frequently a good estimate of the mean in- and out-degrees.

Definition 4 (Density and logdensity). The density ρ_0 of a graph is defined as the ratio of the number of edges to the number of vertices. Similarly the logdensity ρ_0' is the logarithm, in any base, of the number of edges to the logarithm in the same base of the number of vertices.

$$\rho_0 \triangleq \frac{|E|}{|V|} \qquad \text{and} \qquad \rho_0' \triangleq \frac{\log|E|}{\log|V|} \tag{27}$$

One approximation evaluation metric is the relative error between the (log)density of the original adjacency matrix and its symmetric polar factor as shown in (28). Similar relative errors can be defined for the (log)density of the graph resulting from the baseline methodology.

$$\left| \frac{\rho_0 - \rho_{0,p}}{\rho_0} \right| \qquad \text{and} \qquad \left| \frac{\rho_0' - \rho_{0,p}'}{\rho_0'} \right| \tag{28}$$

Two alternative measures of graph density, namely the (log)completeness, are described in definition 5.

Definition 5 (Completeness and logcompleteness). The (log)completeness of a directed graph is the ratio of (the logarithm of) its number of edges to the (logarithm of the) number of edges of the full graph with the same number of vertices as shown in equation (29).

$$\begin{aligned} \sigma_0 &\triangleq \frac{|E|}{|V|(|V|-1)} \approx \frac{|E|}{|V|^2} = \frac{\rho_0}{|V|} \\ \sigma_0' &\triangleq \frac{\log|E|}{\log\left(|V|(|V|-1)\right)} \approx \frac{\log|E|}{2\log|V|} = \frac{\rho_0'}{2} \end{aligned} \tag{29}$$

From the connection between equations (28) and (29) it follows that only the former needs to be computed.

5.4. Fiedler Value

A scalar associated with the community structure of each graph is the Fiedler value λ^* which is defined as the smallest non-zero eigenvalue of the graph Laplacian. The latter is an important matrix describing normalized second order graph connectivity and its form is given in definition 6.

Definition 6 (Graph Laplacian matrix). The Laplacian matrix $\mathbf{L} \in \mathbb{R}^{n \times n}$ of a graph is defined as in (30).

$$\mathbf{L} \triangleq \mathbf{I} - \operatorname{diag}\left[\deg(v_1), \ldots, \deg(v_n)\right]^{-1} \mathbf{A} \tag{30}$$

The Laplacian spectrum has the following properties:

- The smallest eigenvalue always equals zero. Each non-empty graph has at least one component.
- Each additional zero eigenvalue denotes a connected component, signifying a disconnected graph.
- The smallest non-zero eigenvalue λ^* is a measure of the overall graph partitioning potential.
- The eigenexpansion of the graph Laplacian matrix is defined as the Laplace transform of the graph.

Once the polar factor $\mathbf{P}$ is available, the relative error between the Fiedler value of the original adjacency matrix λ^* and the corresponding of said factor λ_p^* is given in (31):

$$\left| \frac{\lambda^* - \lambda_p^*}{\lambda^*} \right| \tag{31}$$

Given graph Laplacian $\mathbf{L}$ how can its Fiedler value be efficiently computed? One possible answer is the inverse power method, a matrix free iterative method shown in 3. It works by iteratively estimating the eigenvector corresponding to the smallest eigenvalue. Then the Rayleigh quotient is used to obtain an estimate of the corresponding eigenvalue. A rank one downdate is subsequently used to exclude the estimated eigenvalue from the spectral decomposition of $\mathbf{L}$.

In order to accelerate convergence it is possible to select a small number of starting points and operate on the until one of them starts converging. No matrix inversions are necessary and they are replaced by linear system solvers.

Since the Laplacian of the polar factor is symmetric, the respective right and left eigenvectors coincide. On the other hand, the Laplacian of the original adjacency matrix is non-symmetric and hence the right and left eigenvectors need

Algorithm 3 Inverse power method

```
Require: Matrix $\mathbf{M}$ and termination criterion $\tau_2$
Ensure: Return the smallest eigenvalue and its eigenvector
  select random starting point $\mathbf{g}^{[0]}$
  set $\mathbf{g}^{[0]} \leftarrow \mathbf{g}^{[0]} / \left\|\mathbf{g}^{[0]}\right\|_1$
  repeat
    set $\mathbf{g}^{[k+1]} \leftarrow \mathbf{M}^{-1}\mathbf{g}^{[k]}$
    set $\mathbf{g}^{[k+1]} \leftarrow \mathbf{g}^{[k+1]} / \left\|\mathbf{g}^{[k+1]}\right\|_1$
  until $\tau_2$ is true
  set $\lambda \leftarrow \left(\mathbf{g}^{[k+1]^T}\mathbf{M}^{-1}\mathbf{g}^{[k+1]}\right) / \left(\mathbf{g}^{[k+1]^T}\mathbf{g}^{[k+1]}\right)$
  return $\lambda$, $\mathbf{g}^{[k+1]}$
```

to be separately computed as they are different from each other. Hence a connected graph, like the benchmark Twitter graphs, requires six inverse matrix iterations in total.

5.5. Vertex Centrality Distribution

In order to evaluate the effect of replacing the original directed adjacency matrix $\mathbf{A}$ with the respective undirected polar factor $\mathbf{P}$ on the certain centality metrics two common matrix power series based methods will be used here.

In the case of the odd power centrality or Estrada centrality the power series of equation (32) is formed:

$$\mathbf{W}_e \triangleq \sum_{k=0}^{+\infty} \frac{1}{2k+1}\mathbf{A}^{2k+1} = \sinh \mathbf{A} \tag{32}$$

The rationale behind selecting only odd powers of the adjacency matrix is that closed paths of even length are more probable to correspond to oscillations of length two. On the other hand, closed paths of odd length are much less likely to. Once $\mathbf{W}_e$ is computed, the i-th diagonal element of its inverse is the centrality of the i-th vertex.

The Mercator power series is among the few vertex centality metrics which have negative terms. It is inspired from the Mercator projection commonly used in topological transforms as well as in real world maps. In a graph mining context the Mercator projection is useful since it preserves local properties including connectivity patterns.

$$\mathbf{W}_m \triangleq \sum_{k=1}^{+\infty} \frac{(-1)^{k+1}}{k}\mathbf{A}^{k} \triangleq \ln\left(\mathbf{I}_n + \mathbf{A}\right) \tag{33}$$

Similarly to (32) the centrality of the i-th vertex is the i-th diagonal element of the inverse of matrix $\mathbf{W}_m$.

Since in a directed graph a vertex has a distinct role as a hub and an authority which is respectively codified by the adjacency matrix and its transpose, the above metrics should be computed for both $\mathbf{A}$ and $\mathbf{A}^T$. On the contrary, for $\mathbf{P}$ and $\mathbf{A}_F$ these metrics need only be computed once.

One way to compare the centrality values of (32) or (33) when computed for $\mathbf{A}$, $\mathbf{A}^T$, $\mathbf{P}$, and $\mathbf{A}_F$ is to create the distribution of (34) for each possible case. For instance, for the Estrada metric and $\mathbf{A}$ the distribution is:

$$p_i \triangleq \frac{\mathbf{W}_e^{-1}[i,i]}{\sum_{j=1}^{n}\mathbf{W}_e^{-1}[j,j]} = \frac{\mathbf{W}_e^{-1}[i,i]}{\operatorname{tr}\left(\mathbf{W}_e^{-1}\right)} \tag{34}$$

When the distributions to be compared, they can be compared with the Kullback-Leibler divergence of (35):

$$\langle p \,||\, q \rangle \triangleq \sum_i p_i \log_b\left(\frac{p_i}{q_i}\right) \tag{35}$$

In (35) the cost of replacing the probability mass function p_1 with p_2 is computed. The sum therein ranges over the support of p_1 with i denoting the i-th event, while the logarithm base b determines the divergence units.

6. Conclusions

This conference paper focuses on approximating large directed graphs with directed ones. Specifically, the original adjacency matrix is replaced by the symmetric and positive (semi)definite factor of its polar factorization. To this end three variations of an existing iterative scheme are used. Intuituion is provided by interpreting the linear algebraic properties of polar factorization in a graph signal processing context. The convergence in terms of iterations as well as the goodness of approximation based on graph density, Fiedler value and primary eigenvector, and power series vertex centrality using as benchmarks three directed Twitter graphs obtained with topic sampling. The proposed approximation can be applied to higher dimensionality problems such as community structure discovery and friend recommendation.

This work can be extended in a number of ways. First, approximation techniques for graph multiresolution abstraction scenaria can be developed. Additionally, graph approximation criteria explicity taking into consideration graph density or other structural or even functional properties can be designed. This can be extended to techniques for multiple and perhaps competing criteria in order to achieve a balanced result as the final graph. From a computational perspective higher order iterative methods for the orthogonal factor can achieve the same result with fewer iterations.

Acknowledgment

This work was supported by the Research Initiative Fund (RIF) Grant R18087 by Zayed University, UAE.

References

[1] M. Shao, J. Li, F. Chen, H. Huang, S. Zhang, and X. Chen, “An efficient approach to event detection and forecasting in dynamic multivariate social media networks,” in *WWW*, 2017, pp. 1631–1639.

[2] E. Atik, A. El Fattah, A. Nawar, and M. Atef, “Rough approximation models via graphs based on neighborhood systems,” *Granular computing*, vol. 6, no. 4, pp. 1025–1035, 2021.

[3] N. Entezari, S. A. Al-Sayouri, A. Darvishzadeh, and E. E. Papalexakis, “All you need is low (rank) defending against adversarial attacks on graphs,” in *The 13th International Conference on Web Search and Data Mining*, 2020, pp. 169–177.

[4] N. J. Higham, “Computing polar decompositon with applications,” *J. Sci. Stat. Comput.*, vol. 7, no. 4, pp. 1160–1174, 1986.

[5] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, “Graph signal processing: Overview, challenges, and applications,” *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[6] X. Dong, D. Thanou, L. Toni, M. Bronstein, and P. Frossard, “Graph signal processing for machine learning: A review and new perspectives,” *IEEE Signal processing magazine*, vol. 37, no. 6, pp. 117–127, 2020.

[7] V. N. Ekambaram, G. C. Fanti, B. Ayazifar, and K. Ramchandran, “Multiresolution graph signal processing via circulant structures,” in *DSP/SPE*. IEEE, 2013, pp. 112–117.

[8] Y. Bai, F. Wang, G. Cheung, Y. Nakatsukasa, and W. Gao, “Fast graph sampling set selection using Gershgorin disc alignment,” *IEEE Transactions on signal processing*, vol. 68, pp. 2419–2434, 2020.

[9] L. Chen, Y. Xie, Z. Zheng, H. Zheng, and J. Xie, “Friend recommendation based on multi-social graph convolutional network,” *IEEE Access*, vol. 8, pp. 43 618–43 629, 2020.

[10] L. Stanković, D. Mandic, M. Daković, B. Scalzo, M. Brajović, E. Sejdić, and A. G. Constantinides, “Vertex-frequency graph signal processing: A comprehensive review,” *Digital signal processing*, 2020.

[11] G. Drakopoulos, E. Kafeza, P. Mylonas, and L. Iliadis, “Transform-based graph topology similarity metrics,” *NCAA*, vol. 1, no. 1, 2021.

[12] G. Drakopoulos, I. Giannoukou, P. Mylonas, and S. Sioutas, “A graph neural network for assessing the affective coherence of Twitter graphs,” in *IEEE Big Data*. IEEE, 2020, pp. 3618–3627.

[13] W. Cong, R. Forsati, M. Kandemir, and M. Mahdavi, “Minimal variance sampling with provable guarantees for fast training of graph neural networks,” in *KDD*, 2020, pp. 1393–1403.

[14] Z. Guo, L. Tang, T. Guo, K. Yu, M. Alazab, and A. Shalaginov, “Deep graph neural network-based spammer detection under the perspective of heterogeneous cyberspace,” *Future generation computer systems*, vol. 117, pp. 205–218, 2021.

[15] E. Ceci and S. Barbarossa, “Graph signal processing in the presence of topology uncertainties,” *IEEE Transactions on signal processing*, vol. 68, pp. 1558–1573, 2020.

[16] Y. Brenier, “Polar factorization and monotone rearrangement of vector-valued functions,” *Communications on pure and applied mathematics*, vol. 44, no. 4, pp. 375–417, 1991.

[17] N. J. Higham and R. S. Schreiber, “Fast polar decomposition of an arbitrary matrix,” *J. Sci. Stat. Comput.*, vol. 11, no. 4, pp. 648–655, 1990.

[18] G. Drakopoulos, “Tensor fusion of social structural and functional analytics over Neo4j,” in *IISA*. IEEE, 2016.

[19] H. Tong, B. A. Prakash, T. Eliassi-Rad, M. Faloutsos, and C. Faloutsos, “Gelling, and melting, large graphs by edge manipulation,” in *CIKM*, 2012, pp. 245–254.

[20] J. Fang and F. Y. Partovi, “A hits-based model for facility location decision,” *Expert Systems with Applications*, vol. 159, 2020.

[21] X. Yu, N. Lei, X. Zheng, and X. Gu, “Surface parameterization based on polar factorization,” *Journal of computational and applied mathematics*, vol. 329, pp. 24–36, 2018.

[22] M. T. Jones and P. E. Plassmann, “An improved incomplete Cholesky factorization,” *TOMS*, vol. 21, no. 1, pp. 5–17, 1995.

[23] T. A. Nhan and N. Madden, “An analysis of diagonal and incomplete Cholesky preconditioners for singularly perturbed problems on layer-adapted meshes,” *Journal of Applied Mathematics and Computing*, vol. 65, no. 1, pp. 245–272, 2021.

[24] E. S. Gawlik and M. Leok, “Interpolation on symmetric spaces via the generalized polar decomposition,” *Foundations of computational mathematics*, vol. 18, no. 3, pp. 757–788, 2018.

[25] D. Sukkari, H. Ltaief, M. Faverge, and D. Keyes, “Asynchronous task-based polar decomposition on single node manycore architectures,” *IEEE Transactions on parallel and distributed systems*, vol. 29, no. 2, pp. 312– 323, 2017.

[26] H. Guo and M. Jerrum, “A polynomial-time approximation algorithm for all-terminal network reliability,” *SIAM Journal on computing*, vol. 48, no. 3, pp. 964–978, 2019.

[27] Z. Wu, H. R. Karimi, and C. Dang, “An approximation algorithm for graph partitioning via deterministic annealing neural network,” *Neural networks*, vol. 117, pp. 191–200, 2019.

[28] R. Shafipour, A. Khodabakhsh, G. Mateos, and E. Nikolova, “A directed graph Fourier transform with spread frequency components,” *IEEE Transactions on signal processing*, vol. 67, no. 4, pp. 946–960, 2018.

[29] M. B. Cohen, J. Kelner, J. Peebles, R. Peng, A. B. Rao, A. Sidford, and A. Vladu, “Almost-linear-time algorithms for Markov chains and new spectral primitives for directed graphs,” in *Symposium on theory of computing*, 2017, pp. 410–419.

[30] R. Chitnis, A. E. Feldmann, and P. Manurangsi, “Parameterized approximation algorithms for bidirected Steiner network problems,” *ACM TALG*, vol. 17, no. 2, pp. 1–68, 2021.

[31] A. Backurs, L. Roditty, G. Segal, V. V. Williams, and N. Wein, “Toward tight approximation bounds for graph diameter and eccentricities,” *SIAM Journal on computing*, vol. 50, no. 4, pp. 1155–1199, 2021.

[32] G. Drakopoulos, S. Kontopoulos, and C. Makris, “Eventually consistent cardinality estimation with applications in biodata mining,” in *SAC*. ACM, 2016.

[33] S. Kontopoulos and G. Drakopoulos, “A space efficient scheme for graph representation,” in *ICTAI*. IEEE, 2014.

[34] J. Scott and M. Tuma, “A computational study of using black-box QR solvers for large-scale sparse-dense linear least squares problems,” *ACM TOMS*, 2020.

[35] J. Cape, “Spectral analysis of networks with latent space dynamics and signs,” *Stat*, vol. 10, no. 1, 2021.

[36] G. Drakopoulos, P. Mylonas, and S. Sioutas, “A case of adaptive non-linear system identification with third order tensors in TensorFlow,” in *INISTA*. IEEE, 2019.

[37] F. Schäfer, M. Katzfuss, and H. Owhadi, “Sparse Cholesky factorization by Kullback–Leibler minimization,” *SIAM Journal on scientific computing*, vol. 43, no. 3, pp. A2019–A2046, 2021.

[38] L. Chen, S. Zhou, J. Ma, and M. Xu, “Fast kernel k-means clustering using incomplete Cholesky factorization,” *Applied Mathematics and Computation*, vol. 402, 2021.

[39] S. E. Ahmed, D. Aydin, and E. Yilmaz, “Estimating the nonparametric regression function by using Padé approximation based on total least squares,” *Numerical functional analysis and optimization*, vol. 41, no. 15, pp. 1827–1870, 2020.

[40] Y. Ding, J. Tang, and F. Guo, “Identification of drug–target interactions via dual Laplacian regularized least squares with multiple kernel fusion,” *Knowledge-Based Systems*, vol. 204, 2020.

[41] S. Dittmer, T. Kluth, P. Maass, and D. O. Baguer, “Regularization by architecture: A deep prior approach for inverse problems,” *Journal of Mathematical Imaging and Vision*, vol. 62, no. 3, pp. 456–470, 2020.

[42] F. Farokhi, “Why does regularization help with mitigating poisoning attacks?” *Neural Processing Letters*, pp. 1–13, 2021.

[43] X. Zhu, S. Zhang, Y. Li, J. Zhang, L. Yang, and Y. Fang, “Low-rank sparse subspace for spectral clustering,” *IEEE Transactions on knowledge and data engineering*, vol. 31, no. 8, pp. 1532–1543, 2018.

[44] H. Akaike, “Block Toeplitz matrix inversion,” *Journal on applied mathematics*, vol. 24, no. 2, pp. 234–241, 1973.

[45] R. Byers and H. Xu, “A new scaling for Newton’s iteration for the polar decomposition and its backward stability,” *Journal on matrix analysis and applications*, vol. 30, no. 2, pp. 822–843, 2008.