Matrix Resolvent Eigenembeddings for Dynamic Graphs

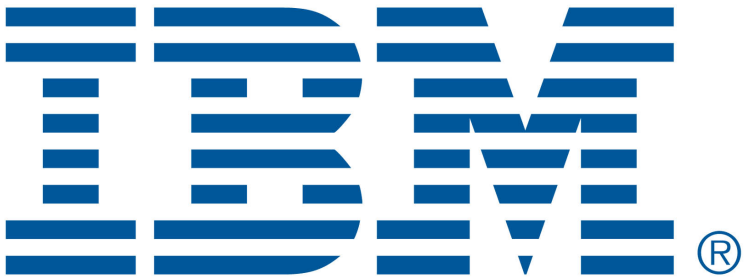
Vassilis Kalantzis and Panagiotis Traganitis

May 2023

EPrint ID: 2023.2

IBM Research
Thomas J. Watson Research Center

Preprints available from:
https://researcher.watson.ibm.com/researcher/view.php?person=ibm-vkal



MATRIX RESOLVENT EIGENEMBEDDINGS FOR DYNAMIC GRAPHS

Vasileios Kalantzis[⋆] *Panagiotis A. Traganitis*[†]

[⋆] IBM Research, Thomas J. Watson Research Center, Yorktown Heights, NY, USA
[†]Dept. of Electrical and Computer Engineering, Michigan State University, East Lansing, MI, USA

ABSTRACT

Eigenvector embeddings have been widely used to study graph properties in signal processing, mining, and learning tasks. However, if a graph is changing dynamically, these embeddings have to be recomputed. In this work we introduce a novel matrix resolvent expansion-based projection scheme to update eigenvector embeddings of dynamic graphs. The proposed method can tackle graph updates where both new vertices and edges are added, and its potential is illustrated via numerical tests on real data.

Index Terms— Eigenvector, Embedding, Graph, Dynamic, Matrix Resolvent

1. INTRODUCTION

Graphs are mathematical constructs that are used to succinctly describe complex systems or networks, such as power, social and brain networks to name a few. Due to their central role in network science a plethora of tools have been developed to characterize graph properties and perform data mining and machine learning tasks. Multiple state-of-the-art approaches that describe properties of a graph or network, or perform learning tasks rely on so-called embeddings, that is, vector representations of the vertices of a given graph. Of particular interest in this paper, are eigenvector embeddings (termed henceforth eigenembeddings) of graphs, that, as the name suggests, are derived from the eigenvectors of adjacency matrices. Notable applications of eigenembeddings include *node or vertex centrality* [1, 2, 3] and *spectral clustering* [4, 5] among others. Variants of eigenvector centrality include PageRank [6], exponential subgraph centrality [7] and Katz centrality [8].

Despite the popularity of these eigenvector embeddings, if a change occurs in the graph, i.e., a set of edges or vertices is either added or removed, these embeddings have to be computed from scratch, leading to potentially intractable computational costs for graphs that change frequently. Thus, methods that can rapidly update the eigenvector embeddings of dynamic graphs without performing a full eigendecomposition are well motivated.

A perturbation theory based algorithm was proposed in [9] to track the top-k eigenpairs of a dynamic graph. Similar approaches were also considered in [10, 11]. For spectral clustering of dynamic graphs [12] introduced a compressive method, while [13] used a perturbation theory approach. Most similar to this work, an incremental projection-based SVD update method was advocated in [14].

In this work, we consider a projection-based approach to update the eigenembeddings of evolving graphs where new vertices are constantly added to the system. Compared to the prior art, the proposed method approximates the top-k eigenpairs of the updated adjacency matrix via a Rayleigh-Ritz projection onto a subspace formed by matrix resolvent expansions. Similar techniques have been considered for the update of top-k singular value decompositions [15], however this is the first time that matrix resolvent expansions are considered for eigenvalue problems; and in particular those stemming from tasks in graph analysis.

Notation: Lowercase bold letters, $\boldsymbol{x}$, denote vectors, uppercase bold letters, $\mathbf{X}$, represent matrices, and calligraphic uppercase letters, $\mathcal{X}$, stand for sets. The (i,j)-th entry of matrix $\mathbf{X}$ is denoted by $[\mathbf{X}]_{ij}$. $\mathrm{Ran}(\mathbf{X})$ and $\mathrm{Rank}(\mathbf{X})$ denote the rangespace and rank of $\mathbf{X}$, respectively, and $|\mathcal{X}|$ denotes the cardinality of set $\mathcal{X}$. '$\oplus$' denotes the direct sum between two subspaces, and $\mathbf{I}$ and $\mathbf{0}$ the identity and all-zeroes matrices of appropriate dimension, respectively.

2. PROBLEM STATEMENT AND PRELIMINARIES

Consider at time-step 't' a graph $\mathcal{G}_t := \{\mathcal{V}_t, \mathcal{E}_t\}$, where $\mathcal{V}_t, \mathcal{E}_t$ denote its vertex (or node) and edge sets respectively, and let $n_t := |\mathcal{V}_t|$ denote the number of vertices. Associated with $\mathcal{G}_t$ is a $n_t \times n_t$ adjacency matrix $\mathbf{A}_t$, that encodes the connectivity between vertices in $\mathcal{V}_t$, i.e., $[\mathbf{A}_t]_{i,j} = 1$ if $(i,j) \in \mathcal{E}_t$, and $[\mathbf{A}_t]_{i,j} = 0$ otherwise. Let $\{(\lambda_{t,i}, \mathbf{x}_{t,i})\}_{i=1}^{k}$ denote the k leading eigenpairs of $\mathbf{A}_t$, with $\lambda_{t,i}$ being the i-th eigenvalue and $\mathbf{x}_{t,i}$ the corresponding eigenvector, where $\lambda_{t,1} \geq \ldots \geq \lambda_{t,k}$, and also, define $\mathbf{X}_t := [\mathbf{x}_{t,1}, \ldots, \mathbf{x}_{t,k}]$ and $\boldsymbol{\lambda}_t := [\lambda_{t,1}, \ldots, \lambda_{t,k}]$.

In this work we focus exclusively in scenarios where at time-step '$t+1$', the graph $\mathcal{G}_t$ is updated to graph $\mathcal{G}_{t+1} := \{\mathcal{V}_{t+1}, \mathcal{E}_{t+1}\}$, under the constraint $\mathcal{G}_t \subset \mathcal{G}_{t+1}$, i.e., the graph $\mathcal{G}_t$ is an induced subgraph of $\mathcal{G}_{t+1}$, thus $\mathcal{V}_t \subseteq \mathcal{V}_{t+1}$ and $\mathcal{E}_t \subseteq \mathcal{E}_{t+1}$. In addition, $n_{t+1} := |\mathcal{V}_{t+1}| = n_t + s_{t+1}$, where s_{t+1} is the number of newly added vertices. Since $\mathcal{G}_t$ is an induced subgraph of $\mathcal{G}_{t+1}$, $\mathbf{A}_t$ is a leading principal submatrix of the adjacency matrix $\mathbf{A}_{t+1}$ associated with the graph $\mathcal{G}_{t+1}$. Specifically, the adjacency matrix at time-step '$t+1$' can be written as

$$\mathbf{A}_{t+1} = \begin{bmatrix} \mathbf{A}_t & \mathbf{A}_{t,t+1} \\ \mathbf{A}_{t,t+1}^\top & \mathbf{A}_{t+1,t+1} \end{bmatrix}, \tag{1}$$

where $\mathbf{A}_{t,t+1} \in \mathbb{R}^{n_t \times s_{t+1}}$ encodes the coupling between the existing n_t vertices and the newly added s_{t+1} vertices, and $\mathbf{A}_{t+1,t+1} \in \mathbb{R}^{s_{t+1} \times s_{t+1}}$ captures the connectivity between the newly added s_{t+1} vertices. Our objective is to update the k leading eigenpairs of $\mathbf{A}_t$ to the k leading eigenpairs of $\mathbf{A}_{t+1}$, for $t = 1, \ldots, T-1$, with T denoting the maximum number of graph updates.

3. EIGENPAIRS OF EVOLVING GRAPHS

To develop an algorithm for updating the eigenvalues and eigenvectors of $\mathbf{A}_{t+1}$ we will rely on the so-called Rayleigh-Ritz (RR) approximation procedure [16]. Consider an orthonormal basis matrix

Emails: vkal@ibm.com, traganit@msu.edu

$\mathbf{Z}_{t+1}$ and let $\{(\tau_{t+1,i}, \mathbf{h}_{t+1,i})\}_{i=1}^{k}$, denote the k leading eigenpairs of the matrix $\mathbf{Z}_{t+1}^{\top}\mathbf{A}_{t+1}\mathbf{Z}_{t+1}$, where $\tau_{t+1,1} \geq \tau_{t+1,2} \geq \ldots \geq \tau_{t+1,k}$. Then the i-th leading eigenpair $(\lambda_{t+1,i}, \mathbf{x}_{t+1,i})$ of $\mathbf{A}_{t+1}$ can be approximated by the i-th leading Ritz pair $(\tau_{t+1,i}, \mathbf{Z}_{t+1}\mathbf{h}_{t+1,i})$. In fact, when $\mathbf{x}_{t+1,i} \in \mathrm{Ran}(\mathbf{Z}_{t+1})$, $\lambda_{t+1,i} = \tau_{t+1,i}$ and $\mathbf{x}_{t+1,k} = \mathbf{Z}_{t+1}\mathbf{h}_{t+1,i}$ [17, Section 11]. Thus proper design of $\mathbf{Z}_{t+1}$ will enable efficient and accurate computation of the leading eigenpairs of $\mathbf{A}_{t+1}$. A summary of the proposed framework is listed as Alg. 1, and the ensuing sections showcase how to construct such a $\mathbf{Z}_{t+1}$ from the leading eigenpairs of $\mathbf{A}_t$.

Algorithm 1 Eigen-update algorithm.

Input: $\mathbf{A}_1 \in \mathbb{R}^{n_1 \times n_1},\ k \in \mathbb{N}$
Output: $\mathbf{X}_T,\ \mathbf{\Lambda}_T$
▷ Set $t \leftarrow 1$ and compute $(\mathbf{X}_t, \boldsymbol{\lambda}_t)$
do
  ▷ Receive $\mathbf{A}_{t,t+1}$, $\mathbf{A}_{t+1,t+1}$
  ▷ Build projection matrix $\mathbf{Z}_{t+1}$
  ▷ Compute the k leading Ritz pairs $(\mathbf{X}_{t+1}, \boldsymbol{\lambda}_{t+1})$ of $\mathbf{A}_{t+1}$
  ▷ Set $\mathbf{A}_{t+1,t+1} \leftarrow \mathbf{A}_{t+1}$, and $t \leftarrow t+1$
while there exist graph updates

3.1. An exact subspace when $\mathrm{Rank}(\mathbf{A}_t) = k$

Since any eigenvector $\mathbf{x}$ associated with a nonzero eigenvalue λ of the matrix $\mathbf{A}_{t+1}$ must lie in its range [16], the optimal $\mathbf{Z}_{t+1}$ is a basis of $\mathrm{Ran}(\mathbf{A}_{t+1})$. However, computing such a basis at every graph update is impractical since $\mathrm{Rank}\,(\mathbf{A}_{t+1})$ may be too large. Nevertheless, as the next proposition will show, one can easily construct an optimal $\mathbf{Z}_{t+1}$ from $\mathbf{X}_t$, under the assumption $\mathrm{Rank}(\mathbf{A}_t) = k$.

Proposition 1. *Let $\mathbf{A}_{t+1}$ be defined as in (1), assume $\mathrm{Rank}(\mathbf{A}_t) = k$, and consider an orthonormal matrix $\mathbf{Q}_{t+1}$ that satisfies $\mathrm{Ran}(\mathbf{Q}_{t+1}) = \mathrm{Ran}((\mathbf{I} - \mathbf{X}_t\mathbf{X}_t^{\top})\mathbf{A}_{t,t+1})$. Then*

$$\mathrm{Ran}(\mathbf{A}_{t+1}) \subseteq \mathrm{Ran}\left(\begin{bmatrix}\mathbf{X}_t & \mathbf{Q}_{t+1} & \mathbf{0}\\ & \mathbf{0} & \mathbf{I}\end{bmatrix}\right). \tag{2}$$

Proof. The matrix $\mathbf{A}_{t+1}$ can be written as

$$\mathbf{A}_{t+1} = \begin{bmatrix}\mathbf{A}_t & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix} + \begin{bmatrix}\mathbf{0} & \mathbf{A}_{t,t+1}\\ \mathbf{A}_{t,t+1}^{\top} & \mathbf{0}\end{bmatrix} + \begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{A}_{t+1,t+1}\end{bmatrix}. \tag{3}$$

By definition, $\mathrm{Ran}(\mathbf{A}_{t+1})$ lies in the union of the range spaces of the three matrices on the right-hand side in (3). Since $\mathrm{Rank}(\mathbf{A}_t) = k$, it follows that $\mathrm{Ran}(\mathbf{A}_t) = \mathrm{Ran}(\mathbf{X}_t)$. Moreover, notice that $\mathrm{Ran}(\mathbf{A}_{t+1,t+1}) \subseteq \mathrm{Ran}(\mathbf{I})$ and

$$\mathrm{Ran}\left(\begin{bmatrix}\mathbf{0} & \mathbf{A}_{t,t+1}\\ \mathbf{A}_{t,t+1}^{\top} & \mathbf{0}\end{bmatrix}\right) \subseteq \mathrm{Ran}\left(\begin{bmatrix}\mathbf{A}_{t,t+1}\\ \mathbf{0}\end{bmatrix}\right) \oplus \mathrm{Ran}\left(\begin{bmatrix}\mathbf{0}\\ \mathbf{I}\end{bmatrix}\right).$$

Finally, note that $\mathrm{Ran}\,([\mathbf{X}_t,\ \mathbf{A}_{t,t+1}]) \equiv \mathrm{Ran}\,([\mathbf{X}_t,\ \mathbf{Q}_{t+1}])$. □

Prop. 1 suggests that when $\mathrm{Rank}(\mathbf{A}_t) = k$, the exact k leading eigenpairs of $\mathbf{A}_{t+1}$ can be computed via a RR projection [16] onto the orthonormal basis

$$\mathbf{Z}_{t+1} = \begin{bmatrix}\mathbf{X}_t & \mathbf{Q}_{t+1} & \mathbf{0}\\ & \mathbf{0} & \mathbf{I}\end{bmatrix}, \tag{4}$$

where $\mathbf{Q}_{t+1}$ is constructed as an orthonormal basis of $(\mathbf{I}-\mathbf{X}_t\mathbf{X}_t^{\top})\mathbf{A}_{t,t+1}$, using e.g., the Gram-Schmidt process [16]. The projection matrix in (4) has also appeared in the context of updating the truncated SVD of matrices in Latent Semantic Indexing [18, 15], as well as the eigen-update of graph Laplacians in spectral clustering applications [14]. In the latter reference, as well as in [15], it is also discussed to set $\mathbf{Z}_{t+1} = \begin{bmatrix}\mathbf{X}_t & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}$, i.e., to ignore the off-diagonal coupling. However, the approximate eigenpairs produced by such a subspace are typically not very accurate.

When $\mathrm{Rank}(\mathbf{A}_t) > k$, the Ritz values produced by projection with the aforementioned $\mathbf{Z}_{t+1}$ will not match those of $\mathbf{A}_{t+1}$ exactly, however a standard application of the Bauer-Fike theorem reveals that the k leading Ritz values are within a $\lambda_{t,k+1}$ distance of the true eigenvalues of $\mathbf{A}_{t+1}$ [19]. Thus, when $\mathbf{A}_t$ is approximately rank-k, the basis in (4) can still provide a good approximation. Nevertheless, graph adjacency matrices are typically sparse and not low-rank.

3.2. Matrix resolvent expansions

In this section we consider an approach based on domain decomposition [20] which does not assume any specific rank for matrix $\mathbf{A}_{t+1}$ and is appropriate even when the dominant eigenvalues of $\mathbf{A}_{t+1}$ are not well-separated. Consider an eigenpair $(\lambda, \mathbf{x})$ of $\mathbf{A}_{t+1}$, partitioned as $\mathbf{x} = [\mathbf{u}^{\top}, \mathbf{y}^{\top}]^{\top}, \mathbf{u} \in \mathbb{R}^{n_t},\ \mathbf{y} \in \mathbb{R}^{s_{t+1}}$. Then, from (1) we have,

$$\begin{bmatrix}\mathbf{A}_t - \lambda\mathbf{I} & \mathbf{A}_{t,t+1}\\ \mathbf{A}_{t,t+1}^{\top} & \mathbf{A}_{t+1,t+1} - \lambda\mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{u}\\ \mathbf{y}\end{bmatrix} = \mathbf{0} \tag{5}$$

from which it follows $\mathbf{u} = (\lambda\mathbf{I} - \mathbf{A}_t)^{-1}\mathbf{A}_{t,t+1}\mathbf{y}$, and thus

$$\mathbf{x} = \begin{bmatrix}\mathbf{u}\\ \mathbf{y}\end{bmatrix} \in \mathrm{Ran}\left(\begin{bmatrix}(\lambda\mathbf{I} - \mathbf{A}_t)^{-1}\mathbf{A}_{t,t+1} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\right).$$

The above expression suggests that we can compute the exact k eigenpairs of $\mathbf{A}_{t+1}$ if $\mathrm{Ran}(\mathbf{Z}_{t+1})$ is equal to

$$\mathrm{Ran}\left(\begin{bmatrix}\mathbf{T}_t(\lambda_{t+1,1}) & \ldots & \mathbf{T}_t(\lambda_{t+1,k}) & \mathbf{0}\\ & \mathbf{0} & & \mathbf{I}\end{bmatrix}\right), \tag{6}$$

where for any $\lambda \notin \{\lambda_{t+1,i}\}_{i=1}^{n_t+s_{t+1}}$,

$$\mathbf{T}_t(\lambda) := (\lambda\mathbf{I} - \mathbf{A}_t)^{-1}\mathbf{A}_{t,t+1}.$$

The range in (6) can be brought into a form similar to (4), by restricting $\mathbf{T}_t(\lambda)$ onto the orthogonal complement of $\mathrm{Ran}\,(\mathbf{X}_t)$, hereby defining the *deflated* matrix-valued function

$$\mathbf{F}_t(\lambda) = (\mathbf{I} - \mathbf{X}_t\mathbf{X}_t^{\top})\mathbf{T}_t(\lambda) = \sum_{j=k+1}^{n_t} \frac{\mathbf{x}_{t,j}(\mathbf{x}_{t,j}^{\top}\mathbf{A}_{t,t+1})}{\lambda - \lambda_{t,j}}. \tag{7}$$

Computing a basis for such a range is generally impractical. However, note that the contribution of each rank-1 term $\mathbf{x}_{t,j}\mathbf{x}_{t,j}^{\top}\mathbf{A}_{t,t+1}$ in (7) is inversely proportional to the difference $\lambda - \lambda_{t,j}$, thus, when $\lambda_{t+1,i} \gg \lambda_{t,j},\ j > k$, $\mathbf{F}_t(\lambda_{t+1,i})$ will not vary much regardless of the index i. Specifically, for $1 \leq i, q \leq k$ the difference

$$\mathbf{F}_t(\lambda_{t+1,i}) - \mathbf{F}_t(\lambda_{t+1,q}) = \sum_{j=k+1}^{n_t} \frac{d_{t+1,i}^{t+1,q}\mathbf{x}_{t,j}(\mathbf{x}_{t,j}^{\top}\mathbf{A}_{t,t+1})}{d_{t,j}^{t+1,i}d_{t,j}^{t+1,q}},$$

where $d_{t,i}^{t+1,q} := \lambda_{t+1,q} - \lambda_{t,i}$, depends on a) the magnitude of $d_{t+1,i}^{t+1,q}$; and b) the distance of $\lambda_{t+1,q}$ and $\lambda_{t+1,i}$ from the largest non-deflated eigenvalue of $\mathbf{A}_t$, $\lambda_{t,k+1}$. Therefore, it appears reasonable to maintain only the leading matrix-valued function $\mathbf{F}_t(\lambda_{t+1,1})$, in which case the projection subspace becomes

$$\mathrm{Ran}\left(\begin{bmatrix}\mathbf{X}_t & \mathbf{F}_t(\lambda_{t+1,1}) & \mathbf{0}\\ & \mathbf{0} & \mathbf{I}\end{bmatrix}\right). \tag{8}$$

To illustrate the advantages of the matrix resolvent-based projection consider the first n_t entries of the eigenvector $\mathbf{x}_{t+1,i} = [\mathbf{u}_{t+1,i}^\top, \mathbf{y}_{t+1,i}^\top]^\top$, and solve for the $n_t \times 1$ subvector $\mathbf{u}_{t+1,i} = \mathbf{T}_t(\lambda_{t+1,i})\mathbf{y}_{t+1,i} \in \mathrm{Ran}\left([\mathbf{X}_t, (\mathbf{I} - \mathbf{X}_t\mathbf{X}_t^\top)\mathbf{T}_t(\lambda_{t+1,i})]\right)$. The subspace in (8) replaces the ideal subspace with the approximation $\mathrm{Ran}\left([\mathbf{X}_t, (\mathbf{I} - \mathbf{X}_t\mathbf{X}_t^\top)(\lambda_{t+1,1}\mathbf{I} - \mathbf{A}_t)^{-1}\mathbf{A}_{t,t+1}]\right)$. On the other hand, the subspace in (2) approximates the ideal subspace by $\mathrm{Ran}\left([\mathbf{X}_t, (\mathbf{I} - \mathbf{X}_t\mathbf{X}_t^\top)\mathbf{A}_{t,t+1}]\right)$. Thus, the approximation produced by (8) approximates $(\lambda_{t+1,i}\mathbf{I} - \mathbf{A}_t)^{-1}$ by $(\lambda_{t+1,1}\mathbf{I} - \mathbf{A}_t)^{-1}$ while (2) replaces $(\lambda_{t+1,i}\mathbf{I} - \mathbf{A}_t)^{-1}$ with the identity matrix.

3.3. Truncation of $\mathbf{F}_t(\lambda_{t+1,1})$ via randomized SVD

Computing an orthonormal basis of the matrix $\mathbf{F}_t(\lambda_{t+1,1})$ requires the solution of s_{t+1} linear systems of the form $(\lambda_{t+1,1}\mathbf{I} - \mathbf{A}_t)y_{t+1,:,1,j} = (\mathbf{I} - \mathbf{X}_t\mathbf{X}_t^\top)\mathbf{A}_{t,t+1}\mathbf{e}_j$, where $y_{t+1,1,j}$ is the j-th entry of $\mathbf{y}_{t+1,1}$, and $\mathbf{e}_j$ denotes the j-th column of the $s_{t+1} \times s_{t+1}$ identity matrix, which may be impractical for large s_{t+1}. Moreover, as the next proposition will show, the matrix $\mathbf{A}_{t,t+1}$ resulting from typical graph updates, and consequently $\mathbf{F}_t(\lambda_{t+1,1})$, might be low-rank.

Proposition 2. *Let $\mathcal{C}_{t+1} = \mathcal{V}_{t+1} \setminus \mathcal{V}_t$ denote the s_{t+1} added vertices at time-step '$t+1$', r_{t+1} denote the number of vertices of $\mathcal{G}_t$ that are connected to $\mathcal{C}_{t+1}$, and g_{t+1} denote the number of vertices of $\mathcal{C}_{t+1}$ that are connected to $\mathcal{G}_t$. Then,* $\mathrm{Rank}(\mathbf{A}_{t,t+1}) \leq \min(r_{t+1}, g_{t+1})$.

Proof. $[\mathbf{A}_{t,t+1}]_{i,j}$ is nonzero if and only if the i-th vertex of $\mathcal{G}_t$ is connected to the j-th vertex of the set $\mathcal{C}_{t+1}$. Thus, $\mathbf{A}_{t,t+1}$ has at most r_{t+1} nonzero rows and g_{t+1} nonzero columns. □

Prop. 2 suggests that the true rank of $\mathbf{A}_{t,t+1}$ can be much smaller than $\min(n_t, s_{t+1})$, when there exist highly repetitive endpoints of the edges between current and newly added vertices, prompting us to seek an orthonormal basis of a low-rank approximation of $\mathbf{F}_t(\lambda_{t+1,1})$. Here, we approximate the $l \in \mathbb{N}$ leading singular triplets of $\mathbf{F}_t(\lambda_{t+1,1})$ via randomized SVD (RSVD) [21]. Given a $s_{t+1} \times (l + p)$ matrix $\mathbf{\Omega}_{t+1}$ of identically and independently distributed entries, where $p \in \mathbb{N}$ is an oversampling parameter, RSVD first computes the matrix product $\mathbf{W}_{t+1} = \mathbf{F}_t(\lambda_{t+1,1})\mathbf{\Omega}_{t+1}$ by solving $l + p$ deflated linear systems with the coefficient matrix $\lambda_{t+1,1}\mathbf{I} - \mathbf{A}_t$. Let $\mathbf{D}_{t+1}$ denote an orthonormal basis of $\mathrm{Ran}(\mathbf{W}_{t+1})$, and consider the singular value decomposition $\mathbf{D}_{t+1}^\top\mathbf{F}_t(\lambda_{t+1,1}) = \widehat{\mathbf{U}}\widehat{\mathbf{\Sigma}}\widehat{\mathbf{V}}^\top$. The rank-$l$ approximation of $\mathbf{F}_t(\lambda_{t+1,1})$ is then equal to $\mathbf{D}_{t+1}\widehat{\mathbf{U}}\widehat{\mathbf{\Sigma}}\widehat{\mathbf{V}}^\top$.

In summary, $\mathbf{Q}_{t+1}$ in (4) is replaced by the orthonormal matrix $\mathbf{R}_{t+1} = \mathbf{D}_{t+1}\widehat{\mathbf{U}}$, leading to the projection matrix

$$\mathbf{Z}_{t+1} = \begin{bmatrix} \mathbf{X}_t & \mathbf{R}_{t+1} & \mathbf{0} \\ & \mathbf{0} & \mathbf{I} \end{bmatrix}. \tag{9}$$

The asymptotic computational complexity of RSVD is equal to the sum of the following terms: *a)* $O(\mathrm{nnz}(\mathbf{A}_t)(l+p))$ (to form $\mathbf{W}_{t+1}$), *b)* $O(n_t(l+p)^2)$ (to compute $\mathbf{D}_{t+1}$), and *c)* $O(n_t l(l+p))$ (to form $\mathbf{R}_{t+1}$).

The basis in (9) can be further truncated by replacing the lower-right $s_{t+1} \times s_{t+1}$ identity matrix with the matrix $\mathbf{G}_{t+1}$ formed by the ℓ leading eigenvectors of $\mathbf{A}_{t+1,t+1}$. The performance of this approach will be evaluated in the next section.

Table 1. *Summary of options to set $\mathbf{Z}_{t+1}$ and solve the Rayleigh-Ritz eigenvalue problem.*

Option	$\mathbf{Z}_{t+1}$	RR cost
M1 [15, 14]	$\begin{bmatrix} \mathbf{X}_t & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$	$\mathcal{O}((k + s_{t+1})^3)$
M2 [18, 14]	$\begin{bmatrix} \mathbf{X}_t & \mathbf{Q}_{t+1} & \mathbf{0} \\ & \mathbf{0} & \mathbf{I} \end{bmatrix}$	$\mathcal{O}((k + 2s_{t+1})^3)$
M3	$\begin{bmatrix} \mathbf{X}_t & \mathbf{R}_{t+1} & \mathbf{0} \\ & \mathbf{0} & \mathbf{I} \end{bmatrix}$	$\mathcal{O}((k + l + s_{t+1})^3)$
M4	$\begin{bmatrix} \mathbf{X}_t & \mathbf{R}_{t+1} & \mathbf{0} \\ & \mathbf{0} & \mathbf{G}_{t+1} \end{bmatrix}$	$\mathcal{O}((k + l + \ell)^3)$

4. NUMERICAL EXPERIMENTS

Our goal in this section is to: *a)* validate the accuracy of the approximate k leading eigenpairs returned by Algorithm 1 after all graph updates are complete, and *b)* benchmark the qualitative accuracy achieved by the embeddings produced by Algorithm 1 in applications such as exponential and subgraph centrality.

All tests are conducted in MATLAB [22]. Table 1 summarizes four different options to set $\mathbf{Z}_{t+1}$, and their respective asymptotic complexity required to solve the Rayleigh-Ritz eigenvalue problem. Options **M1** and **M2** have been used in [18, 14, 15] to update eigenembeddings of dynamic datasets; option **M2** is outlined in Sec. 3.1. Option **M3** refers to the method introduced in this work, outlined in Secs. 3.2 and 3.3, and option **M4** refers to the same method but with its basis further truncated, using the leading ℓ eigenvectors of $\mathbf{A}_{t+1,t+1}$, as explained at the end of Sec. 3.3. Note that the asymptotic complexity of computing the embeddings at each time step is $\mathcal{O}(n_t^2 k)$. Throughout all experiments we set the target rank l and the oversampling parameter p of RSVD all equal to k. Four graphs from the University of Florida Sparse Matrix collection [23] are considered in this paper: *a)* `wing_nodal`, a Walshaw graph, with $n = 10,937$ vertices and $m = 150,976$ edges, *b)* `socfb-MIT`, an MIT social friendship Facebook network, with $n = 6,400$ vertices and $m = 251,200$ edges, *c)* `ego-facebook`, a dataset modelling 'circles' (or 'friends lists') from a Facebook users' survey, with $n = 4,039$ vertices and $m = 88,234$ edges, and *d)* `USpowerGrid`, a graph modelling the topology of the Western States Power Grid of the US, with $n = 4,941$ vertices and $m = 13,188$ edges.

To emulate dynamic graphs, we partition the vertices of each graph randomly into $T = 20$ subgraphs of similar sizes. Starting from a randomly selected initial subgraph, at each time-step the vertices and edges associated with one of the remaining subgraphs are added, until the entire graph is parsed. Fig. 1 plots the relative eigenvalue errors $|\lambda_{T,i} - \lambda_i|, i = 1, \ldots, k$ and corresponding residual norms $\|(\mathbf{A} - \lambda_{T,i}\mathbf{I})x_{T,i}\|_2$ of the approximate eigenpairs produced by Alg. 1 for the graphs considered, with $\mathbf{A}$ denoting the adjacency matrix for the full graph and λ_i denoting its i-th eigenvalue. Options '**M2**' and '**M3**' are the most accurate, followed by '**M4**' and '**M1**', which, however, are the options with the fewest columns in $\mathbf{Z}_{t+1}$. Moreover, from Table 1, the asymptotic complexity of the Rayleigh-Ritz projection in '**M4**' does not depend on the graph update size s_{t+1}. Thus, for large graph updates, '**M4**' might be the fastest approach.

Finally, we consider a graph top-N vertex recommendation task based on (exponential) subgraph centrality [3]. Recommendation is facilitated by identifying the vertices corresponding to the $N \in \mathbb{N}$

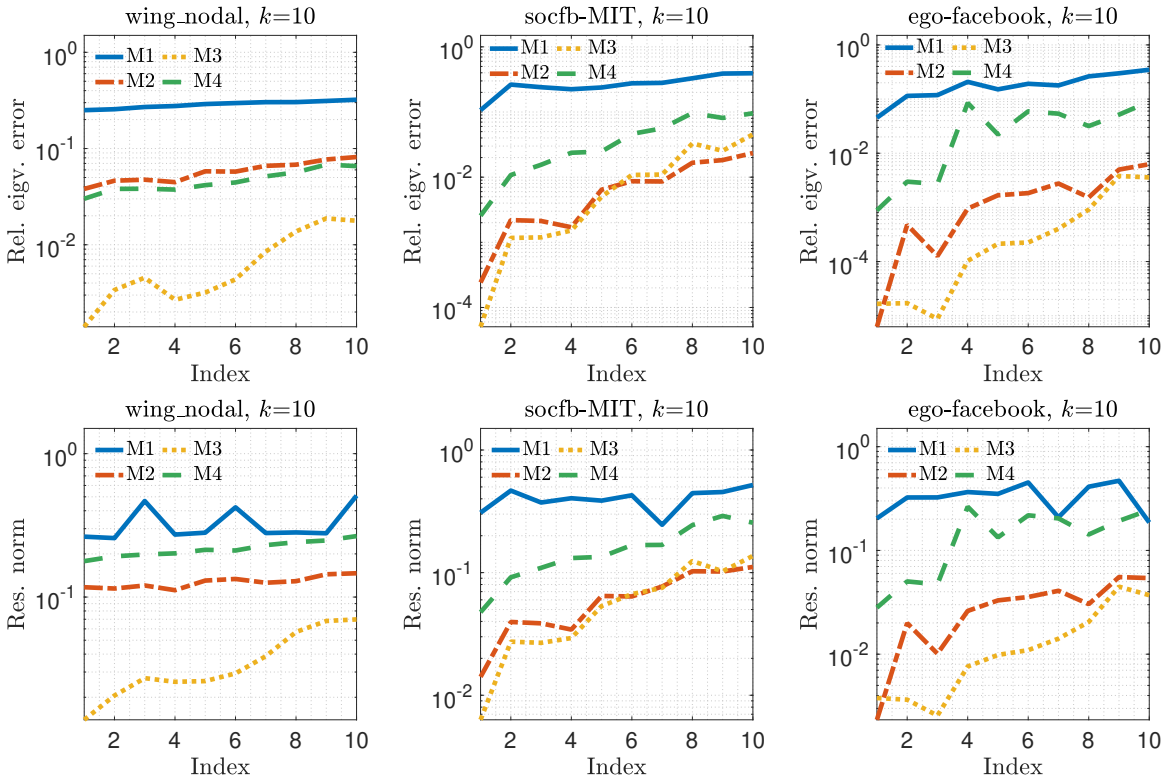


Fig. 1. Top row: Relative Eigenvalue error. Bottom row: Residual norm. *Index* denotes the eigenvalue/eigenvector index.

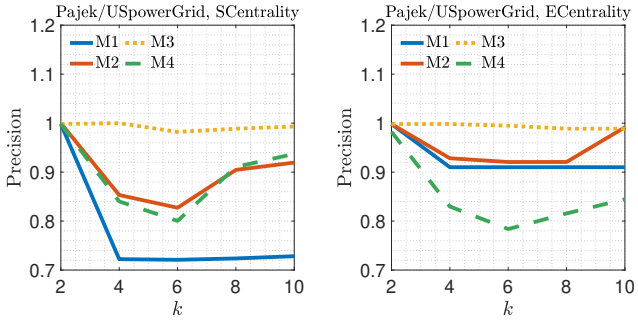


Fig. 2. Left: Subgraph centrality. Right: Exponential subgraph centrality for *USPowerGrid*

largest indices of a) the diagonal of $e^{\mathbf{A}}$ (subgraph centrality), and b) the product $e^{\mathbf{A}}\mathbf{1}$, where $\mathbf{1}$ denotes the vector of all ones (exponential subgraph centrality). In both cases, Alg. 1 is used to approximate $e^{\mathbf{A}}$ by $\mathbf{X}_t e^{\mathbf{\Lambda}_t}\mathbf{X}_t^\top$. Figure 2 plots the 11-point average interpolated precision [24] of the top-N graph recommendation achieved when $k = \{2, 4, 6, 8, 10\}$ and $N = \lfloor n/100 \rfloor$, for the graph `USpowerGrid`. In both cases, the '**M3**' resolvent-based approach led to higher average precision.

5. CONCLUSION

This contribution presented a Rayleigh-Ritz framework to update eigenembeddings of evolving graphs. Several options to update the projection subspace were presented, including a novel scheme based on matrix resolvent expansions which does not impose any constraints on the rank of the adjacency matrix and exhibits improved accuracy compared to previous approaches.

Future work includes the study of the proposed framework for other graph analytic tasks such as spectral graph clustering or triangle counting. Another task of interest is the update of the initial spectral embeddings used in Graph Neural Networks subject to graph updates. Future work will also feature vertex and edge deletions, which occur in many real-world applications such as social network analysis.

6. REFERENCES

[1] U. Kang, S. Papadimitriou, J. Sun, and H. Tong, "Centralities in large networks: Algorithms and observations," in *Proceedings of the 2011 SIAM international conference on data mining*, 2011, pp. 119–130.

[2] D. Koschützki, K. A. Lehmann, L. Peeters, S. Richter, D. Tenfelde-Podehl, and O. Zlotowski, "Centrality indices," in *Network analysis*, pp. 16–61. Springer, 2005.

[3] M. Benzi, E. Estrada, and C. Klymko, "Ranking hubs and authorities using matrix functions," *Linear Algebra and its Applications*, vol. 438, no. 5, pp. 2447–2474, 2013.

[4] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[5] B. Baingana, P. A. Traganitis, G. B. Giannakis, and G. Mateos, "Big data analytics for social networks," *Graph-Based Social Media Analysis*, vol. 39, pp. 293, 2016.

[6] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web.," Tech. Rep., Stanford InfoLab, 1999.

[7] E. Estrada and J. A. Rodriguez-Velazquez, "Subgraph centrality in complex networks," *Physical Review E*, vol. 71, no. 5, pp. 056103, 2005.

[8] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.

[9] C. Chen and H. Tong, "Fast eigen-functions tracking on dynamic graphs," in *Proceedings of the 2015 SIAM international conference on data mining*. SIAM, 2015, pp. 559–567.

[10] J. Li, H. Dani, X. Hu, J. Tang, Y. Chang, and H. Liu, "Attributed network embedding for learning in a dynamic environment," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, New York, NY, USA, 2017, p. 387–396.

[11] D. Zhu, P. Cui, Z. Zhang, J. Pei, and W. Zhu, "High-order proximity preserved embedding for dynamic networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 11, pp. 2134–2144, 2018.

[12] L. Martin, A. Loukas, and P. Vandergheynst, "Fast approximate spectral clustering for dynamic networks," in *Proceedings of the 35th International Conference on Machine Learning*, J. Dy and A. Krause, Eds., 10–15 Jul 2018, vol. 80 of *Proceedings of Machine Learning Research*, pp. 3423–3432.

[13] H. Ning, W. Xu, Y. Chi, Y. Gong, and T. S. Huang, "Incremental spectral clustering by efficiently updating the eigensystem," *Pattern Recognition*, vol. 43, no. 1, pp. 113–127, 2010.

[14] C. Dhanjal, R. Gaudel, and S. Clémençon, "Efficient eigenupdating for spectral graph clustering," *Neurocomputing*, vol. 131, pp. 440–452, 2014.

[15] V. Kalantzis, G. Kollias, S. Ubaru, A. N. Nikolakopoulos, L. Horesh, and K. Clarkson, "Projection techniques to update the truncated SVD of evolving matrices with applications," in *International Conference on Machine Learning*, 2021, pp. 5236–5246.

[16] G. H. Golub and C. F. Van Loan, *Matrix computations*, JHU press, 2013.

[17] B. N. Parlett, *The symmetric eigenvalue problem*, SIAM, 1998.

[18] H. Zha and H. D. Simon, "On updating problems in latent semantic indexing," *SIAM Journal on Scientific Computing*, vol. 21, no. 2, pp. 782–791, 1999.

[19] Y. Saad, *Numerical methods for large eigenvalue problems: revised edition*, SIAM, 2011.

[20] V. Kalantzis, *Domain decomposition algorithms for the solution of sparse symmetric generalized eigenvalue problems*, Ph.D. thesis, University of Minnesota, 2018.

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

[22] MATLAB, *version 9.11.0 (R2021b)*, The MathWorks Inc., Natick, Massachusetts, 2021.

[23] T. A. Davis and Y. Hu, "The University of Florida sparse matrix collection," *ACM Trans. Math. Softw.*, vol. 38, no. 1, Dec 2011.

[24] T. G. Kolda and D. P. O'leary, "A semidiscrete matrix decomposition for latent semantic indexing information retrieval," *ACM Transactions on Information Systems (TOIS)*, vol. 16, no. 4, pp. 322–346, 1998.