Network construction: A learning framework through localizing principal eigenvector

Priodyuti Pradhan\textsuperscript{1} and Sarika Jalan\textsuperscript{2}

\textsuperscript{1} Department of Mathematics, Bar-Ilan University, Ramat Gan 5290002, Israel
\textsuperscript{2} Complex Systems Lab, Department of Physics, Indian Institute of Technology Indore, Khandwa Road, Simrol, Indore-453552, India

Abstract. Information of localization properties of eigenvectors of the complex network has applicability in many different areas which include networks centrality measures, spectral partitioning, development of approximation algorithms, and disease spreading phenomenon. For linear dynamical process localization of principal eigenvector (PEV) of adjacency matrices infers condensation of the information in the smaller section of the network. For a network, an eigenvector is said to be localized when most of its components are near to zero with few taking very high values. Here, we provide three different random-sampling-based algorithms which, by using the edge rewiring method, can evolve a random network having a delocalized PEV to a network having a highly localized PEV. In other words, we develop a learning framework to explore the localization of PEV through a random sampling-based optimization method. We discuss the drawbacks and advantages of these algorithms. Additionally, we show that the construction of such networks corresponding to the highly localized PEV is a non-convex optimization problem when the objective function is the inverse participation ratio. This framework is also relevant to construct a network structure for other lower-order eigenvectors.

1 Introduction

Networks furnish a mathematical framework to model and decipher the collective behavior of complex real-world systems. Scrutiny of principal eigenvector (PEV) and the corresponding eigenvalue of the networks are known to provide an understanding of various local and global structural properties as well as the time evolution of dynamics on those networks \cite{1}, \cite{2}, \cite{3}, \cite{4}, \cite{5}, \cite{6}. Further, different networks’ eigenvector-based centrality measures have been proposed to understand the importance of the nodes forming those networks. For example, eigenvector centrality or Katz centrality provides a ranking to nodes of networks based on the entries of the PEV \cite{7}, \cite{8}. Similarly, the PageRank algorithm which is based on the PEV of Google matrices predicts the importance of the web pages \cite{8}. Also, variants of principal component analysis and independent component analysis have led to radical developments in machine learning approaches \cite{9}. Furthermore, conditions under which the degree vector of a network and PEV
are correlated have been derived leading to the usage of degree vector instead of PEV to approximate various network analysis results \[10\]. Furthermore, a network construction for which the message passing equations are exact have been explained and solutions near the critical point in terms of the PEV components have been analyzed \[11\]. Recently, sensitivity in the network dynamics has been explored using networks' eigenvectors \[12\]. Particularly, development of community detection techniques based on the localization properties of eigenvector \[13\], \[14\], \[15\] have been another significant contribution to the complex network analysis. Further, localization has an important role in quantum physics \[16\], mathematics \[17\], \[18\], \[19\], approximate algorithm development \[20\], machine learning \[21\], numerical linear algebra, matrix gluing, structural engineering, computational quantum chemistry \[22\], \[23\], and in quantum information theory \[24\]. From real-world systems, it is also evident that underlying interaction patterns form the infrastructure for their different emerging dynamical responses. For instance, information or rumor propagates through the Facebook-Twitter networks; in brain, the neurons interact to perform specific functions over the underlying network. Moreover, reconfiguration or rewiring of the functional brain networks are required during the learning phases \[25\]. Therefore a scrutiny of the network architecture is important as ‘underlying structure has crucial impact on its function’ and vice-versa \[26\].

Due to the versatile applications of the eigenvector properties, we analyze the network architecture which optimizes a specific behavior of its PEV. In this article, we study the network structure from a different point of view. Instead of analyzing the properties of a network, we construct a sequence of networks, \(\{G_1, G_2, \ldots, G_{\text{final}}\}\) optimizing a few specific behaviors of the PEV. The primary aim of this framework is to examine the sequence of networks and learn the network properties collectively when optimizing a function \(\zeta: \mathbb{R}^n \rightarrow \mathbb{R}\) on the eigenvectors. In other words, we can represent an undirected network by an adjacency matrix which encodes the interactions or relations among \(n\) objects (nodes) of a real-world complex system. We consider that the adjacency matrix is symmetric, and hence from the eigenvalue equation, we have \(n\) number of eigenvectors which represent \(n\) different solutions of the system. Each eigenvector has a different meaning corresponding to the underlying system. We have an interest in network architecture that satisfies few particular behaviors of the eigenvector (solution). The entry of an eigenvector for a symmetric matrix may contain negative, zero, or positive values. Can we tune the eigenvector entries and construct the network structure accordingly? The question one can ask that how values of the eigenvector and the corresponding network structure are related. Reversely, to get a particular behavior for the eigenvector what will be the interaction matrix which in our case is the network’s adjacency matrix. The tuning can be performed based on a particular function \(\zeta\), and for our purpose here it is the inverse participation ratio (IPR) of the PEV (Eq. \(2\)).

It has been demonstrated that few structural features of networks, such as the presence of a hub node, the existence of dense subgraphs, or a power-law
degree distribution may lead to the localization of the PEV \[32, 33\]. However, the questions which still need investigation and dealt with in this article are:

a. Starting with a network having a delocalized PEV, how can one gradually localize the PEV behavior and construct the corresponding network structures?

b. What is the particular architecture of the optimized network corresponding to a highly localized PEV?

Here, we develop a general framework to evolve a network structure based on rewiring its edges as the PEV of the corresponding adjacency matrix goes from the delocalized to the highly localized state. We devise three different algorithms which use random edge sampling (Hub-based, Monte-Carlo-based, and simulated annealing based) to find out the final network structure. Moreover, we show that the optimized network concerning the highly localized PEV has a distinctive architecture.

The article is organized as follows: Section 2 motivates to study PEV localization. Section 3, contains the notations and definitions used in the later discussion. Also, it includes a brief explanation and formulation of our work. Section 4, illustrates various algorithms on edge rewiring-based optimization in detail. Finally, section 5, summarizes the current study and discusses the open problems for further investigations.

2 Motivation

We know relations or interactions are everywhere either interactions of power grid generators to provide proper functioning of the power supply over a country, or interactions of bio-molecules inside cell to proper functioning of cellular activity or interactions of neurons inside brains to perform specific functions or interactions among satellites to provide accurate GPS services or interactions among quantum particle to form quantum communications or recent coronavirus spread \[26, 25, 27, 28, 29\]. For all of them there are two things that are common — network structure and interactions mechanism among the components or dynamics running on top of that. For instance, for a linear dynamical model (Fig. 1)

\[
x(t + 1) = M^{t+1}x(0) \xrightarrow{t \to \infty} x^* \sim v_1^M
\]

where \(M = aI + bA\) is the transition matrix, \(x(0)\) is the initial state of the system, \(x^*\) is the stationary state and \(v_1^M\) is the PEV of \(M\). To find the pattern or understand the behavior of the linear dynamics on networks, we iterate the matrix multiplication process for a large number of times or we can find out the PEV (\(v_1^M\)) of \(M\) (Eq. 1). There is another way, we can also solve the problem. We can see that eigenvectors of \(M\) are the same as eigenvectors of \(A\) [1].

\[
Mu_i^A = aIv_i^A + bAv_i^A = [a + b\lambda_i^A]v_i^A = \lambda_i^Mv_i^A
\]

where \(I\) is the identity matrix and \(\lambda_i^M = a + b\lambda_i^A\) and. Thus, \(x^* \sim v_1^M = v_1^A\). Hence, understanding the behavior of the PEV of the adjacency matrix is
Fig. 1. A linear dynamical model. Here, $x_i$ captures some information on node $i$ and $A_{ij}$'s are the structure of the network and $a$ and $b$ are the model parameters. We seek to understand the long term behavior of the information exchange pattern.

enough to understand the information flow pattern for linear dynamical systems. Therefore, the network properties which enhance PEV localization of adjacency matrices can implicitly restrict the linear-dynamics in a smaller section of the network. A fundamental question at the core of the structural-dynamical relation is that: How can we find network structure having highly localized PEV? Here, we provide random-sampling based algorithm to construct localized network structure.

3 Problem formulation

We represent a graph, $\mathcal{G} = \{V, E\}$, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of vertices and $E = \{e_1, e_2, \ldots, e_m|e_m = (v_i, v_j)\} \subseteq U$ is the set of edges. We define the universal set $U = V \times V = \{(v_i, v_j)|v_i, v_j \in V \text{ and } i \neq j\}$ which contains all possible ordered pairs of vertices excluding the self-loops and the complementary set can be defined as $E^c = U - E = \{(v_i, v_j)|(v_i, v_j) \in U \text{ and } (v_i, v_j) \notin E\}$ i.e., $E \cap E^c = \emptyset$ and $E \cup E^c = U$. We denote the adjacency matrices corresponding to $\mathcal{G}$ as $A \in \mathbb{R}^{n \times n}$ and which can be defined as $(a)_{ij} = 1$, if $v_i \sim v_j$ and 0 otherwise. The degree of a node can be represented as $d(v_i) = \sum_{j=1}^{n} a_{ij}$, and the average degree of $G$ can be defined as $\langle k \rangle = \frac{1}{n} \sum_{i=1}^{n} d(v_i)$. Here, we consider $|V| = n$, $|E| = m$, and $|E^c| = \frac{n(n-1)}{2} - m$. The spectrum of $\mathcal{G}$ is the set of the eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ of $A$. Without loss of generality we can order the eigenvalues of $A$ as $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ and corresponding eigenvectors as $x_1, x_2, \cdots, x_n$ respectively. Here, $A$ is a real symmetric matrix, and each has real eigenvalues. In addition, the networks are connected. Hence, we know from the Perron-Frobenius theorem \[30\] that all the entries in the PEV ($x_1$) of $A$ are positive. We quantify localization of the PEV through the inverse participation
ratio, which is the sum of fourth power of the eigenvector entries and calculate \[33\], \[3\] as follows:

\[ Y_{x_1} = \sum_{i=1}^{n} (x_1)_i^4 \]  

(2)

where \((x_1)_i\) is the \(i\)th component of \(x_1\) and \(||x_1||^2 = 1\). A delocalized eigenvector with component \((\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}, \ldots, \frac{1}{\sqrt{n}})\) has \(Y_{x_1} = \frac{1}{n}\), whereas the most localized eigenvector with components \((1, 0, \ldots, 0)\) yields an IPR value equal to \(Y_{x_1} = 1\). A network is said to be regular if each node has the same degree \[30\]. It also turns out that for any regular graph (Theorem 6 \[30\]), we get PEV, \(x_1 = (\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}, \ldots, \frac{1}{\sqrt{n}})\). Hence, \(Y_{x_1} = \frac{1}{n}\), corresponds to the most delocalized PEV. Therefore, for any regular network IPR value of the PEV provides the lower bound. Hence, a sparse as well as a dense regular network contains delocalized PEV. Now, we can consider a disconnected graphs where each node is isolated from each other and each node has a self-loop. The adjacency matrix can be represented with the \(n \times n\) identity matrix. For this disconnected networks, \(Y_{x_1} = 1\). In another situation if we consider only \(n\) number of isolated nodes with \(|E| = 0\). We have a zero matrix and for which we can choose \(x_1 = (1, 0, \ldots, 0)\) and \(Y_{x_1} = 1\). These are the special cases. Additionally, for any disconnected network with less than \(n\) number of components, the PEV entries might be zeros. Hence, for a connected network, IPR value lies between \(1/n \leq Y_{x_1} < 1\). Therefore, it is evident that finding out a network architecture for a given \(n\) with delocalized PEV is easier than searching for a connected network structure with highly localized PEV. For a given \(n\) and \(m\), our aim is to get a connected network which has the most localized PEV. In other words, we can state the problem as, we search for a binary symmetric matrix \(A_{final}\) which is irreducible and which has the PEV with maximum IPR value. Also we have an interest to know the properties of sequence of the adjacency matrices \(\{A_1, A_2, \ldots, A_{final}\}\) during the searching process which can maximize the IPR value. The search space for a given \(n\) and \(m\) is of the order \(O(n^2m)\) \[31\]. Therefore, we use random-sampling to formulate this problem.

4 Methodology and Results

We use randomized algorithms based on edge rewiring to construct a network architecture that corresponds to a highly localized PEV in an iterative manner. It is very natural that modification in the entries of an adjacency matrix leads to a change in the spectral properties (eigenvalues and eigenvectors) which also simultaneously change the network architecture. We use this fact to develop randomized algorithms. The modification in the adjacency matrix can be performed by removing or adding edges as well as nodes or rearrangements of the edges in \(G\) \[3\]. Here, we devise algorithms by rearrangement of the edges to get highly localized PEV when \(G\) remains connected, and the number of nodes and edges remain fixed. In the following, we discuss the algorithms in detail.
4.1 Hub-based algorithm

It is well known that networks with localized PEV have a hub node [32]. Here, we attempt to connect PEV localization and the corresponding network structure. We use the presence of a hub node heuristic to develop a randomized algorithm. It iteratively forms a hub node starting from an Erdős-Rényi (ER) random network and records the IPR value as well as stores the sequence of networks \( \{G_{ER}, G_2, \ldots, G_{final}\} \). Starting the algorithm, with an ER random network is an artifact as it provides the delocalized PEV [37]. The initial ER random network, \( G(n, p) \) is generated with an edge probability \( p = \langle k \rangle / n \). Without loss of generality, we consider \( v_1 \) will be the hub node at the end of the iterative process. We select an edge \( e_r \in V - \{v_1\} \times V - \{v_1\} \) uniformly at random from \( G \), and remove it. Simultaneously, add it between \( v_1 \) to \( v_k \), if \( (v_1, v_k) \notin E \). We repeat the process until \( v_1 \) connects to all the remaining nodes and becomes the hub node (algorithm 1). At the end of the iterative process we get \( d(v_1) = n - 1 \) for \( G_{final} \). This random iterative hub formation algorithm keeps unchanged the network size. We can see the IPR value during the evolution from Fig. 2(a). Interestingly, it shows the changes in the IPR value for the sequence of networks \( \{G_{ER}, G_2, \ldots, G_{final}\} \) collectively. Moreover, we depict the sorted PEV entry values in Fig. 2(d), which indicates the magnitude of the maximum PEV entry value is much larger than the rest of the entry values. The question will arise whether the IPR value of \( G_{final} \) is close to the optimal. We use some results from previous research on the upper bound on the maximal entry value of the PEV for a connected network.
Fig. 3. Single edge rewiring. Select an edge randomly between a node pair \((p,q)\) and remove it. Add the edge between a randomly selected node pair \((r,s)\).

to make a possible conclusion about the optimality of our results. The maximal PEV entry value can be obtained for the star network and it is \(1/\sqrt{2}\) and all other entries are same which are \(1/2\sqrt{2(1-n)}\) \[38\]. The maximum PEV entry value obtains from the \(G_{final}\) is close to \(1/\sqrt{2}\). This simple hub node formation-based algorithm works well and is easier to implement to get networks with highly localized PEV. Later on, we devise other algorithms which provide better results than algorithm[I] \(1\) We use the C++ language and STL library to implement all the algorithms. To find out the eigenvector, we bind the LAPACK routine \texttt{ssyevr}

\begin{algorithm}
\begin{algorithmic}[1]
\State \(A \leftarrow \mathcal{G}(n,p)\)
\State \(Y_{x_1} \leftarrow \zeta(x_1)\)
\While {\(k = 2 \text{ to } n\)}
\State choose an edge \((v_i, v_j) \in V - \{v_1\} \times V - \{v_1\}\) uniformly at random
\State \If {\((v_1, v_k) \notin E\)}
\State remove \((v_i, v_j)\) and add as \((v_1, v_k)\) in \(G\) and denotes it as \(G'\)
\EndIf
\If {\(G'\) is not connected} then goto step 4
\State \(A \leftarrow A'\)
\State \(Y_{x_1} \leftarrow \zeta(x_1)\)
\EndWhile
\State store \(Y_{x_1}\) and \(A\)
\end{algorithmic}
\end{algorithm}
with C++ code.

4.2 Monte-Carlo based algorithm

In the previous algorithm, we select an edge at random from the set \(V - \{v_1\} \times V - \{v_1\}\) and always add it to the node \(v_1\). However, if we make the position of the edge removal and addition (or we can say edge rewiring) more flexible, and accept those edge rewirings which can improve the IPR value, we get an...
Fig. 4. (a) Intermediate network structure during optimization (b) Optimized structure obtained at the end of the iteration process with $Y_{x_1} \approx 0.20$; $n = 500$ nodes and $\langle k \rangle = 10$ edges.

impressive result. In particular, we achieve significant improvement as well as in the network structure than the Hub-based algorithm. We formulate an optimization problem as: given a connected network $\mathcal{G}$ with $n$ vertices, $m$ edges and a function $\zeta : \mathbb{R}^n \rightarrow \mathbb{R}$, we want to compute the maximum possible value of an objective function $\zeta(x_1) = \sum_{i=1}^{n}(x_1)_i^4$ over all the simple, connected, and undirected network $\mathcal{G}$. The optimization problem can be written as finding an irreducible binary symmetric matrix $A$, for which $\sum_{i=1}^{n}(x_1)_i^4$ will be maximum such that $Ax_1 = \lambda_1 x_1$, $||x_1||_2 = 1$ and $(x_1)_i > 0$, $i \in \{1, 2, \ldots, n\}$ (detail about the objective function in Appendix). The first constraint simply says that $x_1$ is the PEV of a symmetric matrix $A$ and it is in $l_2$ norm. The second constraint implicitly stipulates that the network must be connected (from the Perron-Frobenius theorem). We refer the initial network as $\mathcal{G}_{\text{init}}$ and the optimized network as $\mathcal{G}_{\text{final}}$. The network evolution emerges sequence of networks as $\{\mathcal{G}_{\text{init}}, \mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_{\text{final}}\}$.

For a single-edge rewiring, we choose an edge $e_i \in E$ uniformly at random from $\mathcal{G}$ and remove it (Fig. 3). At the same time, we introduce an edge in the $\mathcal{G}$ from $E^c$, which preserves the total number of edges during the network evolution in $\mathcal{G}$. Hence, each edge rewiring is a two-step process, (i) removal of an edge followed by (ii) addition of an edge (Fig. 3). We remark that during the network evolution there is a possibility that an edge rewiring disconnects the network. To avoid this situation, we only approve those rewirings which yield the network connected. To check the connectedness after an edge rewiring, we use a depth-first search (DFS) algorithm [39].

The Monte-Carlo (MC) based optimization (in algorithm 2) can be summarized as follows. We find $x_1$ of an ER random graph $\mathcal{G}$ and calculate the IPR value of $x_1$. We rewire one edge uniformly and independently at random in $\mathcal{G}$ to obtain another graph $\mathcal{G}'$. We check whether $\mathcal{G}'$ is connected, if not the edge rewiring step is repeated till we get another $\mathcal{G}'$ which is a connected network. We
Algorithm 2: MC-based-IPR-Optimization(n, ⟨k⟩)

1. \( A \leftarrow \mathcal{G}(n, p) \)
2. \( Y_{x_1} \leftarrow \zeta(x_1) \)
3. while \( Y_{x_1} \) not saturated do
   4. rewire an edge uniformly at random in \( \mathcal{G} \) and denotes it as \( \mathcal{G}' \)
   5. if \( \mathcal{G}' \) is not connected then goto step 4
   6. \( Y_{x_1}' \leftarrow \zeta(x_1) \)
   7. if \( Y_{x_1}' > Y_{x_1} \) then
      8. \( A \leftarrow A' \)
      9. \( Y_{x_1} \leftarrow Y_{x_1}' \)
   10. end
11. store \( Y_{x_1} \) and \( A' \)
12. end

Fig. 5. Changes of IPR value when initial network as (a) star graph (b) path graph, where \( n = 500 \) nodes.

find out the PEV of \( A' \) matrix and calculate the IPR value of \( x_1' \). We replace \( A \) with \( A' \), if \( Y_{x_1'} > Y_{x_1} \). Steps are repeated until the IPR value gets saturated which corresponds to the optimized network. The recorded value of \( Y_{x_1} \) variable during the optimization process gives an increment in the IPR value which is depicted in Fig 2(b). We depict a network at an intermediate evolution stage and the final optimized one in Fig. 4. It indicates that the optimized network structure contains two graph components connected via a single node as we have seen in [3]. The sorted PEV entries obtained from the optimized network in Fig. 2(e) portrays that we are very close to the optimal IPR value. However, there is a difference in the eigenvector entries from Fig. 2(d) and the \( \mathcal{G}_{opt} \) obtain from MC based algorithm has maximum degree, \( d_{max} \ll n - 1 \). It indicates that optimal IPR value depends on the particular entry value behavior of PEV. Here, we consider ER random network as initial. Now, if we change the initial network instead of ER random network, then there is a chance of failure to the MC method. Interestingly, we have found out one such situation and discussed it in the following using the simulated annealing-based method.
Algorithm 3: SA-based-IPR-Optimization($n$, $(k)$, $temp$, $\kappa$)

1. $A \leftarrow G(n, p)$
2. $Y_{x_1} \leftarrow \zeta(x_1)$
3. $temp = 0.9$
4. while $Y_{x_1}$ not saturated do
5.   rewire an edge uniformly at random in $G$ and denotes it as $G'$
6.   if $G'$ is not connected then goto step 5
7.   $Y'_{x_1} \leftarrow \zeta(x_1)$
8.   if $Y'_{x_1} > Y_{x_1}$ then
9.     $A \leftarrow A'$
10.    $Y_{x_1} \leftarrow Y'_{x_1}$
11.   end
12. else
13.     pick a random number $r$ from uniform distribution in $(0,1)$
14.     if $r < e^{(Y'_{x_1} - Y_{x_1})/(\kappa \cdot temp)}$ then
15.         $A \leftarrow A'$
16.         $Y_{x_1} \leftarrow Y'_{x_1}$
17.     end
18. end
19. store $Y_{x_1}$ and $A'$
20. $temp \leftarrow temp \ast 0.998$
21. end

4.3 Simulated annealing based algorithm

The simulated annealing (SA) is a randomized algorithm widely used in solving optimization problems motivated by the principles of statistical mechanics [40]. The important part of the SA-based algorithm is accepting solutions that satisfy the Gibbs-Boltzmann function $e^{-E/\kappa \ast temp}$. In our problem, we consider the objective function to maximize instead of minimizing, so we have made the changes accordingly in the algorithm. We set the initial temperature, $temp = 0.9$ and after each iteration decreases it by the cooling schedule $tem = temp \ast 0.98$ and also fix the Boltzmann constant $\kappa$ to 100.

If we consider a star network with $n$ nodes labelled as $\{1, 2, \ldots, n\}$ with the hub node being labelled with 1, then using eigenvalue equation corresponding to $\lambda_1$, we get the PEV,

$$x_1 = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2(n-1)}}, \ldots, \frac{1}{\sqrt{2(n-1)}}\right)$$

and $Y_{x_1}^s = \frac{1}{4} + \frac{1}{4(n-1)}$

Therefore, when $n \rightarrow \infty$, we get $Y_{x_1} \rightarrow \frac{1}{4} \approx 0.25$ for the star network. By looking the PEV entries and the IPR value, it might happen that star network has the most localized PEV, but it is not true. It becomes clear if we provide a star

...
Fig. 6. Optimized structure obtained through the SA algorithm when the initial structure is the star network. The $Y_{x_1} \approx 0.267$, $n = 500$ nodes and $m = 499$ edges.

Fig. 7. MC algorithm with (a) initial star network (b) rewired network configuration.

structure as an initial network to the MC-based algorithm. We do not show any increment in the IPR value, and it sticks to the local maxima (Fig. 5(a)). We explain the reason behind the failure in the following. After removing an edge connected to the hub node in the star network (Fig. 7(a)), it must be connected to any peripheral node (Fig. 7(b)). From the adjacency matrix of the rewired network structure (in Fig. 7(b)) we solve the eigenvalue equation and find out,

$$x_1 = \left( (x_1)_1, \frac{1}{\lambda_1} (x_1)_1, \ldots, \frac{1}{\lambda_1} (x_1)_1, \frac{\lambda_1}{\lambda_1^2 - 1} (x_1)_1, \frac{1}{\lambda_1^2 - 1} (x_1)_1 \right)$$

and

$$Y_{x_1}^c = (x_1)_1^4 \left( 1 + \frac{n - 3}{\lambda_1^4 + 1} \right)$$

where $(x_1)_1^2 = \frac{\lambda_1^4 - \lambda_2^2}{2\lambda_1^4 - (n-3)\lambda_1^2 - (n-3)}$ and $\lambda_2^2 = \frac{(n-1) + \sqrt{(n-1)^2 - 4(n-3)}}{2}$. We observe numerically that for varying network size up to $n = 2000$, $Y_{x_1}^c < Y_{x_1}^s$ and after that $Y_{x_1}^c \approx Y_{x_1}^s$. Hence, MC based algorithm does not accept the configuration in Fig. 7(b) and iterates forever without enhancement to the IPR value. However, in case of SA based algorithm, initially, when the temp is high then from uniform distribution the algorithm accepts the configuration as in Fig. 7(b). It leads to a better optimal value and gives an optimized structure which is different from.
the star network (Fig. 6). Moreover, giving a path network as an initial network to both the MC and SA based method gives an optimized structure as in Fig. 6 and get an improvement in the IPR value (Fig. 6b). It indicates that success of the MC algorithm depends on the choice of the initial network. Furthermore, from the numerical simulations, we have learned that when the number of edges $|E| >> n - 1$ then the MC and SA based algorithm works well to find out an optimized network structure.

5 Conclusion

We explore different network construction algorithms which optimizing the behavior of PEV. We construct the network structure through the optimization process that possesses highly localized PEV quantified by the IPR value. This approach provides a comprehensive way to investigate not only the optimized network but also intermediate networks before an optimized structure is found. Furthermore, we restrict our study to the adjacency matrix of the network. It is also interesting to examine for other matrices related to graphs such as Laplacian matrix, modularity matrix. Here, eigenvector behavior has been regulated based on the particular function which is IPR. It is good to define another function that can tune the negative or zeros entries of the eigenvectors or some part of the eigenvector and based on that one can construct a network structure. This framework also helps to construct a network for other lower-order eigenvectors. Low order eigenvectors have also been studied to develop machine learning tools. They used IPR as well as another kind of measure for the eigenvector localization, called statistical leverage scores [34], [35] which has an impact on statistics and modern big data analysis.

It is also interesting to distribute weights in a weighted network so that PEV becomes localized [33]. Moreover, removing edges make the PEV delocalized, and also adding edges to the network produces delocalized PEV, but it may be possible to add a proper number of edges to the network which forms the highly localized PEV. We have not included the complexity analysis of the algorithms, and it is an exciting part to do in the future. Finally, we devise edge rewiring-based optimization algorithms which allow us to learn about the network structure from the PEV and it may be relevant to be used later to develop machine learning tools.

Acknowledgment: SJ acknowledges DST, Govt. of India grant EMR/2014/000368 for financial support. PP is indebted to Michael C. Grant (StackExchange, CVX Research, Stanford University) and Amit Reza (IIT Gandhinagar, India) for useful discussion on the optimization problem and members of CSL at IIT Indore for discussions.

References

[1] J. Aguirre, D. Papo, and J. M. Buldú, Successful strategies for competing networks, Nat. Phys. 9, 230 (2013).
[2] C. Castellano and R. Pastor-Satorras, Topological determinants of complex networks spectral properties: structural and dynamical effects, Phys. Rev. X 7, 041024 (2017).

[3] P. Pradhan, A. Yadav, S. K. Dwivedi, and S. Jalan, Optimized evolution of networks for principal eigenvector localization, Phys. Rev. E 96, 022312 (2017).

[4] P. Pradhan and S. Jalan, From Spectra to Localized Networks: A Reverse Engineering Approach, IEEE Transactions on Network Science and Engineering 7(4), (2020).

[5] D. Mohapatra, Jaccard Guided Perturbation for Eigenvector Localization, New Generation Computing 39, 159-179 (2021).

[6] R. Chaudhuri, A. Bernacchia, and X. Wang, A diversity of localized timescales in network activity, eLIFE 3, e01239 (2014).

[7] D. Taylor, S. A. Myres, Aaron Clauset, M. A. Porter, and P. J. Mucha, Eigenvector-based Centrality Measures for Temporal networks, SIAM Multiscale Model. Simul., vol. 15, No. 1, pp. 537-574, (2017).

[8] M. E. J. Newman, Networks: An Introduction, Oxford University Press, (2010).

[9] Ravindran Kannan and Santosh Vempala, Spectral Algorithms, Journal Foundations and Trends in Theoretical Computer Science, vol 4, Issue 3-4, pp. 157-288, (2009).

[10] C. Li, H. Wang and P. V. Mieghem, Degree and Principal Eigenvectors in Complex Networks, Networking, Lecture Notes in Computer Science, vol 7289, 149 (2012).

[11] G. Timár, R. A. da Costa, S. N. Dorogovtsev, and J. F. F. Mendes, Nonbacktracking expansion of finite graphs, Phys. Rev. E 95, 042322 (2017).

[12] T. Nishikawa, J. Sun, and A. E. Motter, Sensitive Dependence of Optimal Network Dynamics on Network Structure, Phys. Rev. X 7, 041044 (2017).

[13] F. Slanina, and Z. Konopasek, Eigenvector Localization as a tool to study small communities in Online Social Networks, World Scientific, Advances in Complex Systems, (2010).

[14] Think locally, act locally: Detection of small, medium-sized, and large communities in large networks, Phys. Rev. E 91, 012821 (2015).

[15] D. Taylor, R. S. Caceres, and P. J. Mucha, Super-Resolution Community Detection for Layer-Aggregated Multilayer Networks, Phys. Rev X 7, 031056 (2017).

[16] P. A. Lee and T. V. Ramakrishnam, Disordered electronic systems Rev. Mod. Phys. 57, 287 (1985).

[17] Y. Dekel, J. R. Lee, N. Linial, Eigenvectors of Random Graphs: Nodal Domains, Random Structures & Algorithms 39, 39 (2011).

[18] C. Bordenave and A. Guionnet, Localization and delocalization of eigenvectors for heavy-tailed random matrices Probability Theory and Related Fields 157, 885-953 (2013).

[19] X. Liu, G. Strang, and S. Ott, Localized eigenvectors from widely spaced matrix modifications, SIAM J. Discrete Math., vol. 16, No. 3, pp. 479-498, (2003).

[20] David F. Gleich, Michael W. Mahoney, Using Local Spectral Methods to Robustify Graph-Based Learning Algorithms, KDD’15, Sydney, NSW, Australia, 10-13, (2015).

[21] Pan Zhang, Robust Spectral Detection of Global Structures in the Data by Learning a Regularization, 29th Conference on Neural Information Processing Systems (NIPS), Barcelona, Spain (2016).

[22] C. Vömel, and B. N. Parlett, Detecting localization in an invariant subspace, Society for industrial and Applied Mathematics (SIAM) 33, 3447 (2011).
Next, we discuss in details about the objective function and constraints.

**Lemma 1.** $\zeta(x_1) = \sum_{i=1}^{n} (x_1)_i^4$ is a convex function when $(x_1)_i \in \{0, 1\}, i \in \{1, 2, \ldots, n\}$.

**Proof.** Convexity of the objective function $\zeta(x_1)$ can be examined by employing Hessian test [36]. One can construct the Hessian matrix from $\zeta(x_1)$ and show that it is positive semidefinite. The partial derivative of $\zeta(x_1)$ are given by

$$\frac{\partial \zeta(x_1)}{\partial (x_1)_i} = 4(x_1)_i^3, \ i \in \{1, 2, \ldots, n\}$$
and hence,
\[ \frac{\partial^2 \zeta(x_1)}{\partial (x_1)_i \partial (x_1)_j} = \begin{cases} 12(x_1)_i^2 & i = j \\ 0 & i \neq j \end{cases} \]  

(4)

Now we can write the Hessian matrix as
\[ \nabla^2 \zeta(x_1) = \begin{bmatrix} 12(x_1)_1^2 & 0 & \cdots & 0 \\ 0 & 12(x_1)_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 12(x_1)_n^2 \end{bmatrix} \]

Hessian matrix is positive semidefinite if all the eigenvalues of \( \nabla^2 \zeta(x_1) \) are non-negative. Here it is clear that eigenvalues of \( \nabla^2 \zeta(x_1) \) are \( \{12(x_1)_i^2 : i = 1, 2, \cdots n\} \) as \( \nabla^2 \zeta(x_1) \) is a diagonal matrix. Since \( (x_1)_i \in (0, 1) \), therefore all the eigenvalues of \( \nabla^2 \zeta(x_1) \) are non-negative, and hence the Hessian matrix is a positive semidefinite matrix. Therefore, the objective function \( \zeta(x_1) \) is a convex function.

**Lemma 2.** \( C = \{ x \in (0, 1)^n | ||x||_2 = 1 \} \) is a non-convex set.

**Proof.** A set \( C \subseteq \mathbb{R}^n \) is called convex if for any \( x, y \in C, x \neq y \) and any \( \theta \in [0, 1] \), the point \( \theta x + (1 - \theta)y \) belongs to \( C \) \[36\]. To validate \( C \) as a non-convex set, any arbitrary point \( z \in \mathbb{R}^n \) has been considered and it can be written as a convex combinations of \( x \) and \( y \) i.e., \( z = \theta x + (1 - \theta)y \) by choosing an arbitrary value of \( \theta \). Thus, we have
\[ (z_1, z_2, \ldots, z_n) = (\theta x_1 + (1 - \theta)y_1, \theta x_2 + (1 - \theta)y_2, \ldots, \theta x_n + (1 - \theta)y_n) \]  

(5)

From the above equation, we get,
\[ \sum_{i=1}^{n} z_i^2 = \theta^2 + (1 - \theta)^2 + 2\theta(1 - \theta) \sum_{i=1}^{n} x_i y_i \]  

(6)

Now, to check the convexity, one has to show that \( z \in C \), i.e., \( \sum_{i=1}^{n} z_i^2 = 1 \). Since \( x \neq y \), it gives \( \sum_{i=1}^{n} x_i y_i \neq 1 \). Now, for specific \( \theta = 1/2 \), \( \sum_{i=1}^{n} z_i^2 \neq 1 \), this implies that the relation \( \sum_{i=1}^{n} z_i^2 = 1 \) does not satisfy for any arbitrary value of \( \theta \). Hence, \( z \notin C \) and therefore, \( C \) is a non-convex set.

**Theorem 1.** Considering \( \zeta(x_1) \) as an objective function, principal eigenvector localization over undirected network is a non-convex optimization problem.

**Proof.** It is notable from Lemma 1 that the objective function \( \zeta(x_1) \) is a convex function but on the other hand, Lemma 2, says that the constraint, \( C = \{ x \in (0, 1)^n | ||x||_2 = 1 \} \) is a non-convex set. By definition, a convex optimization problems consist of minimizing of a convex functions over convex sets, or maximizing a concave functions over convex sets \[36\]. Jointly, conflicting characteristic of constraint and objective function shows that the principal eigenvector localization over simple undirected, and unweighted network is a non-convex optimization problem.