SIAM J. MATH. DATA SCI.
Vol. 3, No. 2, pp. 656–675

Nonbacktracking Eigenvalues under Node Removal: X-Centrality and Targeted Immunization[*]

Leo Torres[†], Kevin S. Chan[‡], Hanghang Tong[§], and Tina Eliassi-Rad[¶]

Abstract. The *nonbacktracking matrix* and its eigenvalues have many applications in network science and graph mining, such as node and edge centrality, community detection, length spectrum theory, graph distance, and epidemic and percolation thresholds. In network epidemiology, the reciprocal of the largest eigenvalue of the nonbacktracking matrix is a good approximation for the epidemic threshold of certain network dynamics. In this work, we develop techniques that identify which nodes have the largest impact on this leading eigenvalue. We do so by studying the behavior of the spectrum of the nonbacktracking matrix after a node is removed from the graph. From this analysis we derive two new centrality measures: *X-degree* and *X-nonbacktracking centrality*. We perform extensive experimentation with targeted immunization strategies derived from these two centrality measures. Our spectral analysis and centrality measures can be broadly applied, and will be of interest to both theorists and practitioners alike.

Key words. nonbacktracking, node immunization, centrality measure, complex networks

AMS subject classifications. 05C50, 05C82, 05C85, 68R10

DOI. 10.1137/20M1352132

1. Introduction.

A *nonbacktracking walk* (NB-walk) in a graph is a sequence of pairwise adjacent edges such that no edge is traversed twice in succession, i.e., the walk does not contain *backtracks*. NB-walks are known to mix faster than standard random walks [3], while *nonbacktracking cycles* (i.e., closed NB-walks) contain important topological information from the so-called *length spectrum* of the graph [42]. The associated *nonbacktracking matrix* is the unnormalized transition matrix of a random walker that does not trace backtracks, and it has many applications in community detection [9, 27], influencer identification [33, 32], graph

[*] Received by the editors July 13, 2020; accepted for publication (in revised form) January 12, 2021; published electronically May 13, 2021.
https://doi.org/10.1137/20M1352132
Funding: The first and fourth authors were funded in part by National Science Foundation grant IIS-1741197 and by the Combat Capabilities Development Command Army Research Laboratory through Cooperative Agreement W911NF-13-2-0045 and U.S. Army Research Lab Cyber Security CRA. The third author was partially supported by NSF grants 1947135, 2003924, and 1939725. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Combat Capabilities Development Command Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes not withstanding any copyright notation hereon.

[†] Network Science Institute, Northeastern University, Boston, MA 02115 USA (leo@leotrs.com).
[‡] U.S. Army Research Lab, Adelphi, MD 20783 USA (kevin.s.chan.civ@mail.mil).
[§] Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801-2302 USA (htong@illinois.edu).
[¶] Network Science Institute and Khoury College of Computer Sciences, Northeastern University, Boston, MA 02115 USA (tina@eliassi.org).

distance [42, 31], centrality [30, 6], etc. The nonbacktracking framework has also been adapted to directed [4], weighted [23], and multilayer [5] networks. In this paper, to avoid repetition we use the prefix "NB" to mean nonbacktracking; for example, we refer to the nonbacktracking matrix as the NB-matrix.

The eigenvalues of the NB-matrix (or NB-eigenvalues for short) are related to certain kinds of network dynamics. Karrer, Newman, and Zdeborová [22] and Hamilton and Pryadko [20] showed that the percolation threshold is approximated by the inverse of the largest NB-eigenvalue λ_1. This implies that the epidemic threshold of susceptible-infectious-recovered (SIR) dynamics can also be approximated by λ_1 [36, 34]. Shrestha, Scarpino, and Moore [40] argued the same for susceptible-infectious-susceptible (SIS) dynamics, though Castellano and Pastor-Satorras [10] highlight that this may only hold for networks with certain amounts of degree heterogeneity. Whether one is talking about percolation or dynamics, λ_1 provides a better approximation to the true threshold than the largest eigenvalue of the adjacency matrix [40, 22].

Given the importance of the largest NB-eigenvalue, we ask the following: Which nodes influence the largest NB-eigenvalue the most? In the cases of SIR and SIS dynamics, answering this question will lead to targeted immunization strategies, as it is equivalent to asking which are the nodes whose removal from the network causes the epidemic threshold to increase the most. Operationally, we frame this question as follows. Consider a graph G with largest NB-eigenvalue λ_1. Given a node c in G, let $\lambda_1(c)$ be the largest NB-eigenvalue of the network after removing c. Define $\lambda_1 - \lambda_1(c)$ as the *eigendrop induced by c. Which node c induces the maximum eigendrop?*

In section 2 we present the necessary background theory. In section 3 we develop a spectral perturbation theory of the NB-matrix. We use this theory in section 4 to introduce two new centrality measures and argue why they are effective at identifying nodes with large eigendrops. In section 5 we review previous studies related to the present work. In section 6 we provide experimental evidence for our claims. We conclude the paper in section 7.

2. Background.

Let G be a simple, unweighted, undirected graph with node set V and edge set E. We consider the set of directed edges $\overline{E}$ where each undirected edge $(i,j) \in E$ gives rise to two directed edges $i \to j$ and $j \to i$ in $\overline{E}$. A *walk* in G is a sequence of directed edges $i_1 \to j_1, \ldots, i_k \to j_k$, where $j_r = i_{r+1}$ for each $r = 1, \ldots, k-1$. Here, k is the *length* of the walk. A walk is *closed* if $j_k = i_1$. A *backtrack* is a walk of length 2 of the form $i \to j, j \to i$. A walk is an *NB-walk*, if no two consecutive edges in it form a backtrack. The *nonbacktracking matrix*, or *NB-matrix*, B is the unnormalized transition matrix of a walker that does not perform backtracks. B is indexed in the rows and columns by elements of $\overline{E}$. Let $m = |E|$; then B is of size $2m \times 2m$, and it is defined by

$$B_{k\to l, i\to j} = \delta_{jk}\left(1 - \delta_{il}\right), \tag{2.1}$$

where δ is the Kronecker delta. In words, $B_{k\to l,i\to j}$ is 1 iff $j = k$ and $i \to j, j \to l$ is not a backtrack. Notably, the powers of B count the number of NB-walks in G, i.e., $(B^r)_{k\to l,i\to j}$ is the number of NB-walks that start with $i \to j$ and end with $k \to l$ of length $r+1$.

Among other applications, the NB-matrix has been used to define a notion of node centrality [30, 38]. Concretely, let λ_1 be the largest eigenvalue of B, and let $\mathbf{v}$ be the corresponding unit right eigenvector. By the Perron–Frobenius theory, λ_1 is positive, real, and has multiplicity one, while $\mathbf{v}$ can be chosen to be nonnegative. If $A = (a_{ij})$ is the adjacency matrix of G, the *NB-centrality* of a node i is defined as

$$\mathbf{v}^i = \sum_j a_{ij} \mathbf{v}_{j \to i}. \tag{2.2}$$

The NB-matrix is not symmetric. Therefore, its eigenvalues, other than the largest one, can be complex numbers. Even so, it contains a subtle structure, sometimes called PT-symmetry [9]. Indeed, let P be the matrix such that $P\mathbf{x}_{i\to j} = \mathbf{x}_{j \to i}$ for any vector $\mathbf{x}$ indexed by $\overline{E}$. It is readily checked that (i) $P^2 = I$, and (ii) the product PB is symmetric.

3. NB-eigenvalues under node removal.

We are interested in the behavior of the NB-eigenvalues when we remove a node from G. Suppose the target node we want to remove is $c \in V$. Then partition the edges in $\overline{E}$ as those that are incident to c and those that are not. Sort the rows and columns of B accordingly so that it takes the block form

$$B = \begin{pmatrix} B' & D \\ E & F \end{pmatrix}, \tag{3.1}$$

as shown in Figure 1. Here, B' is the NB-matrix of the graph after node c is removed, while F is the NB-matrix of the star graph centered at c; if d is the degree of c, then F is of size $2d \times 2d$. Further, D is indexed in the rows by directed edges not incident to c, and in the columns by directed edges incident to c, and vice versa for E.

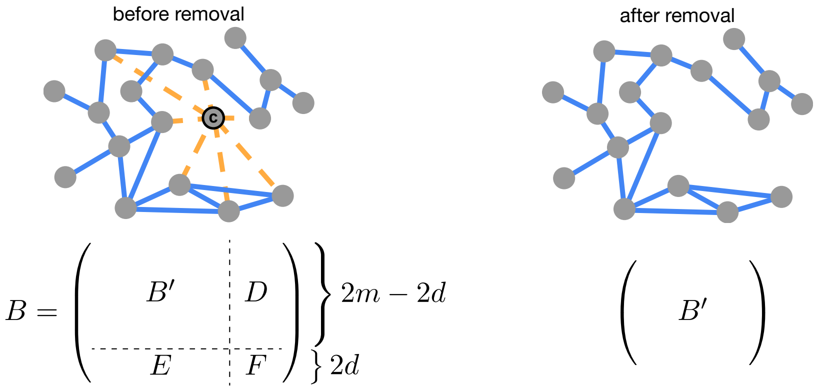
before removal
c
after removal
$B = \left(\begin{array}{c:c} B' & D \\ \hdashline E & F \end{array}\right) \begin{array}{l} \}\ 2m-2d \\ \}\ 2d \end{array}$
$\begin{pmatrix} B' \end{pmatrix}$

Figure 1. Top: *Graph G with target node c before and after removal. G has m edges and c has degree d. Dashed yellow edges are incident to c, all other edges in solid blue.* Bottom: *Corresponding NB-matrices.*

3.1. The characteristic polynomial.

The NB-eigenvalues are the roots of the characteristic polynomial $\det(B - tI)$. Applying the theory of Schur complements gives

$$\det(B - tI) = \begin{vmatrix} B' - tI & D \\ E & F - tI \end{vmatrix} = \det(F - tI) \det\left(B' - tI - D(F - tI)^{-1} E\right), \tag{3.2}$$

where the size of I is given by context. Equation (3.2) holds whenever $(F - tI)$ is invertible, i.e., whenever t is not an eigenvalue of F. To simplify (3.2), we use the following lemma.

Lemma 3.1. *Let d be the degree of target node c. With D, E, F as in (3.1), we have $DE = 0$ and $F^2 = 0$. Therefore, F is nilpotent, that is, all its eigenvalues are zero, and hence $\det(F - tI) = t^{2d}$. Finally, we have $(F - tI)^{-1} = -(F + tI)/t^2$ when $t \neq 0$.*

Proof. Since D, E, F are submatrices of B, their element is given by (2.1). Hence, computing DE and F^2 is straightforward, as long as care is placed in keeping track of the appropriate indices for the rows and columns of the involved matrices. Now, $F^2 = 0$ implies that all its eigenvalues are zero and that $\det(F - tI) = t^{2d}$. Finally, one can manually check that $(F - tI)(F + tI) = -t^2 I$. ■

Now define $X = DFE$. One can manually check that

$$X_{k\to l, i\to j} = a_{ck} a_{cj} (1 - \delta_{kj}). \tag{3.3}$$

Per the previous lemma, (3.2) holds for $t \neq 0$ and

$$\det(B - tI) = t^{2d} \det\left(B' - tI + \frac{DFE}{t^2} + \frac{DE}{t}\right) = t^{2d} \det\left(B' - tI + \frac{X}{t^2}\right). \tag{3.4}$$

Theorem 3.2. *For a graph G and target node c, suppose the NB-matrix of G is B and the NB-matrix after removing c is B', and let X be as in (3.4). If t is not an eigenvalue of B', then we have*

$$\frac{\det(B - tI)}{\det(B' - tI)} = t^{2d} \det\left(I + \frac{1}{t^2}(B' - tI)^{-1} X\right). \tag{3.5}$$

Proof. The proof is immediate from (3.4) by factoring out $B' - tI$. ■

If c has degree 1, then X equals the zero matrix, and (3.5) simplifies showing that removing c has no influence on the nonzero NB-eigenvalues. More generally, the nodes whose removal does not influence the nonzero NB-eigenvalues are characterized as follows. Let the 2-*core* of G be the graph that remains after iteratively removing nodes of degree 1. Let the 1-*shell* of G be the graph induced by the nodes outside of the 2-core. Nodes in the 1-shell do not affect the nonzero NB-eigenvalues. This fact is well known [15, 23, 26, 45], though we present here a new proof.

Corollary 3.3. *Removing a node c from the 1-shell doesn't change the nonzero NB-eigenvalues.*

Proof. If c has degree 1, (3.3) gives $X = 0$ and (3.5) becomes $\det(B - tI) = t^{2d} \det(B' - tI)$, which implies the assertion. In general, if c is in the 1-shell, then it must have degree 1 after iteratively removing some sequence of nodes each of which has degree 1 at the time of removal. Each of these removals has no effect on the nonzero eigenvalues. Therefore, neither does the removal of c. ■

Remark 3.4. Intuitively, since F is the NB-matrix of a star graph, which contains no NB-walks of length 3 or more, then we immediately have $F^2 = 0$. Following Figure 1, F^2 counts the number of NB-walks of length 3 whose edges are yellow-yellow-yellow, of which there are none. Similarly, DE counts the NB-walks whose edges are blue-yellow-blue, which also do not exist. Finally, $X = DFE$ counts the NB-walks of color blue-yellow-yellow-blue, which are precisely those that are destroyed when removing c. It is then no surprise that the rest of our analysis pivots fundamentally on the matrix X.

3.2. The largest eigenvalue.

We now study the eigendrop induced by removing c. The larger this eigendrop, the more influential c is in determining the epidemic threshold.

Theorem 3.5. *With the same assumptions as in Theorem 3.2, let λ_1 be the largest eigenvalue of B, and let $\mathbf{w}$ be a vector such that in (3.4) we have*

$$\left(B' - \lambda_1 I + \frac{X}{\lambda_1^2}\right)\mathbf{w} = 0. \tag{3.6}$$

Suppose $\{\mathbf{v}_i\}$ is a basis of right eigenvectors of B' and write $\mathbf{w}$ in this basis, $\mathbf{w} = \sum_i w_i \mathbf{v}_i$. Let $\mathbf{u}_1$ be the left eigenvector of B' corresponding to $\mathbf{v}_1$, and set $\alpha_i = \mathbf{u}_1^T X \mathbf{v}_i$. Finally, let λ_1' be the largest eigenvalue of B', so that the eigendrop induced by c is $\lambda_1 - \lambda_1'$. Then, we have

$$\lambda_1 - \lambda_1' = \frac{1}{\lambda_1^2}\sum_i \frac{w_i}{w_1}\alpha_i. \tag{3.7}$$

Proof. If $\mathbf{v}_i$ corresponds to the eigenvalue λ_i', then (3.6) gives

$$\sum_i w_i \left(B' - \lambda_1 I + \frac{X}{\lambda_1^2}\right)\mathbf{v}_i = \sum_i w_i \left(\lambda_i'\mathbf{v}_i - \lambda_1\mathbf{v}_i + \frac{X\mathbf{v}_i}{\lambda_1^2}\right) = 0. \tag{3.8}$$

Let $\mathbf{u}_1$ be the left eigenvector corresponding to $\mathbf{v}_1$ normalized such that $\mathbf{u}_1^T\mathbf{v}_1 = 1$. Recall that $\mathbf{u}_1$ is orthogonal to every right eigenvector corresponding to a different eigenvalue. Since λ_1' has multiplicity one, we have $\mathbf{u}_1^T\mathbf{v}_i = 0$ for each $i \neq 1$. Multiply by $\mathbf{u}_1$ on the left to get

$$w_1\left(\lambda_1' - \lambda_1\right) + \sum_i w_i \frac{\mathbf{u}_1^T X \mathbf{v}_i}{\lambda_1^2} = 0. \tag{3.9}$$

Define $\alpha_i = \mathbf{u}_1^T X \mathbf{v}_i$ and rearrange to get (3.7). ■

Remark 3.6. We can interpret our arguments in terms of node addition rather than removal. Suppose the original graph does not contain c, and therefore, its NB-matrix is B'. Then, the NB-matrix *after adding node c* is given by (3.1). All our arguments are valid in this setting, and (3.7) then says that the new largest NB-eigenvalue is the solution to a third-degree polynomial, the coefficients of which depend on the full eigendecomposition of B'.

3.2.1. An approximation.

Unfortunately, (3.7) requires knowledge of all eigenvectors of B'. However, in our experience, the vector $\mathbf{w}$ is extremely closely aligned to $\mathbf{v}_1$, and therefore, the coefficients $w_i/w_1 \ll 1$. In this case, all but one term on the right-hand side of (3.7) can be neglected and we get

$$\lambda_1^2\left(\lambda_1 - \lambda_1'\right) - \alpha_1 \approx 0. \tag{3.10}$$

Here, the larger α_1, the larger the eigendrop $\lambda_1 - \lambda_1'$. Therefore, we focus on α_1 next.

Proposition 3.7. *Let $\mathbf{u}_1, \mathbf{v}_1$ be the left and right eigenvectors of B' normalized such that $\mathbf{u}_1^T\mathbf{v}_1 = 1$. Then we have*

$$\alpha_1 = \mathbf{u}_1^T X \mathbf{v}_1 = \mathbf{v}_1^T P X \mathbf{v}_1 = \left(\sum_i a_{ci}\mathbf{v}_1^i\right)^2 - \sum_i a_{ci}\left(\mathbf{v}_1^i\right)^2, \tag{3.11}$$

where $\mathbf{v}_1^i$ *is the NB-centrality of node* i *in the graph after removal (see* (2.2)*). We call* α_1 *the* X-nonbacktracking centrality*, or* X-NB centrality*, of* c*.*

Proof. The first equality comes from the fact that $\mathbf{u}_1 = P\mathbf{v}_1$, by Lemma A.1. We can find $PX_{k\to l,i\to j} = a_{cl}a_{cj}\left(1-\delta_{lj}\right)$ using (3.3) and the fact that $P^2 = I$. The result then follows from manually computing $\mathbf{v}_1^T PX\mathbf{v}_1$ and applying (2.2). ■

The previous proposition establishes that the behavior of the (approximate) eigendrop in (3.10) is governed by the X-NB centrality of c in (3.11), which is a function only of the NB-centralities of c's neighbors. Importantly, these centralities are measured *after* c is removed. We come back to this point in section 4. Notably, the principal eigenvector is normalized by $\mathbf{u}_1^T\mathbf{v}_1 = \mathbf{v}_1^T P\mathbf{v}_1 = 1$, i.e., it does not have unit length. Finally, note that Proposition 3.7 can be considered a generalization of equation D1 of [47], which treats the case of single edge removal.

3.2.2. An upper bound. An alternative way of studying the eigendrop is by choosing $\mathbf{w}$ such that $w_1 = 1$, and bounding $q = q(c) = \sum_i w_i\alpha_i$, which drives the right-hand side of (3.7). Suppose that R is the matrix whose columns are the eigenvectors $\{\mathbf{v}_i\}$, and let $L = R^{-1}$ such that $B' = R\Lambda L$, where Λ is the diagonal matrix of the eigenvalues $\{\lambda_i'\}$. The rows of L are left eigenvectors of B', in particular, $\mathbf{u}_1^T$ is the first row of L. Then we have $\alpha_i = \mathbf{u}_1^T X\mathbf{v}_i = (LXR)_{1i}$, and q is the dot product between the first row of LXR and $\mathbf{w}$,

$$q = \mathbf{e_1^T} LXR\mathbf{w} = \mathrm{Tr}\left(LXR\,\mathbf{w}\mathbf{e_1^T}\right), \tag{3.12}$$

where $\mathbf{e_1} = (1,0,\ldots,0)$. Using the cyclic property of the trace, and that $P^2 = I$, we have

$$q = \mathrm{Tr}\left(LXR\,\mathbf{w}\mathbf{e_1^T}\right) = \mathrm{Tr}\left(XR\,\mathbf{w}\mathbf{e_1^T}\,L\right) = \mathrm{Tr}\left(PXR\,\mathbf{w}\mathbf{e_1^T}\,LP\right). \tag{3.13}$$

Applying the Cauchy–Schwarz inequality for the trace gives us $q \le |PX|_F\,\left|R\,\mathbf{w}\mathbf{e_1^T}\,LP\right|_F$, where $|M|_F^2 = \mathrm{Tr}\left(M^TM\right)$. Finally, since $\mathbf{w}\mathbf{e_1^T}$ is a matrix with rank one, we have

$$q \le |PX|_F\,\left(\mathbf{e_1^T}LPR\mathbf{w}\right). \tag{3.14}$$

As before, we have $w_i/w_1 \ll 1$ and since we chose $w_1 = 1$, the term $\left(\mathbf{e_1^T}LPR\mathbf{w}\right)$ is very close to 1. Therefore, we obtain $|PX|_F$ as an (approximate) upper bound for q. Observe that since PX is nonnegative, we have $|PX|_F = \mathbf{1}^TPX\mathbf{1}$, where $\mathbf{1} = (1,1,\ldots,1)$.

Proposition 3.8. *In* (3.7)*, let* $\mathbf{w}$ *be such that* $w_1 = 1$*, and define* $q = \sum_i w_i\alpha_i$*. The quantity* $\mathbf{1}^TPX\mathbf{1}$ *is an approximate upper bound for* q*, that is,* $q \le \mathbf{1}^TPX\mathbf{1}\left(\mathbf{e_1^T}LPR\mathbf{w}\right)$*. Furthermore, we have*

$$\mathbf{1}^TPX\mathbf{1} = \left(\sum_i a_{ci}\left(d_i-1\right)\right)^2 - \sum_i a_{ci}\left(d_i-1\right)^2, \tag{3.15}$$

where d_i *is the degree of node* i *before removal. We call* $\mathbf{1}^TPX\mathbf{1}$ *the* X-degree centrality *of* c*.*

Proof. The first claim was proved in the previous paragraphs. The second claim comes from direct evaluation of $\mathbf{1}^TPX\mathbf{1}$ using $PX_{k\to l,i\to j} = a_{cl}a_{cj}\left(1-\delta_{lj}\right)$. ■

Remark 3.9. Our arguments assume the existence of a full basis of eigenvectors of the NB-matrix. The proof of existence of such a basis evades us, though every single nonregular graph we have observed admits such a basis. Similarly, the fact that the X-degree of c bounds q only approximately also merits further theoretical consideration. However, these two theoretical caveats notwithstanding, our experiments show that, in practice, the X-degree and X-NB centrality of nodes are excellent predictors of the node's eigendrop.

3.3. X-centrality. In section 3.2.1 we use the X-NB centrality, $\mathbf{v}_1^T PX\mathbf{v}_1$, while in section 3.2.2 we use the X-degree centrality, $\mathbf{1}^T PX\mathbf{1}$, both for the purpose of studying the eigendrop induced by c. The former is a function of the NB-centralities of the neighbors of c (Proposition 3.7), while the latter is a function of their degrees (Proposition 3.8). Importantly, both are computed using the same quadratic form PX, applied to two different centralities, which are both measured *after* c has been removed.

Fix a target node c; this fixes X and P. We can use the matrix PX to define new node-level statistics as follows. Recall that if G has m (undirected) edges and c has degree d, then X and P are of size $2m-2d$. Given an arbitrary vector $\mathbf{z}$ of size $2m-2d$, we have

$$\mathbf{z}^T PX\mathbf{z} = \left(\sum_i a_{ci} \sum_j \mathbf{z}_{j\to i}\right)^2 - \sum_i a_{ci} \left(\sum_j \mathbf{z}_{j\to i}\right)^2 . \tag{3.16}$$

One can evaluate the right-hand side of (3.16) for any vector $\mathbf{z}$ of size $2m$, and use only the $2m-2d$ entries that correspond to edges not incident to c. In other words, we do not need to know X or P, but only who the neighbors of c are. Since c determines both X and P, the same vector $\mathbf{z}$ can be evaluated using different target nodes. Therefore, the quantity in (3.16) naturally corresponds to whichever target node was used to evaluate it, and can be thought of as a node-level quantity derived from $\mathbf{z}$ and aggregated via the PX matrix.

Now define $\mathbf{z}^i = \sum_j \mathbf{z}_{j\to i}$ and let $\mathrm{Var}_c\left(\mathbf{z}^i\right)$ be the variance of the $\mathbf{z}^i$ values corresponding to neighbors of c. Then we have

$$\mathrm{Var}_c\left(\mathbf{z}^i\right) = \frac{\sum_i a_{ci}\left(\mathbf{z}^i\right)^2}{d} - \left(\frac{\sum_i a_{ci}\mathbf{z}^i}{d}\right)^2 , \tag{3.17}$$

which differs from (3.16) only in sign and a (nonlinear) normalization. Accordingly, $\mathbf{z}^T PX\mathbf{z}$ will have large values when $\mathbf{z}^i$ has little variability among the neighbors of c.

Using this framework we could define, for example, *X-closeness centrality*, *X-betweenness centrality*, etc., though the utility of these other measures remains an open question.

4. Node immunization. Targeted immunization works as follows. Given a graph G and an integer p, we want to remove from G the p nodes that increase the epidemic threshold the most (equivalently, decrease the largest NB-eigenvalue the most). Common strategies involve three steps: (i) the nodes are sorted by decreasing values of a certain statistic, for example degree; (ii) the node with the highest value of this statistic is removed from the graph; and (iii) the statistic has to be recomputed after each time a node is removed. Repeat these steps p times. In this context, our framework presents two major obstacles:

(a) Both the X-NB and X-degree centralities of a node must be computed *after* the node has been removed. So, to execute step (i) above, we need to temporarily remove each node in turn before we decide which one to ultimately remove.

(b) For step (iii), we must guarantee that recomputing the statistic of every node at each step is an efficient procedure.

4.1. Using X-NB centrality.

Algorithm 1 naively follows the steps above to implement an immunization strategy based on X-NB centrality. We are tempted to think this strategy is the "right" one, as it approximates the true effect of a node's removal in the epidemic threshold. However, we must address the obstacles mentioned above.

```
Input: graph $G$, integer $p$
Output: removed, an ordered list of nodes to immunize
removed $\leftarrow \emptyset$
XNB[i] $\leftarrow 0$ for each node i
while length(removed) $< p$ do
    foreach node c in $G$ do
        $H \leftarrow$ RemoveNode($G$, c)
        $v_H \leftarrow$ principal eigenvector of AuxNBMatrix($H$)
        XNB[c] $\leftarrow$ XNBCentrality($v_H$, c)
    node $\leftarrow \arg\max_i$ XNB[i]
    $G \leftarrow$ RemoveNode($G$, node)
    removed.append(node)
return removed
```

Algorithm 1: Naive X-NB immunization strategy.

```
Input: graph $G$, integer $p$
Output: removed, an ordered list of nodes to immunize
removed $\leftarrow \emptyset$
XNB[i] $\leftarrow 0$ for each node i
while length(removed) $< p$ do
    $v_G \leftarrow$ principal eigenvector of AuxNBMatrix($G$)
    foreach node c in $G$ do
        XNB[c] $\leftarrow$ XNBCentrality($v_G$, c)
    node $\leftarrow \arg\max_i$ XNB[i]
    $G \leftarrow$ RemoveNode($G$, node)
    removed.append(node)
return removed
```

Algorithm 2: Approximate X-NB immunization strategy.

To overcome obstacle (a), we propose approximating (3.11) by using the NB-centralities in the original graph before removing any node even temporarily. Algorithm 2 takes this approximation into account. The error incurred by this approximation is dampened by the fact that what we are ultimately interested in is the ranking of the nodes rather than the actual values of their centralities. For obstacle (b), one could use a strategy similar to [11], where the authors devise an algorithm to approximate the impact on a node's eigenvector centrality after the removal of a node without having to recompute the values again. However, doing so for X-NB centrality remains an open question.

Complexity analysis. We assume that G is given in adjacency list format. In Algorithm 1, lines 2 and 8 take n operations each. Line 5 creates a copy H of the adjacency list and removes the target node c from it (but leaves G intact). Line 6 uses Lemma A.2 to compute the NB-centralities without computing the NB-matrix itself. Instead, it uses the auxiliary NB-matrix from (A.1), which takes $O(m)$ time, and it takes $O\left(m\right)$ to compute its principal eigenvector (using, e.g., the Lanczos algorithm with a number of iterations that does not depend on the parameters). Line 7 takes n time. The remaining lines take constant time. Accounting for loops, Algorithm 1 takes a total of $O\left(n + p\left(n\left(m+n\right)+n\right)\right) = O\left(pn\left(m+n\right)\right)$. In Algorithm 2, the NB-centralities are computed outside of the inner loop, which gives a complexity of $O\left(p\left(m+n\right)\right)$, or $O\left(m+n\right)$ for constant p.

4.2. Using X-degree.

X-degree can be easily computed without temporarily removing any nodes; see (3.15). Indeed, all we need to know about the graph after removal is the degree of each node. Hence, obstacle (a) is easily overcome in this case. Further, after each step we need not recompute the X-degree of all nodes, but only of those nodes two steps away from the target node. Indeed, removing c changes the degree of its neighbors, which in turn changes the X-degree of its neighbors' neighbors. So obstacle (b) is also overcome. Algorithm 3 implements this strategy. Importantly, it does not involve the computation of any matrices or their eigenvectors.

Complexity analysis. Lines $1, 6, 10$, and 11 of Algorithm 3 take constant time, while line 2 takes $O(m)$. When using a standard map (or dictionary) to store the X-degree values, line 4 takes $O(n)$ operations, and line 9 takes $O(1)$. Now suppose that the nodes removed by Algorithm 3 are, in order, $i_1, \ldots, i_p$. At iteration j, the loop in line 5 takes d_{i_j} operations, and the double loop in lines 7 and 8 takes as many iterations as the number of nodes two steps away from i_j, say D_{i_j}. This yields a total of $O\big(m + pn + \sum_{j=1}^{p} d_{i_j} + \sum_{j=1}^{p} D_{i_j}\big)$. We can also implement Algorithm 3 using an indexed priority queue (IPQ) to store the X-degree values instead of a map; see Appendix B. In this case the worst-case scenario complexity is $O\big(m + p \log n + \sum_{j=1}^{p} d_{i_j} + \log n \sum_{j=1}^{p} D_{i_j}\big)$. In Appendix B we refine this analysis for networks with homogeneous or heterogeneous degree distributions, and show that neither the map nor the IPQ version is better than the other in all cases.

Importantly, the average runtime of both versions is, in fact, close to linear, with the IPQ version being the faster. Figure 2 shows the average runtime of both versions on random power-law configuration model graphs with varying degree exponent γ and constant p (see Appendix B for details). The reason the average runtime is considerably faster than the worst-case scenario is because graphs typically have very few large hubs. That is, roughly speaking, there are $O(1)$ many nodes that take $O(n)$ time to process, while there are $O(n)$ many nodes that take $O(1)$ time to process. This effect is intensified the closer γ is to 2, which

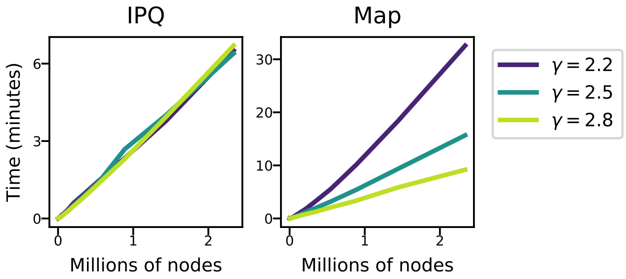

Figure 2. *Average runtime of Algorithm 3 on random power-law graphs with degree exponent γ.*

counterbalances the exponent $\frac{2}{\gamma-1}$ in the worst-case scenario.

```
Input: graph $G$, integer $p$
Output: removed, an ordered list of nodes to immunize
removed $\leftarrow \emptyset$
XDeg[i] $\leftarrow$ XDegree($G$, i) for each node i
while length(removed) $< p$ do
    node $\leftarrow \max_i$ XDeg[i]
    foreach i in $G$.neighbors[node] do
        $G$.neighbors[i].remove(node)
    foreach i in $G$.neighbors[node] do
        foreach j in $G$.neighbors[i] do
            XDeg[j] $\leftarrow$ XDegree($G$, j)
    $G$.neighbors[node] $\leftarrow \emptyset$
    removed.append(node)
return removed
```

Algorithm 3: X-degree immunization strategy.

5. Related work.

Perturbation of NB-matrix. Zhang [47] briefly treats the case of eigenvalue perturbation of a matrix derived from the NB-matrix in the case of edge removal, while Coste and Zhu [12] analyze the perturbation of quadratic eigenvalue problems, with applications to the NB-eigenvalues of the stochastic block model. Our theory is more general since it studies node removal (as opposed to single edge removal), and it applies to any arbitrary graph.

NB centrality. Many notions of centrality based on the NB-matrix exist, for example NB-PageRank [6], NB-centrality [30, 38], and Collective Influence [32, 33]. The latter two have been proposed as solutions to the problem of "influencer identification." This problem aims to find nodes that determine the course of spreading dynamics, and is thus more general than our objective of increasing the epidemic threshold. Collective Influence in particular is similar to X-degree; see Appendix C.1. Also in this context, Kitsak et al. [24] propose using the k-core index, and Poux-Médard, Pastor-Satorras, and Castellano [37] highlight the importance of node degree. We compare our algorithms to all of these baselines in section 6. Finally, Everett and Borgatti [16] study the influence of a node's removal in other nodes' centrality, which is reminiscent of our X-centrality framework.

Targeted immunization. Pastor-Satorras et al. [36] review the generalities of spreading dynamics on networks, including immunization strategies. Chen et al. [11] propose `NetShield`, an algorithm for immunization focusing on decreasing the largest eigenvalue of the adjacency matrix. We prefer to focus on decreasing the largest NB-eigenvalue instead since it provides a tighter bound to the true epidemic threshold in most cases [22, 20, 40]. Lin, Chen, and Zhang, [29] study the percolation threshold in terms of so-called high-order NB-matrices. Percolation thresholds are tightly related to epidemic thresholds of SIR dynamics [36, 34].

6. Experiments.

6.1. Approximating the eigenvalue.

How close is the approximation in (3.10)*?* We first compute the largest NB-eigenvalue λ_1 of a graph G. Then we fix a target node c and remove it from G and compute the new eigenvalue λ_c. (For ease of notation, in this section we use λ_c instead of λ'_1, and α instead of α_1.) Finally, we use (3.10) to compute two approximations, $\widehat{\lambda}_c = \lambda_1 - \alpha/\lambda_1^2$ and $\widetilde{\lambda}_c = \lambda_1 - \widetilde{\alpha}/\lambda_1^2$, where α is the true X-NB centrality of c, and $\widetilde{\alpha}$ is the approximate X-NB centrality used in Algorithm 2, i.e., it is computed using the NB-centralities before removing c. We now compare the approximations $\widehat{\lambda}_c$ and $\widetilde{\lambda}_c$ to the true value of λ_c for randomly selected nodes of synthetic graphs. We use different synthetic random graph models: Watts–Strogatz (WS) [44], Stochastic Block Model (SBM) [17, 21], Barabási–Albert (BA) [1], and Block Two–Level Erdős–Rényi (BTER) [39]. See section C.2 for details on the data sets, and section C.3 for the experimental setup.

Figure 3a shows that our approximation is extremely close for all graphs tested, though it tends to underestimate the eigendrop in WS graphs. Figure 3c shows the average relative error versus degree. Our approximation worsens as degree increases, though it is quite small for most degrees. In the worst case, the relative error is less than 10^{-4}, or 0.01%. Figure 3b shows the eigendrop computed using the approximate version of X-NB. This approximation is systematically overestimating the true eigendrop. Figure 3d shows that this systematic error is of the order of 10% in the worst case, though it is negligible for small degrees. Overall, Figure 3 confirms the accuracy of our approximations, and it points to the fact that the terms neglected in (3.10) will become larger as degree increases.

6.2. Predicting the eigendrop.

How well can X-NB centrality and X-degree predict a node's eigendrop? Unlike in subsection 6.1, here we do not approximate the eigendrop, but only seek to predict its size. See section C.4 for experimental setup, and section C.2 for details on data sets.

In Figure 4a we measure how correlated the true value of X-NB, denoted by α, is to the true eigendrop. For SBM, BA, and BTER graphs, the magnitude of α lines up extremely closely with the value of the eigendrop, showing a correlation coefficient of $r = 1.00$. In all cases, α is better correlated to the eigendrop than degree (as shown by the correlation coefficients r_{deg}). In WS we see considerably more variance than in other ensembles though α is still an excellent predictor of the eigendrop at $r = 0.98$. This picture repeats itself when using the approximate value of X-NB, $\widetilde{\alpha}$ (Figure 4b), and X-degree (Figure 4c). $\widetilde{\alpha}$ seems to

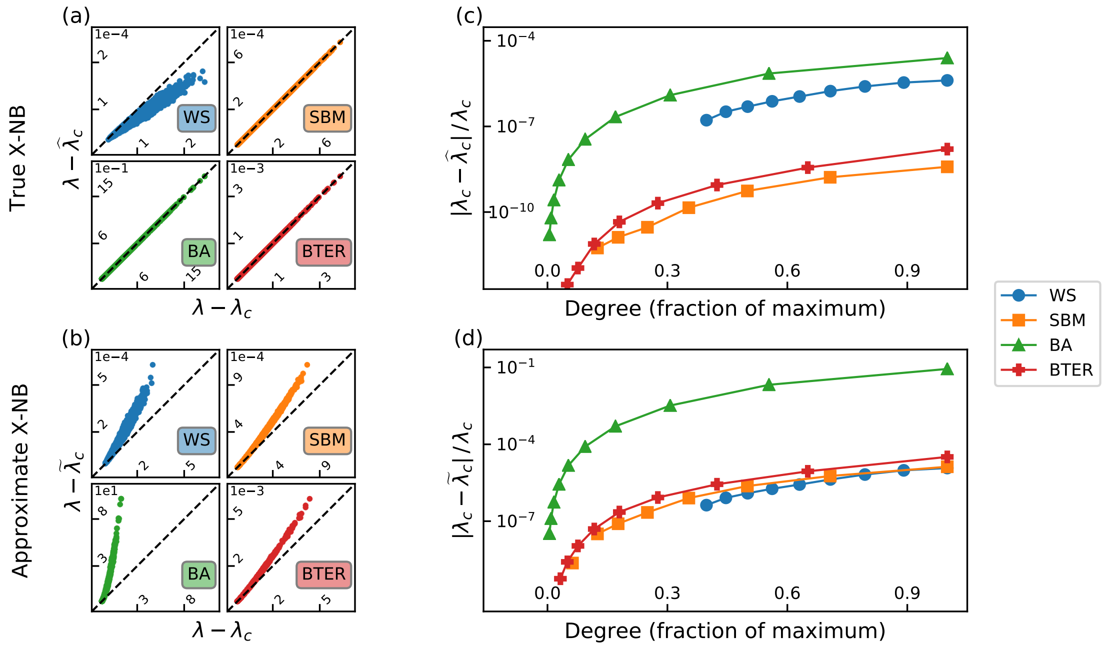

Figure 3. Left: *True eigendrop (horizontal axis) versus approximate eigendrop (vertical axis) using* (a) *true and* (b) *approximate values of X-NB. Dashed line is $y = x$. Each marker represents one node.* Right: *Relative error when predicting λ_c, as a function of degree, using* (c) *true and* (d) *approximate values of X-NB. Degrees expressed as a fraction of the maximum degree among graphs are in the same ensemble. Each marker is the average within log-binned values of degree; error bars too small to show at this scale. WS graphs (blue circles) have no nodes whose degree is less than 30% of the maximum.*

slightly underestimate the eigendrop, while X-degree has noticeably more variance than the other two statistics, especially in WS. All three statistics are better correlated to the eigendrop than degree in all graph ensembles. We highlight that even when a few of the panels in Figure 4 are not precisely linear, they all show that the eigendrop is an increasing function of all of α, $\widetilde{\alpha}$, and X-degree. Note that for X-NB and X-degree to be effective immunization strategies, all we need is that they be positively correlated to the true eigendrop, which is shown by the results of Figure 4. Further, using α has very little advantage over $\widetilde{\alpha}$, and therefore, we are justified in using Algorithm 2 instead Algorithm 1 for computational reasons.

6.3. Immunization with X-NB and X-degree. *How effective are X-NB centrality and X-degree at immunization?* We remove 1%, 2%, and 3% of nodes using different strategies and evaluate the resulting eigenvalue. We use the immunization strategies node degree (`degree`), k-core index (`core`), highest degree within the 2-core (`coreHD`), NetShield (`NS`), Collective Influence (`CI`), NB-centrality (`NB`), X-degree (`Xdeg`), and approximate X-NB (`XNB`). For computational reasons, we do not use the true value of X-NB; for more details on baselines, see section C.1. In all data sets, `core` had the least performance and is therefore not shown in our results. We hypothesize this is because many nodes can have the same k-core index at

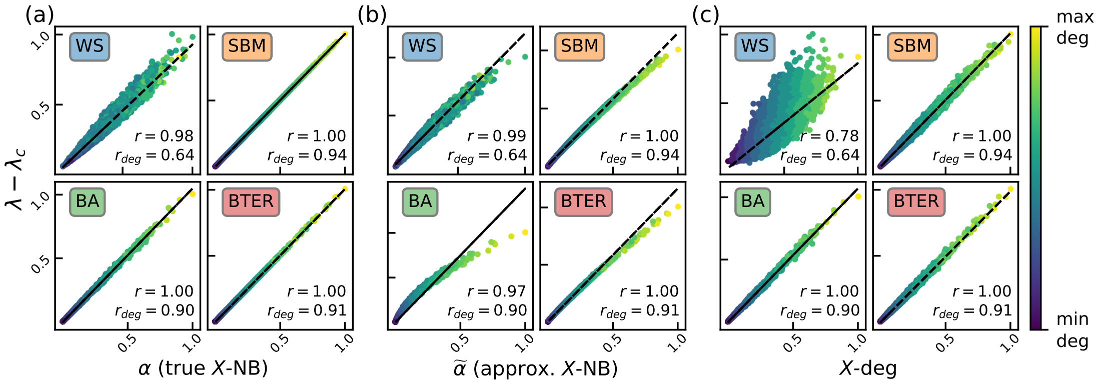

Figure 4. *Predicting the eigendrop using* (a) *true value of X-NB,* (b) *approximate value of X-NB, and* (c) *X-degree. Markers are colored by degree, expressed as a fraction of the largest degree in the same ensemble. Each panel shows the correlation coefficient between the corresponding statistic and the eigendrop (r), and the correlation between degree and the eigendrop (r_{deg}). Dashed lines are linear regression lines. Our proposed node-level statistics accurately track eigendrops in random graph models such as BA, SBM, and BTER. X-degree underestimates the eigendrop in WS graphs.*

Table 1

Average percentage eigendrop (larger is better) on synthetic graphs after removing 1%, 2%, and 3% of nodes. Strategies are (column) grouped in performance tiers. `NB` *and* `XNB` *have the best performances.*

	`coreHD`	`NS`	`CI`	`Xdeg`	`NB`	`XNB`
1%	62.76	61.44	62.88	62.90	62.92	62.91
BA 2%	68.84	66.94	68.97	68.99	69.01	69.01
3%	72.42	70.09	72.56	72.57	72.59	72.59
1%	6.28	6.40	6.41	6.45	6.46	6.46
BTER 2%	10.61	10.72	10.80	10.85	10.86	10.86
3%	14.31	14.40	14.55	14.61	14.63	14.63
1%	3.31	3.41	3.40	3.43	3.44	3.44
SBM 2%	6.00	6.16	6.19	6.23	6.25	6.25
3%	8.52	8.66	8.76	8.80	8.82	8.82
1%	1.41	1.17	1.50	1.52	1.63	1.63
WS 2%	2.52	2.09	2.97	2.98	3.11	3.11
3%	3.66	2.94	4.41	4.41	4.57	4.58

the same time, so `core` cannot identify which is the best one among all of them. In all data sets, `coreHD` and `degree` had very similar performance. Therefore, we only show `coreHD` in our results. We hypothesize this is because `coreHD` only differs from `degree` when the graph has many nodes outside the 2-core, or when removing a single node reduces the size of the 2-core considerably, which is not the case in our data sets.

Table 1 shows the percentage reduction of the eigenvalue after immunization, averaged over repetitions on synthetic graphs. We can arrange immunization strategies in tiers according to increasing performance: strategies within a tier have comparable performance across data

Table 2

Average percentage eigendrop on real networks (larger is better) when removing $p = 1, 10,$ or 100 nodes. A performance of 0.00 means the eigendrop is less than 10^{-2}. Details about these data sets are in Table 3.

	$p = 1$			$p = 10$			$p = 100$		
	coreHD	CI	Xdeg	coreHD	CI	Xdeg	coreHD	CI	Xdeg
AS-1	0.74	0.74	**2.35**	10.13	13.51	**15.43**	74.71	**78.26**	75.92
AS-2	2.02	2.02	**4.00**	26.13	22.36	**28.17**	**90.09**	89.61	87.02
Social-Slashdot	0.95	**1.02**	**1.02**	4.63	6.06	**6.94**	24.14	28.11	**30.30**
Social-Twitter	**2.18**	**2.18**	1.98	13.21	**13.97**	13.68	41.13	42.88	**43.39**
Transport-California	0.00	0.00	**0.65**	**2.65**	0.65	**2.65**	5.09	5.09	**7.80**
Transport-Sydney	**0.00**	**0.00**	**0.00**	0.00	0.00	**6.50**	7.71	7.37	**9.49**
Web-NotreDame	**9.34**	**9.34**	**9.34**	12.10	**13.79**	**13.79**	14.37	14.37	**19.22**

sets. The third tier is made up of NS and coreHD. They perform similarly because NS targets the largest eigenvalue of the adjacency matrix, which is largely dominated by node degree. Strategies in this tier perform substantially better than core (not shown), and are very close to the strategies in the next two tiers, i.e., coreHD is a very strong baseline in this task. The second tier is comprised of CI and Xdeg, with Xdeg having a slight advantage over CI. Finally, the best performance was achieved by NB and XNB. Their performances were almost indistinguishable in most data sets, though they have a small margin over CI and Xdeg.

Strategies in the best two tiers, i.e., CI, Xdeg, NB, and XNB, all showed standard deviations of similar magnitude across all data sets (not shown), and the ordering in increasing performance CI $<$ Xdeg $<$ NB $\approx$ XNB is statistically significant at $p \ll 10^{-10}$ (see Appendix C.5). Further, the best two (NB and XNB) use the principal NB-eigenvector, whereas CI and Xdeg depend only on node degree, and are therefore much more computationally efficient.

Table 2 shows the results on real data sets, where we have run only degree, CI, and Xdeg for computational reasons. We use social networks [14, 28], transportation networks, [35, 43, 28], Autonomous Systems (AS) of the Internet networks [46, 22], and web crawl networks [2]. See section C.2 for data set descriptions. We remove from each network 1, 10, and 100 nodes at a time. Again, degree is a very strong baseline, but it is never better than both CI and Xdeg at the same time. All three strategies are able to drastically immunize the autonomous systems networks AS-1 and AS-2 at 100 nodes removed, probably owing to the fact that their degree distribution is extremely heterogeneous and thus the nodes with largest degree have a large eigendrop. In all other networks, X-degree achieves the best performance. An interesting case is that of Transport-Sydney. The node identified by all three strategies has an eigendrop of exactly 0.0. Following Corollary 3.3, this means that the chosen node lies outside of the 2-core of the graph and thus has no impact on nonzero NB-eigenvalues. After 10 nodes are removed, both degree and CI continue to achieve zero eigendrop, while Xdeg already identifies the correct nodes and achieves 6.50% decrease. Even at 100 nodes removed, degree cannot identify nodes that generate an eigendrop. A similar case occurs on Transport-California, where the first node identified by degree and CI generates no eigendrop, while Xdeg is able to correctly identify influential nodes.

We conclude that in cases where efficiency is of the essence, Xdeg is the best overall immunization strategy, as it has a slight advantage over CI and its performance is close to optimal. If effectiveness is more important than efficiency, either XNB or NB should be used.

7. Conclusion.

We developed a theory of spectral analysis for the NB-matrix by studying what happens to its largest eigenvalue when one node is removed from the network. Our theory is independent of the structure of the graph, i.e., we make no assumptions of locally tree-like structure or density or length of cycles, as is usual in other studies. We find two new node-level statistics, or centrality measures, X-NB centrality and X-degree, which are excellent predictors of a node's influence on the largest NB-eigenvalue. Finally, we focused on the application of targeted immunization, for which we propose two new algorithms that are shown to be more effective than others on a variety of real and synthetic graphs. Code for these algorithms may be found at https://github.com/leotrs/inbox. Many other applications exist. Further studying the behavior of NB-eigenvalues under small perturbations of the graph using the framework presented here has potential to affect those applications.

Our techniques open many possibilities for further research. For instance, the left-hand side of (3.5) is reminiscent of certain quantities used in the theory of eigenvalue interlacing [18], while the matrix $(B' - tI)^{-1}$ on the right-hand side is known as the *resolvent* of B', which has many applications in random matrix theory [41]. On a different note, Cvetkovic, Doob, and Sachs [13] highlight that most matrices associated to graphs are *linear* combinations of I, A, and D, whereas the NB-matrix is associated with a *quadratic* combination of I, A, and D, via the celebrated Ihara–Bass formula [8, 26] (see also (A.1) in Appendix A). In the future, we will explore which other matrices associated with graphs can be studied via quadratic, or higher order, combinations of I, A, and D.

Appendix A. Technical lemmas.

Let B be the NB-matrix of a graph G, P be as defined in section 2, and $\lambda, \mathbf{v}$ be the Perron eigenvalue and corresponding unit right eigenvector of B.

Lemma A.1. *$P\mathbf{v}$ is a left eigenvector of B corresponding to λ.*

Proof. Since PB is symmetric and $P^2 = I$, we have $B = PB^TP$. Now $B\mathbf{v} = \lambda\mathbf{v}$ implies $B^TP\mathbf{v} = \lambda P\mathbf{v}$, which completes the proof. ■

Now let $D = \operatorname{diag}(A\mathbf{1})$ be the diagonal degree matrix, and let $\mathbf{v}_{aux}$ be the left principal eigenvector of

$$B_{aux} = \begin{pmatrix} 0 & D - I \\ -I & A \end{pmatrix}. \tag{A.1}$$

It is known that $\mathbf{v}_{aux} = (\mathbf{f}, -\lambda\mathbf{f})$, where $\mathbf{f}$ is of size n, and it is parallel to the vector of NB-centralities, i.e., $\mathbf{f}^i \propto \mathbf{v}^i$ [30], where $\mathbf{v}^i$ is defined in (2.2). Note it is more efficient to use B_{aux} than B when computing the NB-centrality, since the former is a $2n \times 2n$ matrix with $O(n\langle k\rangle)$ nonzero entries, while the latter is a $2m \times 2m$ matrix with $O(n\langle k^2\rangle)$ nonzero entries [42]. (Here, $\langle k^r\rangle$ is the rth moment of the degree distribution of the graph.)

Lemma A.2. *Suppose $\mathbf{v}$ is such that $\mathbf{v}^TP\mathbf{v} = 1$. Let $\mathbf{v}_{aux} = (\mathbf{f}, -\lambda\mathbf{f})$ be the left unit eigenvector of B_{aux} corresponding to λ, and let $\overline{\mathbf{v}} = \left(\mathbf{v}^1, \dots, \mathbf{v}^n\right)$ be the vector of NB-centralities. Then, we have $\|\overline{\mathbf{v}}\| = \mu\|\mathbf{f}\|$, where*

$$\mu = \sqrt{\frac{\lambda\left(\lambda^2 - 1\right)}{1 - \mathbf{f}^T D\,\mathbf{f}}}. \tag{A.2}$$

Proof. First, from $\mathbf{v}^T P\mathbf{v} = 1$ we get $\mathbf{v}^T PB\mathbf{v} = \lambda\mathbf{v}^T P\mathbf{v} = \lambda$, and we can expand $\mathbf{v}^T PB\mathbf{v}$ to find $\|\overline{\mathbf{v}}\|^2 - \|\mathbf{v}\|^2 = \lambda$. Second, since $(\mathbf{f}, -\lambda\mathbf{f})$ has unit length, we have $\|\mathbf{f}\|^2 = 1/\left(\lambda^2 + 1\right)$. Therefore,

$$\mu^2 = \left(\lambda^2 + 1\right)\left(\lambda + \|\mathbf{v}\|^2\right). \tag{A.3}$$

Now, $B\mathbf{v} = \lambda\mathbf{v}$ implies $\mathbf{v}_{j\to i} + \lambda\mathbf{v}_{i\to j} = \mathbf{v}^i$ for any i, j. Plug this identity in $\|\mathbf{v}\|^2 = \sum_{i,j} a_{ij}\mathbf{v}^2_{i\to j}$ to find

$$\|\mathbf{v}\|^2\left(\lambda^2 + 1\right) + 2\lambda = \sum_i \left(\mathbf{v}^i\right)^2 \deg i = \overline{\mathbf{v}}^T D\overline{\mathbf{v}} = \mu^2\mathbf{f}^T D\,\mathbf{f}. \tag{A.4}$$

Using (A.3) and (A.4) together finishes the proof. ■

Remark A.3. Both $\overline{\mathbf{v}}$ and $\mathbf{f}$ determine the same node centrality ranking, though the latter is easier to compute. However, the normalization $\mathbf{v}^T P\mathbf{v} = 1$ is fundamental in our theory, which makes $\overline{\mathbf{v}}$ the correct choice. Lemma A.2 allows us to compute $\overline{\mathbf{v}}$ only with the knowledge of $\mathbf{f}, \lambda$, and D, which is much more efficient than computing $\mathbf{v}$ and $\overline{\mathbf{v}}$ directly.

Appendix B. Complexity analysis of Algorithm 3.

In section 4.2 we used a standard map (i.e., hash table or dictionary) to store the X-degree values in line 2 of Algorithm 3. Alternatively, we can use an indexed priority queue (IPQ). An IPQ is a data structure that behaves like a priority queue except that, additionally, elements in the IPQ can be updated efficiently. The underlying data structure is a max-heap. An IPQ can find the maximum element in the heap, as well as update any element, in logarithmic time.

In this case, line 2 of Algorithm 3 takes m operations to compute the X-degree values plus n operations to heapify the IPQ. Further, lines 4 and 9 take $O(\log n)$ time, which yields a time complexity of

$$O\left(m + n + p\log n + \sum_{j=1}^{p} d_{i_j} + \log n\sum_{j=1}^{p} D_{i_j}\right). \tag{B.1}$$

B.1. Homogeneous degree distribution.

In networks with a homogeneous degree distribution (e.g., Poisson) we can estimate $d_{i_j} \approx \langle k\rangle$ and $D_{i_j} \approx \langle k\rangle^2$, where $\langle k\rangle$ is the average degree. This yields $O\left(m + n + p\langle k\rangle^2 \log n\right)$ total complexity for the IPQ version, while the map version gives $O\left(m + pn + p\langle k\rangle^2\right)$. If $p = O(n)$ and $\langle k\rangle = O(1)$, the IPQ version scales better in the worst-case scenario.

B.2. Heterogeneous degree distribution.

In networks whose degree distribution is well approximated by a power law, the probability of finding a node of degree d scales as $d^{-\gamma}$ for some $\gamma > 0$. In this case, the first few nodes removed by Algorithm 3 will usually have large degree, comparable to the largest degree in the network, $d_{i_j} = O\left(d_{\max}\right)$ for each j. Further, in the worst-case scenario, each of their neighbors will also have a degree comparable to $d_{\max}$ and thus $D_{i_j} = O\left(d^2_{\max}\right)$ for each j. Using $d_{\max} = O\left(n^{\frac{1}{\gamma-1}}\right)$ [7] yields $O\left(m + pn + pn^{\frac{2}{\gamma-1}}\right)$ for the map version and $O\left(m + pn^{\frac{2}{\gamma-1}}\log n\right)$ for the IPQ version. In the typical case $2 \le \gamma \le 3$, the exponent $\frac{2}{\gamma-1}$ varies between 1 and 2.

Table 3

Real-world data sets. n: number of nodes, m: number of edges, λ_1: largest NB-eigenvalue, $d_{\max}$: largest degree. AS stands for autonomous systems.

	Nodes	Edges	n	m	λ_1	$d_{\max}$
`AS-1` [46]	AS	digital communication	34,761	107,720	151.442	2,760
`AS-2` [22]	AS	digital communication	22,963	48,436	64.678	2,390
`Social-Slashdot` [28]	users	friendships	77,360	469,180	128.550	2,539
`Social-Twitter` [14]	users	friendships	456,290	12,508,221	636.147	51,386
`Transport-California` [43]	intersections	roads	1,957,027	2,760,388	3.321	12
`Transport-Sydney` [43]	intersections	roads	32,956	38,787	2.266	10
`Web-NotreDame` [2]	websites	hyperlinks	325,729	1,090,108	175.657	10,721

B.3. Average runtime. We have provided the analysis of worst-case scenario runtime. However, the average runtime of both the IPQ and map versions is close to linear, as shown in Figure 2. This figure was generated by first sampling a degree sequence from a power-law density $p_d \propto d^{-\gamma}$, and then generating a graph at random using the configuration model. Self-loops and multiedges were removed and only the largest component was kept. Each marker is the average of 30 repetitions. We used $p = 100$.

Appendix C. Experimental setup.

C.1. Base lines.

Degree. The degree of a node i, denoted d_i is the number of neighbors it has in the graph. Nodes of degree 1 have zero CI, X-degree, NB-centrality, and X-NB centrality.

k-core index. The k-core index of a node is defined as follows. First, iteratively remove all nodes of degree 1 until there are none. All nodes removed in this step are assigned a value of k-core index of 1. Then, iteratively remove all nodes of degree 2; all nodes removed at this step have k-core 2. Repeat this process until there are no more nodes in the graph. Notably, following Corollary 3.3, all nodes with k-core value of 1 have zero NB-centrality.

NB-centrality. The NB-centrality of a node is defined in (2.2). It was proposed in [38] as an indicator of influential spreaders on locally tree-like networks for the SIR model.

NetShield. NetShield is an efficient algorithm that identifies a subset of nodes with the highest "shield-value," which is defined as the impact a node, or set of nodes, has on the largest eigenvalue of the adjacency matrix [11].

Collective Influence. The CI of i is $CI_i = (d_i - 1) \sum_j a_{ij} (d_j - 1)$, though this definition can be generalized to include nodes in arbitrarily large neighborhoods around i [32]. Note that this is quite similar in nature to X-degree in (3.15). We think of X-degree as a second-order aggregation of the values $(d_j - 1)$ of the neighbors of i, while CI is a first-order aggregation. We believe this similarity is by no means an accident, and will be the subject of future studies. Further, one can apply Algorithm 3 to perform targeted immunization based on CI instead of X-degree, and hence they have the same running time complexity (see section 4.2 and Appendix B). Morone et al. [33] claim that the CI algorithm runs in $O(n \log n)$ time, though we were not able to reproduce this result. In any case, any efficient algorithm that computes CI can be used to compute X-degree as well.

coreHD. In [45], the authors present the algorithm `coreHD` for dismantling a network in many disconnected components. `coreHD` iteratively removes from the network the node with highest degree inside the 2-core of the network. Edges that lie outside of the 2-core or that connect the 2-core with its complement are ignored.

C.2. Data sets. All synthetic graphs have $n = 10^5$ nodes, and parameters were chosen so that the average degree was approximately 12. SBM graphs were generated with two blocks, or communities, so that the average within-block degree is 9 and the between-block degree is 3. WS graphs were generated with rewiring probability 0.1. BTER graphs were generated with target average local clustering coefficient of 0.98, and target global clustering coefficient of 0.4. BTER graphs were generated with the authors' implementation [25]; all other graphs were generated using NetworkX [19] version 2.3. After generation, we extracted the largest connected component of each graph and converted all multiedges to single edges and deleted self-loops. One hundred graphs were generated from each ensemble. Table 3 describes the real data sets used. Directed networks were converted to undirected, and only the largest connected component of each data set was used.

C.3. Approximating the largest eigenvalue. Since nodes of large degree are bound to induce a larger eigendrop than those of small degree, we chose target nodes at random by sampling 1% of nodes from each graph, proportionally to their degree. This was achieved by sampling one edge at random, with replacement, and then choosing one of its endpoints randomly. This yields a probability of sampling node i equal to $d_i/2m$.

C.4. Predicting the eigendrop. Nodes were sampled in the same way as in C.3. Figure 4 shows correlation coefficients, defined as the covariance divided by the product of the standard deviations of the two variables. We computed the correlation between the eigendrop and each of the statistics: α, $\widetilde{\alpha}$, X-degree, and degree. No three-way correlation was computed.

C.5. Immunization with X-NB and X-degree. To confirm the ordering in increasing performance `CI` < `Xdeg` < `NB` ≈ `XNB`, we used a one-sided Wilcoxon signed-rank test, which is a nonparametric version of the paired T-test. In a paired sample setting, this test tests the null hypothesis that the median of the differences between the two samples is positive, against the alternative that it is negative. Therefore, a small p-value means that there is little probability that the first sample's median is smaller than the second's. For each graph ensemble and each percentage of removed nodes (1%, 2%, 3%), the ranking `CI` < `Xdeg` < `NB` was confirmed with $p \ll 10^{-10}$ in all cases. Further, we have `NB` < `XNB` in WS networks ($p \ll 10^{-10}$) and BTER networks ($p < 0.05$), and `NB` > `XNB` in BA networks ($p \ll 10^{-10}$) and SBM networks ($p < 0.05$). We summarize these results by writing `CI` < `Xdeg` < `NB` ≈ `XNB`.

Acknowledgment. The first author thanks Gabor Lippner for many invaluable discussions.

REFERENCES

[1] R. Albert and A.-L. Barabási, *Statistical mechanics of complex networks*, Rev. Mod. Phys., 74 (2002), pp. 47–97.

[2] R. Albert, H. Jeong, and A.-L. Barabási, *Diameter of the world-wide web*, Nature, 401 (1999), pp. 130–131.

[3] N. Alon, I. Benjamini, E. Lubetzky, and S. Sodin, *Non-backtracking random walks mix faster*, Commun. Contemp. Math., 9 (2007), pp. 585–603.

[4] F. Arrigo, P. Grindrod, D. J. Higham, and V. Noferini, *Non-backtracking walk centrality for directed networks*, J. Complex Networks, 6 (2018), pp. 54–78.

[5] F. Arrigo, P. Grindrod, D. J. Higham, and V. Noferini, *On the exponential generating function for non-backtracking walks*, Linear Algebra Appl., 556 (2018), pp. 381–399.

[6] F. Arrigo, D. J. Higham, and V. Noferini, *Non-backtracking pagerank*, J. Sci. Comput., 80 (2019), pp. 1419–1437.

[7] A.-L. Barabási, *Network Science*, Cambridge University Press, 2016.

[8] H. Bass, *The Ihara-Selberg zeta function of a tree lattice*, Internat. J. Math., 3 (1992), pp. 717–797.

[9] C. Bordenave, M. Lelarge, and L. Massoulié, *Non-backtracking spectrum of random graphs: Community detection and non-regular Ramanujan graphs*, in Proceedings of the 2015 IEEE 56th Annual Symposium on Foundations of Computer Science–FOCS 2015, IEEE, Los Alamitos, CA, 2015, pp. 1347–1357.

[10] C. Castellano and R. Pastor-Satorras, *Relevance of backtracking paths in recurrent-state epidemic spreading on networks*, Phys. Rev. E, 98 (2018), 052313.

[11] C. Chen, H. Tong, B. A. Prakash, C. E. Tsourakakis, T. Eliassi-Rad, C. Faloutsos, and D. H. Chau, *Node immunization on large graphs: Theory and algorithms*, IEEE Trans. Knowl. Data Eng., 28 (2016), pp. 113–126.

[12] S. Coste and Y. Zhu, *Eigenvalues of the Non-backtracking Operator Detached from the Bulk*, preprint, https://arxiv.org/abs/1907.05603, 2019.

[13] D. M. Cvetkovic, M. Doob, and H. Sachs, *Spectra of Graphs: Theory and Application*, Pure and Applied Mathematics 87, Academic Press, New York, 1980.

[14] M. De Domenico, A. Lima, P. Mougel, and M. Musolesi, *The anatomy of a scientific rumor*, Sci. Rep., 3 (2013), 2980.

[15] C. Durfee and K. Martin, *Distinguishing graphs with zeta functions and generalized spectra*, Linear Algebra Appl., 481 (2015), pp. 54–82.

[16] M. G. Everett and S. P. Borgatti, *Induced, endogenous and exogenous centrality*, Social Networks, 32 (2010), pp. 339–344.

[17] M. Girvan and M. E. Newman, *Community structure in social and biological networks*, Proc. Natl. Acad. Sci. USA, 99 (2002), pp. 7821–7826.

[18] C. Godsil and G. F. Royle, *Algebraic Graph Theory*, Grad. Texts Math. 207, Springer, 2013.

[19] A. Hagberg, P. Swart, and D. A. Schult, *Exploring Network Structure, Dynamics, and Function Using NetworkX*, Tech. report, Los Alamos National Lab (LANL), Los Alamos, NM, 2008.

[20] K. E. Hamilton and L. P. Pryadko, *Tight lower bound for percolation threshold on an infinite graph*, Phys. Rev. Lett., 113 (2014), 208701.

[21] B. Karrer and M. E. Newman, *Stochastic blockmodels and community structure in networks*, Phys. Rev. E, 83 (2011), 016107.

[22] B. Karrer, M. E. Newman, and L. Zdeborová, *Percolation on sparse networks*, Phys. Rev. Lett., 113 (2014), 208702.

[23] M. Kempton, *Non-backtracking random walks and a weighted Ihara's theorem*, Open J. Discrete Math., 6 (2016), pp. 207–226.

[24] M. Kitsak, L. K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. E. Stanley, and H. A. Makse, *Identification of influential spreaders in complex networks*, Nat. Phys., 6 (2010), pp. 888–893.

[25] T. G. Kolda, A. Pinar, T. D. Plantenga, and C. Seshadhri, *A scalable generative graph model with community structure*, SIAM J. Sci. Comput., 36 (2014), pp. C424–C452, https://doi.org/10.1137/130914218.

[26] M. Kotani and T. Sunada, *Zeta functions of finite graphs*, J. Math. Sci. Univ. Tokyo, 7 (2000), pp. 7–25.

[27] F. Krzakala, C. Moore, E. Mossel, J. Neeman, A. Sly, L. Zdeborová, and P. Zhang, *Spectral redemption in clustering sparse networks*, Proc. Natl. Acad. Sci. USA, 110 (2013), pp. 20935–20940.

[28] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, *Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters*, Internet Math., 6 (2009), pp. 29–123.

[29] Y. Lin, W. Chen, and Z. Zhang, *Assessing percolation threshold based on high-order non-backtracking*

matrices, in Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 223–232.

[30] T. Martin, X. Zhang, and M. E. J. Newman, *Localization and centrality in networks*, Phys. Rev. E, 90 (2014), 052808.

[31] A. Mellor and A. Grusovin, *Graph comparison via the nonbacktracking spectrum*, Phys. Rev. E, 99 (2019), 052309.

[32] F. Morone and H. A. Makse, *Influence maximization in complex networks through optimal percolation*, Nature, 524 (2015), p. 65.

[33] F. Morone, B. Min, L. Bo, R. Mari, and H. A. Makse, *Collective influence algorithm to find influencers via optimal percolation in massively large social media*, Sci. Rep., 6 (2016), 30062.

[34] M. E. Newman, *Spread of epidemic disease on networks*, Phys. Rev. E, 66 (2002), 016128.

[35] T. Opsahl, *Why Anchorage Is Not (That) Important: Binary Ties and Sample Selection*, available online at https://toreopsahl.com/datasets/#usairports, 2011.

[36] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, and A. Vespignani, *Epidemic processes in complex networks*, Rev. Mod. Phys., 87 (2015), pp. 925–979.

[37] G. Poux-Médard, R. Pastor-Satorras, and C. Castellano, *Influential Spreaders for Recurrent Epidemics on Networks*, preprint, https://arxiv.org/abs/1912.08459, 2019.

[38] F. Radicchi and C. Castellano, *Leveraging percolation theory to single out influential spreaders in networks*, Phys. Rev. E, 93 (2016), 062314.

[39] C. Seshadhri, T. G. Kolda, and A. Pinar, *Community structure and scale-free collections of Erdős-Rényi graphs*, Phys. Rev. E, 85 (2012), 056109.

[40] M. Shrestha, S. V. Scarpino, and C. Moore, *Message-passing approach for recurrent-state epidemic models on networks*, Phys. Rev. E, 92 (2015), 022821.

[41] T. Tao, *The Semi-Circular Law*, available online at https://terrytao.wordpress.com/2010/02/02/254a-notes-4-the-semi-circular-law/, 2010.

[42] L. Torres, P. Suárez-Serrato, and T. Eliassi-Rad, *Non-backtracking cycles: Length spectrum theory and graph mining applications*, Appl. Netw. Sci., 4 (2019), 41.

[43] Transportation Networks for Research Core Team, *Transportation Networks for Research*, available online at https://github.com/bstabler/TransportationNetworks/, 2013.

[44] D. J. Watts and S. H. Strogatz, *Collective dynamics of 'small-world' networks*, Nature, 393 (1998), pp. 440–442.

[45] L. Zdeborová, P. Zhang, and H.-J. Zhou, *Fast and simple decycling and dismantling of networks*, Sci. Rep., 6 (2016), 37954.

[46] B. Zhang, R. A. Liu, D. Massey, and L. Zhang, *Collecting the internet AS-level topology*, Comput. Commun. Rev., 35 (2005), pp. 53–61.

[47] P. Zhang, *Nonbacktracking operator for the Ising model and its applications in systems with multiple states*, Phys. Rev. E, 91 (2015), 042120.