Repeated Active Screening of Networks for Diseases

Biswarup Bhattacharya*, Han Ching Ou*, Arunesh Sinha^, Sze-Chuan Suen*, Bistra Dilkina* & Milind Tambe*
*University of Southern California (USA), ^University of Michigan – Ann Arbor (USA)

Introduction

Broad Objective: Decreasing the number of individuals affected by a recurrent disease in a network.

Problem Statement: Selecting k out of n individuals to screen at every timestep against SEIS type diseases (diseases w/ latent stages & w/o permanent immunity) in a network. [NP-hard]

Contributions:

1. A new model for SEIS type diseases using a novel belief-state representation to model unknown health states.
2. A community and eigenvalue-based algorithm (TRACE) to perform multi-round active screening.

Model

SEIS Disease Model: Example – Tuberculosis (TB).

$$\text{Susceptible } (S) \xrightarrow{\alpha} \text{Exposed } (E)$$
$$\text{Exposed } (E) \xrightarrow{\beta} \text{Infected } (I)$$
$$\text{Infected } (I) \xrightarrow{c} \text{Susceptible } (S)$$

Active Screening Model: n individuals, each in one of the three states (S, E, or I) (unknown) in a known contact network. k actions. Reward of +1 for every S (healthy) individual per round. Transition rules:

$$T^0 = \begin{array}{c c} & \begin{array}{ccc} S & E & I \end{array} \\ \begin{array}{c} S \\ E \\ I \end{array} & \begin{bmatrix} q_j & 1-q_j & 0 \\ 0 & 1-\beta & \beta \\ c & 0 & 1-c \end{bmatrix} \end{array}, \quad T^1 = \begin{array}{c c} & \begin{array}{ccc} S & E & I \end{array} \\ \begin{array}{c} S \\ E \\ I \end{array} & \begin{bmatrix} q_j & 1-q_j & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \end{array},$$

$$\text{and } q_j = (1-\alpha)^{|\{k \in N(j) \mid s_k^t = I\}|}$$

Belief States: A vector $[b_{i,S}, b_{i,E}, b_{i,I}]$ maintained for each node i.
Belief Update: Two step process: $b_i^t \rightarrow \bar{b}_i^t \rightarrow b_i^{t+1}$

Intermediate beliefs updated to:

- [0,0,1] if individual showed up to a clinic voluntarily.
- Actual/true state, if individuals was chosen to be screened.
- Else,

$$\bar{b}_i^t = \frac{[b_{i,S}^t, b_{i,E}^t, (1-c)b_i^t]}{b_{i,S}^t + b_{i,E}^t + (1-c)b_i^t}$$

Final beliefs updated to:

- [1,0,0] if individual showed up to a clinic voluntarily or was screened.
- Else update according to natural transition rules.

$$b_i^{t+1} = \bar{b}_i^t \, \mathbf{\Gamma^t}, \text{ where}$$

$$\mathbf{\Gamma^t} = \begin{bmatrix} w_i^t & 1-w_i^t & 0 \\ 0 & 1-\beta & \beta \\ c & 0 & 1-c \end{bmatrix}, \quad w_i^t = \prod_{j \in N(i)} (1 - \alpha \bar{b}_{j,I}^t).$$

TRACE Algorithm

Key Ideas:

- Maximizing the reduction in the largest eigenvalue (spectral radius) can lead to a reduction in disease prevalence.
- Neither of eigenvalue-based, community-based, or greedy methods are dominant (shown by Observations). Each method has its own set of advantages based on graph type.

Algorithm:

1. Rank and classify the nodes into *types* according to an *attractiveness* score. (lines 1-3)
2. Cluster nodes of same *types* ("coarsening" the graph) to form *supernodes.* (line 4)
3. Pick *supernodes* to screen such that maximal expected reduction in eigenvalue is achieved. (line 5)
4. Make sure that only k nodes are selected at the end for active screening. (lines 6-11)

Due to uncertainty of health states, we estimate benefit by using $b_{i,S}$ as a proxy for probability of node i being healthy.

Algorithm 1 TRACE Algorithm

```
Input: Adjacency Matrix $A$ of graph, Belief $b^t$, Budget $k$
 1: for all $i \in \{1, \dots, n\}$ do
 2:     $R_i^t = \sigma b_{i,E}^t + b_{i,I}^t$
 3: Sort $R^t$ and label each node as $g_1, g_2$, or $g_3$
 4: $\overline{\mathrm{A}}, \overline{\mathrm{b}}^t, size \leftarrow Coarsen(A, g_1, g_2, g_3, b^t)$
 5: $\mathrm{U} \leftarrow$ DynamicEigen($\overline{\mathrm{A}}, \overline{\mathrm{b}}^t, size, k$)
 6: if $\sum_{\mathbf{u} \in \mathrm{U}} size_{\mathbf{u}} > k$ then
 7:     $\mathbf{u}' \leftarrow$ the last selected super-node from U
 8:     $\kappa = k - \sum_{\mathbf{u} \in \mathrm{U} \setminus \mathbf{u}'} size_{\mathbf{u}}$
 9:     $\underline{A}, \underline{b}^t \leftarrow$ remove all nodes in $\mathrm{U} \setminus \mathbf{u}'$ from $A, b^t$
10:     $a \leftarrow$ DynamicEigen($\underline{A}, \underline{b}^t, 1, \kappa$)
11: Active screen nodes $\{v \mid v \in a$ or $v \in \mathbf{u}$ for $\mathbf{u} \in \mathrm{U} \setminus \mathbf{u}'\}$
```

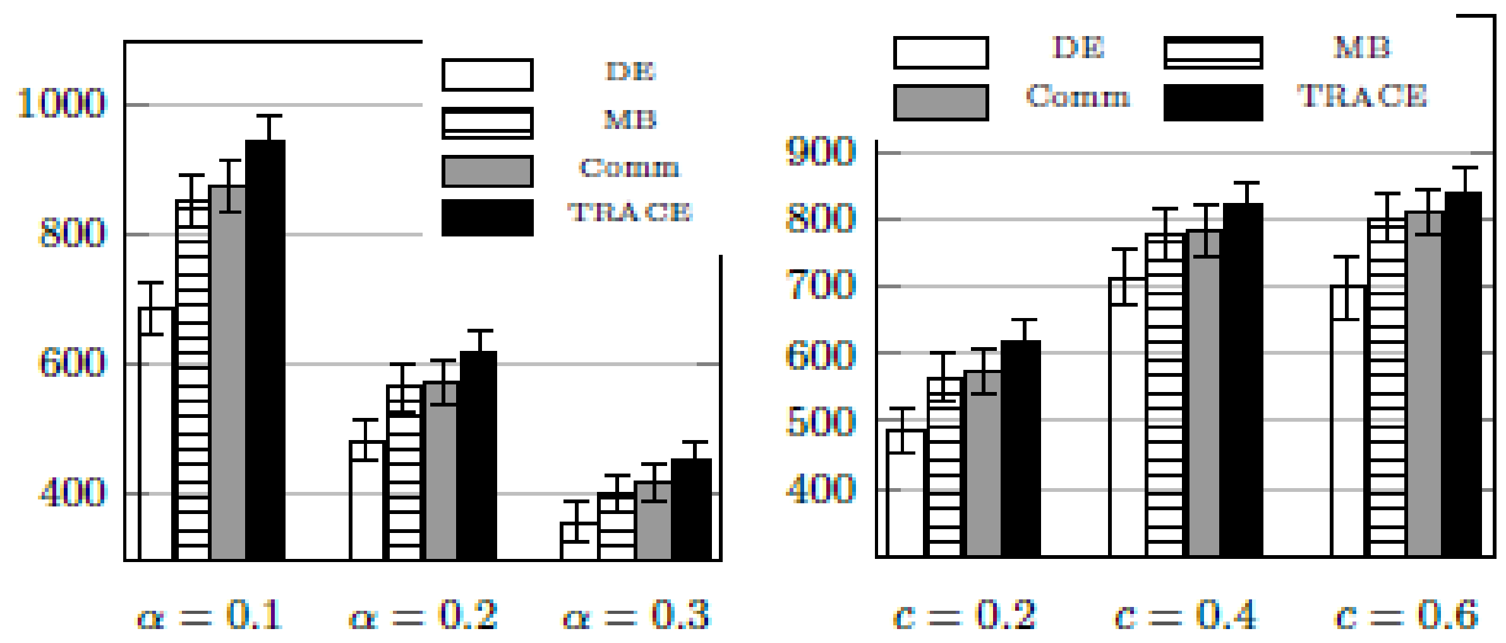
Algorithm 2 DynamicEigen(A, b^t, w, k)

```
Input: Adjacency matrix $A$, belief $b$, function $w$ for weight of each
       node, min total weight of nodes to remove $k$
1: $V \leftarrow$ Number of vertex of input graph
2: for all $i \in \{1, \dots, V\}$ do
3:     $A_{i,:} = A_{i,:} * (1 - b_{i,S})$                ▹ Multiply $i^{th}$ row
4: for all $i \in \{1, \dots, V\}$ do
5:     $A' \leftarrow A$
6:     $\mathrm{A}'_{i,:} \leftarrow \mathbf{0}$, $\mathrm{A}'_{:,i} \leftarrow \mathbf{0}$                ▹ Remove $i^{th}$ node
7:     $\lambda^i = LargestEigenvalue(A')$
8: Sort nodes $\langle v_1, \dots, v_V \rangle$ corresponding to increasing $\lambda^i$
9: return first $h$ nodes such that $\sum_{i=1}^{h} w(v_i) \geq k$
```

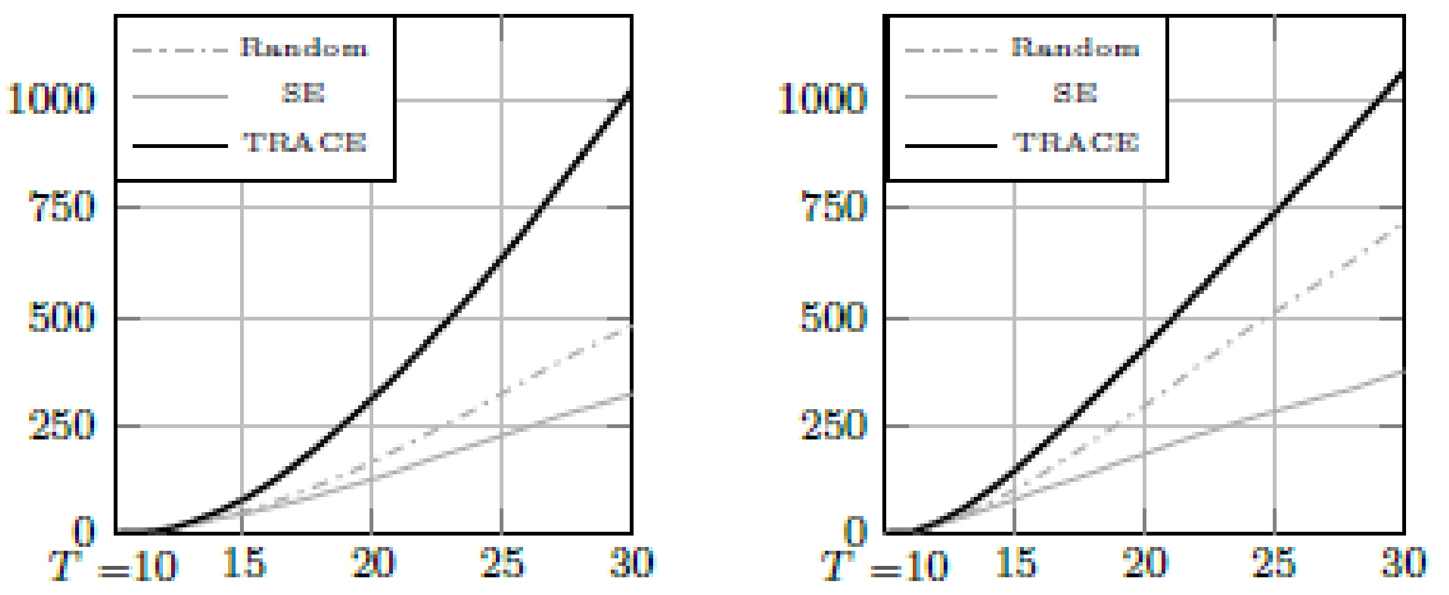
Experiments & Results

- **Networks**: India (202), Exhibition (410), Irvine (1899).
- **Metric**: Increase in number of S individuals over no intervention.
- **Settings**: k=5%, β=0.25, σ=0.5.
- **Setup**: No screening (t=1 to 10), Screening (t=11 to 30).

a) Baselines: *Random*, Eigen (*SE*).
b) Components: *MB* (Choose top k $b_{i,I}$), DynamicEigen/*DE* (no supernodes), *Comm*unity only (0-1 knapsack).



(a) Varying α (c = 0.2)
(b) Varying c (α = 0.2)



(a) India network
(b) Exhibition network