Convex Optimization for Finding Influential Nodes in Social Networks

Stephen Vavasis[1]

[1]Department of Combinatorics & Optimization
University of Waterloo
This talk represents work of myself, L. Elkin, T. K. Pong

2013-Aug-8 / NLO - Overton meeting

Social networks

- Define social networks to be directed graphs, e.g., Twitter posters and followers.
- The network can be either deterministic (arcs pass every message) or probabilistic (arcs flip coins to determine whether to pass messages).
- Maximizing influence means finding the set of nodes (with a constraint or cost on the size of the set) that reaches the largest number of nodes through the network. Many variations of the problem.

Social networks

- Define social networks to be directed graphs, e.g., Twitter posters and followers.
- The network can be either deterministic (arcs pass every message) or probabilistic (arcs flip coins to determine whether to pass messages).
- Maximizing influence means finding the set of nodes (with a constraint or cost on the size of the set) that reaches the largest number of nodes through the network. Many variations of the problem.

Social networks

- Define social networks to be directed graphs, e.g., Twitter posters and followers.
- The network can be either deterministic (arcs pass every message) or probabilistic (arcs flip coins to determine whether to pass messages).
- Maximizing influence means finding the set of nodes (with a constraint or cost on the size of the set) that reaches the largest number of nodes through the network. Many variations of the problem.

Social networks

- Define social networks to be directed graphs, e.g., Twitter posters and followers.
- The network can be either deterministic (arcs pass every message) or probabilistic (arcs flip coins to determine whether to pass messages).
- Maximizing influence means finding the set of nodes (with a constraint or cost on the size of the set) that reaches the largest number of nodes through the network. Many variations of the problem.

Applications of maximizing influence

- Influential nodes can be used to spread an important announcement.
- Advertisers wish to reach potential customers.
- Epidemiologists can use social networks to predict spread of a communicable disease.

Applications of maximizing influence

- Influential nodes can be used to spread an important announcement.
- Advertisers wish to reach potential customers.
- Epidemiologists can use social networks to predict spread of a communicable disease.

Applications of maximizing influence

- Influential nodes can be used to spread an important announcement.
- Advertisers wish to reach potential customers.
- Epidemiologists can use social networks to predict spread of a communicable disease.

Applications of maximizing influence

- Influential nodes can be used to spread an important announcement.
- Advertisers wish to reach potential customers.
- Epidemiologists can use social networks to predict spread of a communicable disease.

Influence in numerical analysis

Gene Golub, center, with some of his former students at his 60th Birthday celebration in Minneapolis, 1992. Others are, from left, Oliver Ernst, Alan George, Franklin Luk, Jim Varah, Eric Grosse, Petter Bjørstad, Margaret Wright, Gene Golub, Dan Boley, Dianne O'Leary, Steve Vavasis, Tony Chan, Bill Coughran, Michael Heath, Michael Overton, and Nick Trefethen.

Influence in numerical analysis

Gene Golub, center, with some of his former students at his 60th Birthday celebration in Minneapolis, 1992. Others are, from left, Oliver Ernst, Alan George, Franklin Luk, Jim Varah, Eric Grosse, Petter Bjørstad, Margaret Wright, Gene Golub, Dan Boley, Dianne O'Leary, UNIDENTIFIED, Tony Chan, Bill Coughran, Michael Heath, Michael Overton, and Nick Trefethen.

Unweighted deterministic case

- Input: Digraph $G = (V, \mathcal{A})$ and integer k.
- Output: The subset $V^* \subset V$, $|V^*| = k$ that maximizes the number of vertices V' such that there exists a directed path from a node in V^* to a node in V'
- Deterministic case not widely used in literature but serves as warm-up for probabilistic case. But still NP-hard.

Unweighted deterministic case

- Input: Digraph $G = (V, \mathcal{A})$ and integer k.
- Output: The subset $V^* \subset V$, $|V^*| = k$ that maximizes the number of vertices V' such that there exists a directed path from a node in V^* to a node in V'
- Deterministic case not widely used in literature but serves as warm-up for probabilistic case. But still NP-hard.

Unweighted deterministic case

- Input: Digraph $G = (V, \mathcal{A})$ and integer k.
- Output: The subset $V^* \subset V$, $|V^*| = k$ that maximizes the number of vertices V' such that there exists a directed path from a node in V^* to a node in V'
- Deterministic case not widely used in literature but serves as warm-up for probabilistic case. But still NP-hard.

Unweighted deterministic case

- Input: Digraph $G = (V, \mathcal{A})$ and integer k.
- Output: The subset $V^* \subset V$, $|V^*| = k$ that maximizes the number of vertices V' such that there exists a directed path from a node in V^* to a node in V'
- Deterministic case not widely used in literature but serves as warm-up for probabilistic case. But still NP-hard.

IP formulation–setup

- Let A be the $|V| \times |V|$ matrix of 0's and 1's such that $A(i,j) = 1$ iff there is a directed path from i to j.
- Let $\mathbf{x}$ be the $0-1$ decision vector that indicates membership in the influential subset, and let $\mathbf{t}$ be the $0-1$ vector that indicates whether a node is reached by the influential subset.
- Let $\mathbf{e}$ denote the vector of all 1's.

IP formulation–setup

- Let A be the $|V| \times |V|$ matrix of 0's and 1's such that $A(i,j) = 1$ iff there is a directed path from i to j.
- Let $\mathbf{x}$ be the $0-1$ decision vector that indicates membership in the influential subset, and let $\mathbf{t}$ be the $0-1$ vector that indicates whether a node is reached by the influential subset.
- Let $\mathbf{e}$ denote the vector of all 1's.

IP formulation–setup

- Let A be the $|V| \times |V|$ matrix of 0's and 1's such that $A(i,j) = 1$ iff there is a directed path from i to j.
- Let $\mathbf{x}$ be the $0-1$ decision vector that indicates membership in the influential subset, and let $\mathbf{t}$ be the $0-1$ vector that indicates whether a node is reached by the influential subset.
- Let $\mathbf{e}$ denote the vector of all 1's.

IP formulation–setup

- Let A be the $|V| \times |V|$ matrix of 0's and 1's such that $A(i,j) = 1$ iff there is a directed path from i to j.
- Let $\mathbf{x}$ be the $0-1$ decision vector that indicates membership in the influential subset, and let $\mathbf{t}$ be the $0-1$ vector that indicates whether a node is reached by the influential subset.
- Let $\mathbf{e}$ denote the vector of all 1's.

Integer LP formulation

$$\begin{aligned} \max \quad & \mathbf{e}^T \mathbf{t} \\ \text{s.t.} \quad & \mathbf{t} \le A^T \mathbf{x}, \\ & \mathbf{t} \le \mathbf{e}, \\ & \mathbf{e}^T \mathbf{x} = k, \\ & \mathbf{x} \in \{0, 1\}^{|V|}. \end{aligned}$$

Convex formulation

$$
\begin{aligned}
\max \quad & \mathbf{e}^T\mathbf{t} \\
\text{s.t.} \quad & \mathbf{t} \leq A^T\mathbf{x}, \\
& \mathbf{t} \leq \mathbf{e}, \\
& \mathbf{e}^T\mathbf{x} = k, \\
& \mathbf{x} \in [0, 1]^{|V|}.
\end{aligned}
$$

When does the convex relaxation solve the original problem?

- Since 2005, many papers have appeared in the optimization literature showing that convex relaxation can solve an NP-hard problem assuming the problem data satisfies certain assumptions.
- Famous example: compressive sensing (Candès, Romberg and Tao; Donoho).

When does the convex relaxation solve the original problem?

- Since 2005, many papers have appeared in the optimization literature showing that convex relaxation can solve an NP-hard problem assuming the problem data satisfies certain assumptions.
- Famous example: compressive sensing (Candès, Romberg and Tao; Donoho).

When does the convex relaxation solve the original problem?

- Since 2005, many papers have appeared in the optimization literature showing that convex relaxation can solve an NP-hard problem assuming the problem data satisfies certain assumptions.
- Famous example: compressive sensing (Candès, Romberg and Tao; Donoho).

Rationale for this style of analysis

- Many NP-hard data-mining problems are routinely solved in practice using heuristics.
- Users are happy with the results. How is this possible?
- Hypothesis: heuristic algorithms succeed because real data has underlying structure that algorithms can recover.
- Leads to consideration of instances in which the sought-after structure is present but hidden by noise.

Rationale for this style of analysis

- Many NP-hard data-mining problems are routinely solved in practice using heuristics.
- Users are happy with the results. How is this possible?
- Hypothesis: heuristic algorithms succeed because real data has underlying structure that algorithms can recover.
- Leads to consideration of instances in which the sought-after structure is present but hidden by noise.

Rationale for this style of analysis

- Many NP-hard data-mining problems are routinely solved in practice using heuristics.
- Users are happy with the results. How is this possible?
- Hypothesis: heuristic algorithms succeed because real data has underlying structure that algorithms can recover.
- Leads to consideration of instances in which the sought-after structure is present but hidden by noise.

Rationale for this style of analysis

- Many NP-hard data-mining problems are routinely solved in practice using heuristics.
- Users are happy with the results. How is this possible?
- Hypothesis: heuristic algorithms succeed because real data has underlying structure that algorithms can recover.
- Leads to consideration of instances in which the sought-after structure is present but hidden by noise.

Rationale for this style of analysis

- Many NP-hard data-mining problems are routinely solved in practice using heuristics.
- Users are happy with the results. How is this possible?
- Hypothesis: heuristic algorithms succeed because real data has underlying structure that algorithms can recover.
- Leads to consideration of instances in which the sought-after structure is present but hidden by noise.

Two-layer specialization

- Assume that graph is bipartite, i.e., $V = S \cup R$, and all arcs are directed from S, the *senders*, to R, the *receivers*.
- This is WLOG: Replace original graph by two copies of V; arcs denoted 'reaches' relation.
- Problem is NP-hard since it is equivalent to the set-cover problem: Given a universe $U = \{1, \ldots, n\}$ of objects and a collection $\mathcal{T}$ of subsets of U, find the subcollection $\mathcal{T}^* \subset \mathcal{T}$ minimizing $|\mathcal{T}^*|$ such that $\bigcup_{T \in \mathcal{T}^*} T = U$.

Two-layer specialization

- Assume that graph is bipartite, i.e., $V = S \cup R$, and all arcs are directed from S, the *senders*, to R, the *receivers*.
- This is WLOG: Replace original graph by two copies of V; arcs denoted 'reaches' relation.
- Problem is NP-hard since it is equivalent to the set-cover problem: Given a universe $U = \{1, \ldots, n\}$ of objects and a collection $\mathcal{T}$ of subsets of U, find the subcollection $\mathcal{T}^* \subset \mathcal{T}$ minimizing $|\mathcal{T}^*|$ such that $\bigcup_{T \in \mathcal{T}^*} T = U$.

Two-layer specialization

- Assume that graph is bipartite, i.e., $V = S \cup R$, and all arcs are directed from S, the *senders*, to R, the *receivers*.
- This is WLOG: Replace original graph by two copies of V; arcs denoted 'reaches' relation.
- Problem is NP-hard since it is equivalent to the set-cover problem: Given a universe $U = \{1, \ldots, n\}$ of objects and a collection $\mathcal{T}$ of subsets of U, find the subcollection $\mathcal{T}^* \subset \mathcal{T}$ minimizing $|\mathcal{T}^*|$ such that $\bigcup_{T \in \mathcal{T}^*} T = U$.

Two-layer specialization

- Assume that graph is bipartite, i.e., $V = S \cup R$, and all arcs are directed from S, the *senders*, to R, the *receivers*.
- This is WLOG: Replace original graph by two copies of V; arcs denoted 'reaches' relation.
- Problem is NP-hard since it is equivalent to the set-cover problem: Given a universe $U = \{1, \ldots, n\}$ of objects and a collection $\mathcal{T}$ of subsets of U, find the subcollection $\mathcal{T}^* \subset \mathcal{T}$ minimizing $|\mathcal{T}^*|$ such that $\bigcup_{T \in \mathcal{T}^*} T = U$.

Generative model: labeling the nodes

- Assume the sender and receiver nodes are partitioned into k *interest groups*: $S = S_1 \cup \cdots \cup S_k$; $R = R_1 \cup \cdots \cup R_k$.
- Assume S_l has exactly one distinguished *influencer*, $l = 1, \ldots, k$.
- Remaining S_l-nodes are called *subordinates.*
- Sought-after IP solution: $x_i = 1$ if i is an influencer; $x_i = 0$ if i is a subordinate.

Generative model: labeling the nodes

- Assume the sender and receiver nodes are partitioned into k *interest groups*: $S = S_1 \cup \cdots \cup S_k$; $R = R_1 \cup \cdots \cup R_k$.
- Assume S_l has exactly one distinguished *influencer*, $l = 1, \ldots, k$.
- Remaining S_l-nodes are called *subordinates*.
- Sought-after IP solution: $x_i = 1$ if i is an influencer; $x_i = 0$ if i is a subordinate.

Generative model: labeling the nodes

- Assume the sender and receiver nodes are partitioned into k *interest groups*: $S = S_1 \cup \cdots \cup S_k$; $R = R_1 \cup \cdots \cup R_k$.
- Assume S_l has exactly one distinguished *influencer*, $l = 1, \ldots, k$.
- Remaining S_l-nodes are called *subordinates*.
- Sought-after IP solution: $x_i = 1$ if i is an influencer; $x_i = 0$ if i is a subordinate.

Generative model: labeling the nodes

- Assume the sender and receiver nodes are partitioned into k *interest groups*: $S = S_1 \cup \cdots \cup S_k$; $R = R_1 \cup \cdots \cup R_k$.
- Assume S_l has exactly one distinguished *influencer*, $l = 1, \ldots, k$.
- Remaining S_l-nodes are called *subordinates.*
- Sought-after IP solution: $x_i = 1$ if i is an influencer; $x_i = 0$ if i is a subordinate.

Generative model: labeling the nodes

- Assume the sender and receiver nodes are partitioned into k *interest groups*: $S = S_1 \cup \cdots \cup S_k$; $R = R_1 \cup \cdots \cup R_k$.
- Assume S_l has exactly one distinguished *influencer*, $l = 1, \ldots, k$.
- Remaining S_l-nodes are called *subordinates.*
- Sought-after IP solution: $x_i = 1$ if i is an influencer; $x_i = 0$ if i is a subordinate.

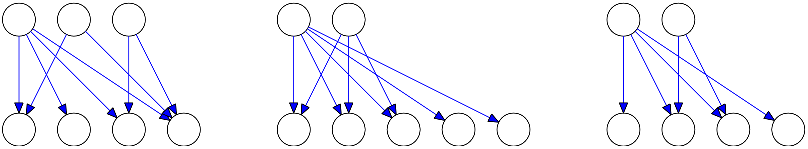
Generative model: choosing edges

Generative model: choosing edges

Simplest case: assume each influencer is connected to every follower in its interest group. Assume every subordinate is connected to a proper subset of followers of its group.

Generative model: choosing edges

Simplest case: assume each influencer is connected to every follower in its interest group. Assume every subordinate is connected to a proper subset of followers of its group.



Theorem for simple deterministic case

- **Theorem.** For this model, sought-after IP solution is the unique solution of the LP relaxation.
- Proof: Use LP duality. Construct a specific dual solution to prove uniqueness.

Theorem for simple deterministic case

- **Theorem.** For this model, sought-after IP solution is the unique solution of the LP relaxation.
- Proof: Use LP duality. Construct a specific dual solution to prove uniqueness.

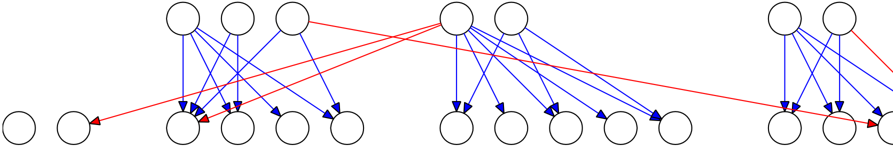
Adding noise to this model

Adding noise to this model

Put in all the edges as in the previous case. Then insert additional random ‘noise’ edges from senders to receivers in other interest groups. Also, include set R_0 of receivers not in an interest group.

Adding noise to this model

Put in all the edges as in the previous case. Then insert additional random 'noise' edges from senders to receivers in other interest groups. Also, include set R_0 of receivers not in an interest group.



Theorem for noise

- **Theorem.** Convex relaxation exactly recovers IP solution with probability exponentially close to 1 assuming certain bounds on the noise.
- Noise arcs and subordinate-receiver arcs inserted at random according to rules to ensure that that each receiver has degree approximately $\sigma \cdot \min_l |S_l|$ where σ a fixed scalar in $(0, 1)$.
- "Exponentially close to 1" means $1 - c_1 \exp(-c_2(\min_l |R_l|))$.

Theorem for noise

- **Theorem.** Convex relaxation exactly recovers IP solution with probability exponentially close to 1 assuming certain bounds on the noise.
- Noise arcs and subordinate-receiver arcs inserted at random according to rules to ensure that that each receiver has degree approximately $\sigma \cdot \min_l |S_l|$ where σ a fixed scalar in $(0, 1)$.
- "Exponentially close to 1" means $1 - c_1 \exp(-c_2(\min_l |R_l|))$.

Theorem for noise

- **Theorem.** Convex relaxation exactly recovers IP solution with probability exponentially close to 1 assuming certain bounds on the noise.
- Noise arcs and subordinate-receiver arcs inserted at random according to rules to ensure that that each receiver has degree approximately $\sigma \cdot \min_l |S_l|$ where σ a fixed scalar in $(0, 1)$.
- "Exponentially close to 1" means $1 - c_1 \exp(-c_2(\min_l |R_l|))$.

Theorem for noise

- **Theorem.** Convex relaxation exactly recovers IP solution with probability exponentially close to 1 assuming certain bounds on the noise.
- Noise arcs and subordinate-receiver arcs inserted at random according to rules to ensure that that each receiver has degree approximately $\sigma \cdot \min_l |S_l|$ where σ a fixed scalar in $(0,1)$.
- "Exponentially close to 1" means $1 - c_1 \exp(-c_2(\min_l |R_l|))$.

Probabilistic graph model

- Each arc is labeled with a probability.
- Each sender i passes its message to a receiver j with specified probability p_{ij} for all $(i, j) \in \mathcal{A}$. Called the *cascade* model by Kempe, Kleinberg and Tardos (2003).
- Problem: given a digraph labeled with probabilities and an integer k, select the k nodes that reach the greatest expected number of followers (expectation over random choices of successful message transmission).
- Algorithm for selecting influential nodes knows the probabilities but not the ultimate coin flips.

Probabilistic graph model

- Each arc is labeled with a probability.
- Each sender i passes its message to a receiver j with specified probability p_{ij} for all $(i,j) \in \mathcal{A}$. Called the *cascade* model by Kempe, Kleinberg and Tardos (2003).
- Problem: given a digraph labeled with probabilities and an integer k, select the k nodes that reach the greatest expected number of followers (expectation over random choices of successful message transmission).
- Algorithm for selecting influential nodes knows the probabilities but not the ultimate coin flips.

Probabilistic graph model

- Each arc is labeled with a probability.
- Each sender i passes its message to a receiver j with specified probability p_{ij} for all $(i, j) \in \mathcal{A}$. Called the *cascade* model by Kempe, Kleinberg and Tardos (2003).
- Problem: given a digraph labeled with probabilities and an integer k, select the k nodes that reach the greatest expected number of followers (expectation over random choices of successful message transmission).
- Algorithm for selecting influential nodes knows the probabilities but not the ultimate coin flips.

Probabilistic graph model

- Each arc is labeled with a probability.
- Each sender i passes its message to a receiver j with specified probability p_{ij} for all $(i, j) \in \mathcal{A}$. Called the *cascade* model by Kempe, Kleinberg and Tardos (2003).
- Problem: given a digraph labeled with probabilities and an integer k, select the k nodes that reach the greatest expected number of followers (expectation over random choices of successful message transmission).
- Algorithm for selecting influential nodes knows the probabilities but not the ultimate coin flips.

Probabilistic graph model

- Each arc is labeled with a probability.
- Each sender i passes its message to a receiver j with specified probability p_{ij} for all $(i, j) \in \mathcal{A}$. Called the *cascade* model by Kempe, Kleinberg and Tardos (2003).
- Problem: given a digraph labeled with probabilities and an integer k, select the k nodes that reach the greatest expected number of followers (expectation over random choices of successful message transmission).
- Algorithm for selecting influential nodes knows the probabilities but not the ultimate coin flips.

Stochastic integer program

- Let $\mathcal{Y}$ denote the finite distribution of possible networks resulting from coin-flips.
- $$\begin{aligned} \max \quad & E[\mathbf{e}^T\mathbf{t} : A \in \mathcal{Y}] \\ \text{s.t.} \quad & \mathbf{t} \leq A^T\mathbf{x}, \\ & \mathbf{t} \leq \mathbf{e}, \\ & \mathbf{e}^T\mathbf{x} = k, \\ & \mathbf{x} \in \{0,1\}^{|V|}. \end{aligned}$$

Stochastic integer program

- Let $\mathcal{Y}$ denote the finite distribution of possible networks resulting from coin-flips.
-

$$
\begin{aligned}
\max \quad & E[\mathbf{e}^T\mathbf{t} : A \in \mathcal{Y}] \\
\text{s.t.} \quad & \mathbf{t} \leq A^T\mathbf{x}, \\
& \mathbf{t} \leq \mathbf{e}, \\
& \mathbf{e}^T\mathbf{x} = k, \\
& \mathbf{x} \in \{0,1\}^{|V|}.
\end{aligned}
$$

Stochastic integer program

- Let $\mathcal{Y}$ denote the finite distribution of possible networks resulting from coin-flips.
-
$$
\begin{aligned}
\max \quad & E[\mathbf{e}^T\mathbf{t} : A \in \mathcal{Y}] \\
\text{s.t.} \quad & \mathbf{t} \leq A^T\mathbf{x}, \\
& \mathbf{t} \leq \mathbf{e}, \\
& \mathbf{e}^T\mathbf{x} = k, \\
& \mathbf{x} \in \{0,1\}^{|V|}.
\end{aligned}
$$

Two-layer assumption

- For the probabilistic model, the two-layer assumption (apparently) entails a loss of generality.
- Nonetheless, it is still somewhat realistic: Bakshy et al. (2011) find that cascades on Twitter are rarely deeper than 1.

Two-layer assumption

- For the probabilistic model, the two-layer assumption (apparently) entails a loss of generality.
- Nonetheless, it is still somewhat realistic: Bakshy et al. (2011) find that cascades on Twitter are rarely deeper than 1.

Two-layer assumption

- For the probabilistic model, the two-layer assumption (apparently) entails a loss of generality.
- Nonetheless, it is still somewhat realistic: Bakshy et al. (2011) find that cascades on Twitter are rarely deeper than 1.

Explicit calculation of expected value

- In the two-layer case, the expected value can be computed in closed form.
- $E[\mathbf{e}^T\mathbf{t}] = \sum_j E[t_j]$.
- t_j is indicator if any arc incoming to j is chosen.
- Hence $E[t_j] = 1 - \prod_{(i,j)\in\mathcal{A}}(1 - p_{ij})^{x_i}$.

Explicit calculation of expected value

- In the two-layer case, the expected value can be computed in closed form.
- $E[\mathbf{e}^T\mathbf{t}] = \sum_j E[t_j]$.
- t_j is indicator if any arc incoming to j is chosen.
- Hence $E[t_j] = 1 - \prod_{(i,j)\in\mathcal{A}}(1 - p_{ij})^{x_i}$.

Explicit calculation of expected value

- In the two-layer case, the expected value can be computed in closed form.
- $E[\mathbf{e}^T\mathbf{t}] = \sum_j E[t_j]$.
- t_j is indicator if any arc incoming to j is chosen.
- Hence $E[t_j] = 1 - \prod_{(i,j)\in\mathcal{A}}(1 - p_{ij})^{x_i}$.

Explicit calculation of expected value

- In the two-layer case, the expected value can be computed in closed form.
- $E[\mathbf{e}^T\mathbf{t}] = \sum_j E[t_j]$.
- t_j is indicator if any arc incoming to j is chosen.
- Hence $E[t_j] = 1 - \prod_{(i,j)\in\mathcal{A}}(1 - p_{ij})^{x_i}$.

Explicit calculation of expected value

- In the two-layer case, the expected value can be computed in closed form.
- $E[\mathbf{e}^T\mathbf{t}] = \sum_j E[t_j]$.
- t_j is indicator if any arc incoming to j is chosen.
- Hence $E[t_j] = 1 - \prod_{(i,j)\in\mathcal{A}}(1 - p_{ij})^{x_i}$.

Rewritten stochastic IP

$$\begin{array}{ll} \min & \sum_{j \in R} \prod_{(i,j) \in \mathcal{A}} (1 - p_{ij})^{x_i} \\ \text{s.t.} & \mathbf{e}^T \mathbf{x} = k, \\ & \mathbf{x} \in \{0,1\}^{|S|}. \end{array}$$

Convex relaxation

$$\begin{array}{ll} \min & \sum_{j \in R} \prod_{(i,j) \in \mathcal{A}} (1 - p_{ij})^{x_i} \\ \text{s.t.} & \mathbf{e}^T \mathbf{x} = k, \\ & \mathbf{x} \in [0, 1]^{|S|}. \end{array}$$

Not SDP-expressible.

Convex solvers

- Interior-point methods for SDP discovered by Alizadeh; Nesterov & Nemirovsky, early 1990s
- Primal-dual path-following proposed by Alizadeh-Haeberly-Overton; H..K..M; and Nesterov-Todd late 1990s.
- Competition led to much software development, early 2000s.
- Software now almost trivial to use even for non-SDP problems via CVX (Boyd-Grant).

Convex solvers

- Interior-point methods for SDP discovered by Alizadeh; Nesterov & Nemirovsky, early 1990s
- Primal-dual path-following proposed by Alizadeh-Haeberly-Overton; H..K..M; and Nesterov-Todd late 1990s.
- Competition led to much software development, early 2000s.
- Software now almost trivial to use even for non-SDP problems via CVX (Boyd-Grant).

Convex solvers

- Interior-point methods for SDP discovered by Alizadeh; Nesterov & Nemirovsky, early 1990s
- Primal-dual path-following proposed by Alizadeh-Haeberly-Overton; H..K..M; and Nesterov-Todd late 1990s.
- Competition led to much software development, early 2000s.
- Software now almost trivial to use even for non-SDP problems via CVX (Boyd-Grant).

Convex solvers

- Interior-point methods for SDP discovered by Alizadeh; Nesterov & Nemirovsky, early 1990s
- Primal-dual path-following proposed by Alizadeh-Haeberly-Overton; H..K..M; and Nesterov-Todd late 1990s.
- Competition led to much software development, early 2000s.
- Software now almost trivial to use even for non-SDP problems via CVX (Boyd-Grant).

Convex solvers

- Interior-point methods for SDP discovered by Alizadeh; Nesterov & Nemirovsky, early 1990s
- Primal-dual path-following proposed by Alizadeh-Haeberly-Overton; H..K..M; and Nesterov-Todd late 1990s.
- Competition led to much software development, early 2000s.
- Software now almost trivial to use even for non-SDP problems via CVX (Boyd-Grant).

Simplifying assumption

Assume all $p_{ij} = p$. Rewritten:

$$\begin{array}{ll} \min & \sum_{j \in R} (1-p)^{A(:,j)^T \mathbf{x}} \\ \text{s.t.} & \mathbf{e}^T \mathbf{x} = k, \\ & \mathbf{x} \in [0,1]^{|S|}. \end{array}$$

Simple generative model

- Assume the noiseless model used earlier.
- Even with this strong assumption, the convex relaxation generally does not recover the influencers. Indeed, they are not necessarily the solution of the IP either.

Simple generative model

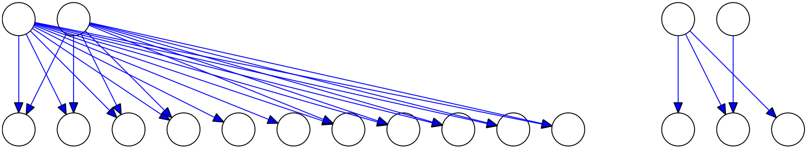
- Assume the noiseless model used earlier.
- Even with this strong assumption, the convex relaxation generally does not recover the influencers. Indeed, they are not necessarily the solution of the IP either.

Simple generative model

- Assume the noiseless model used earlier.
- Even with this strong assumption, the convex relaxation generally does not recover the influencers. Indeed, they are not necessarily the solution of the IP either.

Counterexample: influencers not the IP solution

Counterexample: influencers not the IP solution



Say $p = 0.5$.

Recovery of influencers

- **Theorem.** Suppose that $\frac{\min_l |S_l|}{\max_l |S_l|} \geq \sqrt{1-p}$ and $\mathbf{x}$ is the solution to the convex relaxation. Then $\mathrm{round}(\mathbf{x})$ contains 1 exactly in the positions of the influencers.
- Notation: $\mathrm{round}(\cdot)$ means round each entry to the nearest integer (0 or 1).
- Proof is an elementary application of KKT conditions but not a clean duality argument.

Recovery of influencers

- **Theorem.** Suppose that $\frac{\min_l |S_l|}{\max_l |S_l|} \geq \sqrt{1-p}$ and $\mathbf{x}$ is the solution to the convex relaxation. Then $\mathrm{round}(\mathbf{x})$ contains 1 exactly in the positions of the influencers.
- Notation: $\mathrm{round}(\cdot)$ means round each entry to the nearest integer (0 or 1).
- Proof is an elementary application of KKT conditions but not a clean duality argument.

Recovery of influencers

- **Theorem.** Suppose that $\frac{\min_l |S_l|}{\max_l |S_l|} \geq \sqrt{1-p}$ and $\mathbf{x}$ is the solution to the convex relaxation. Then $\mathrm{round}(\mathbf{x})$ contains 1 exactly in the positions of the influencers.
- Notation: $\mathrm{round}(\cdot)$ means round each entry to the nearest integer (0 or 1).
- Proof is an elementary application of KKT conditions but not a clean duality argument.

Recovery of influencers

- **Theorem.** Suppose that $\frac{\min_l |S_l|}{\max_l |S_l|} \geq \sqrt{1-p}$ and $\mathbf{x}$ is the solution to the convex relaxation. Then $\mathrm{round}(\mathbf{x})$ contains 1 exactly in the positions of the influencers.
- Notation: $\mathrm{round}(\cdot)$ means round each entry to the nearest integer (0 or 1).
- Proof is an elementary application of KKT conditions but not a clean duality argument.

Including noise

- Provided that the number of noise arcs coming into R_l is strongly dominated by the number of nodes of R_l that follow only the influencer, $l = 1, \ldots, k$, a similar result holds with a weaker constant.

Application of convex relaxation to general instances

- Obviously, cannot guarantee exact solution in general case.
- However, if convex relaxation succeeds, can obtain a certificate that optimal integer solution was found.

Future directions

- Multilayer stochastic networks. Kempe et al. have a guaranteed approximation algorithm. Issue: how to evaluate objective function? Sampling?
- More realistic generative model, e.g., forest fire model of Leskovec, Kleinberg, Faloutsos.