2017/8/3 ERATO感謝祭 SeasonIV

Portfolio Optimization for Influence Spread

26th International Conference on World Wide Web

大坂 直人 東京大学
複雑ネットワーク・地図グラフグループ研究協力者

吉田 悠一 NII & PFI
複雑ネットワーク・地図グラフグループリーダ

$$\pi = 0.2\,\{a, b\} + 0.3\,\{a, c\} + 0.5\,\{d, e\}$$

オンラインソーシャルネットワーク上の情報拡散

規模と速度の飛躍
個人の活動履歴のログ

データから学習して、予測・**制御**したい！

グラフ上の最適化問題

影響最大化
[Kempe-Kleinberg-Tardos. *KDD'03*]

拡散発生の早期検知
[Leskovec-Krause-Guestrin-Faloutsos-VanBriesen-Glance. *KDD'07*]

誤情報拡散の阻止
[Budak-Agrawal-Abbadi. *WWW'11*]

バイラルマーケティング
[Domingos-Richardson. *KDD'01*]

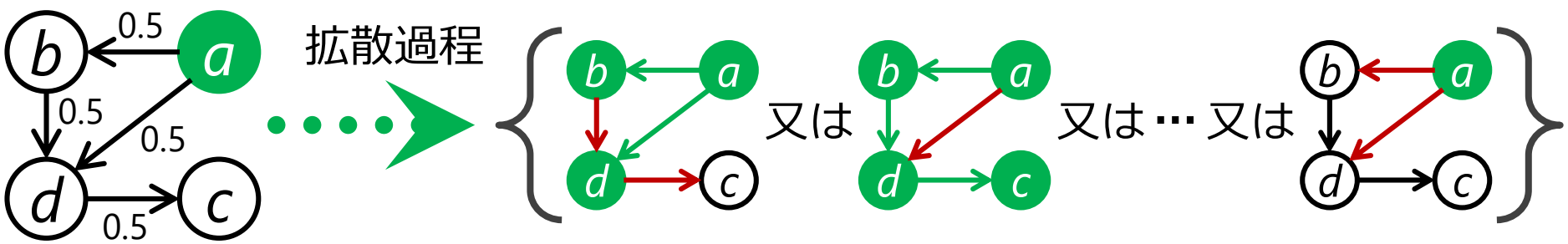
Hotmailの普及 (1996)
18ヶ月で1200万人

Influence Maximization

影響最大化 [Kempe-Kleinberg-Tardos. *KDD'03*]

$$\underset{A:|A|=k}{\operatorname{argmax}}\ \mathbf{E}\left[A\text{の引き起こす拡散のサイズ}\right]$$

(確率的モデルに従う)



劣モジュラ関数の枠組みで定式化
➔ 貪欲算法が$(1-\mathrm{e}^{-1})$近似
[Nemhauser-Wolsey-Fisher. *Math. Program.'78*]

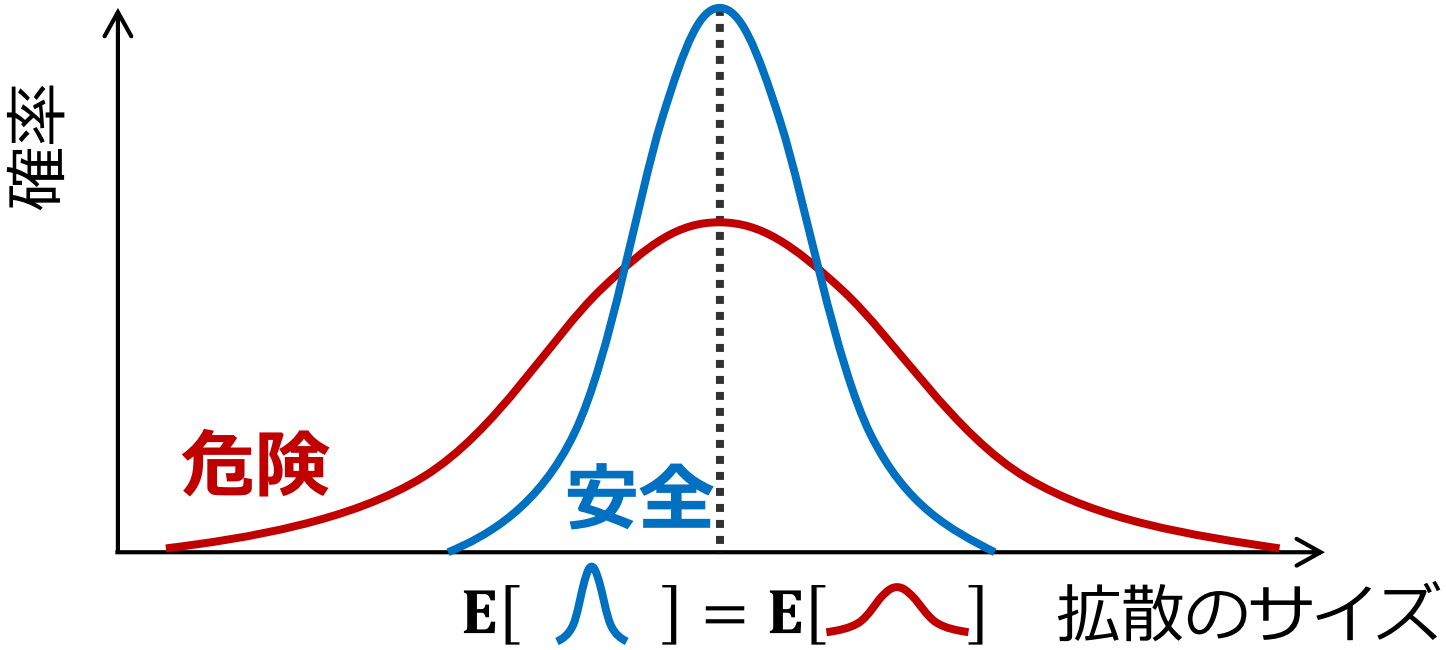
- 効率的計算の追求
- 異なるモデル
- 新たな問題設定

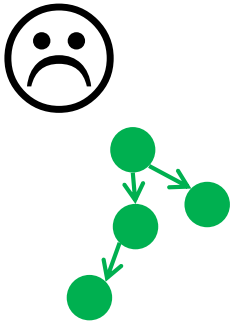
Q. そもそも**期待値最大化**でいいの？

拡散が小さくなるリスク

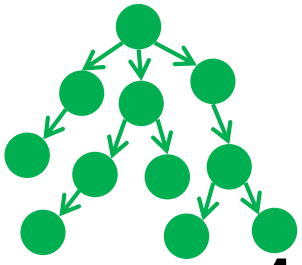
と

のどちらがよいですか?





よいリスク尺度の最適化が必要



下方リスク尺度

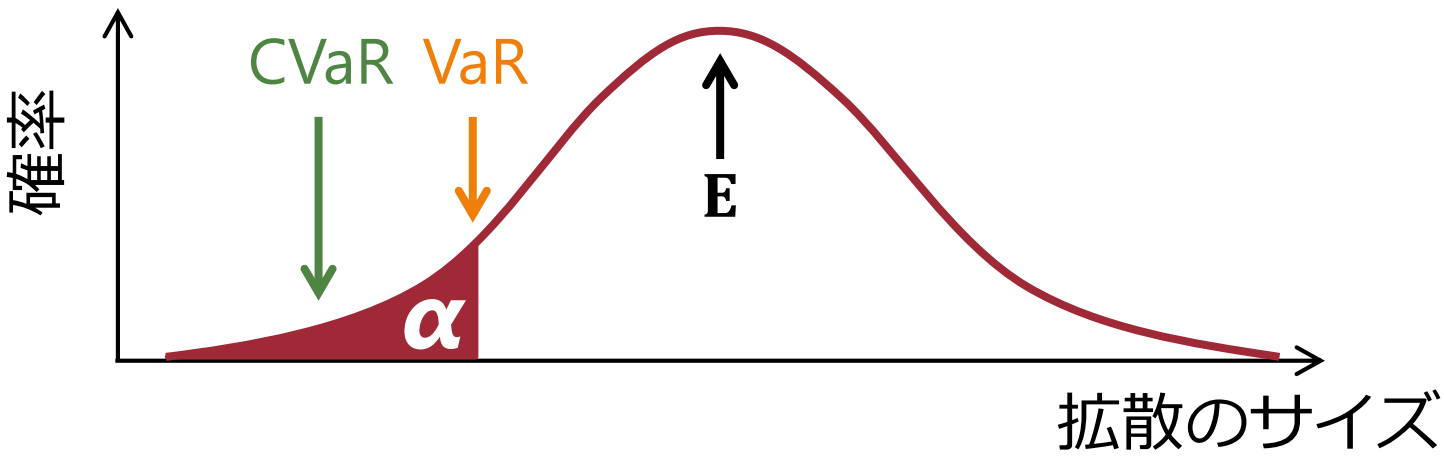
金融経済 & 保険数理の分野から

Value at Risk (VaR) at α = α-percentile

Conditional Value at Risk (CVaR) at α

≒ 最悪時α割合の期待値

α : 有意水準 (典型的に 0.01 又は 0.05)



理論的・実用的に優れた CVaR を採用

コヒーレント性、凸性、等

いかにしてCVaRを最適化するか？

影響最大化の考え方で

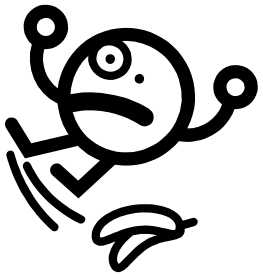
単一集合選択

$A = \{a, b\}$

$\operatorname{argmax}_{A:|A|=k} \mathrm{CVaR}_\alpha[A\text{の拡散サイズ}]$ は解ける?

劣モジュラで無い

多項式時間近似困難 [Maehara. *Oper. Res. Lett.*'15]



分散投資の考え方で

ポートフォリオ構築

$\boldsymbol{\pi} = 0.2\{a, b\} + 0.3\{a, c\} + 0.5\{d, e\}$

連続最適化分野での研究

[Rockafellar-Uryasev. *J. Risk*'00] [Rockafellar-Uryasev. *J. Bank. Financ.*'02]

挑戦：指数個の頂点集合に投資

本研究の貢献

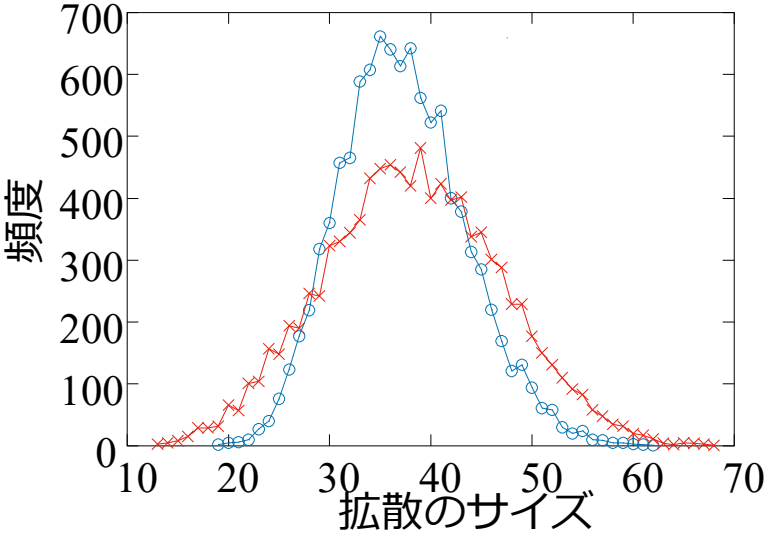
“拡散サイズのCVaRを最大化するポートフォリオ構築”
の多項式時間近似アルゴリズムを提案

影響最大化＝“拡散サイズの期待値を最大化する単一集合選択”

計算機実験による評価

CVaR	大	提案手法	≫	影響最大化	小
E	大	影響最大化	≈	提案手法	小

疎なポートフォリオを出力



拡散現象に伴うリスク回避の関連研究

[Zhang-Chen-Sun-Wang-Zhang. *KDD'14*]

- ▶ “(拡散のサイズのVaR) ≥ 閾値”

[Deng-Du-Jia-Ye. *WASA'15*]

- ▶ “(#高確率でAに影響する頂点) ≥ 閾値”
- ▶ 近似保証無し

ロバスト影響最大化

[Chen-Lin-Tan-Zhao-Zhou. *KDD'16*] [He-Kempe. *KDD'16*]

- ▶ モデルパラメータが不確実
- ▶ $\mathbf{E}$[拡散のサイズ]を安定させる

最大の違い：単一集合選択 OR ポートフォリオ構築

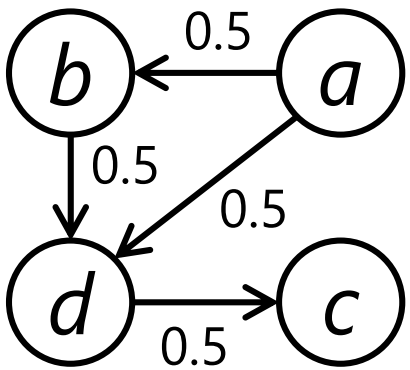
関連研究　　本研究

定式化

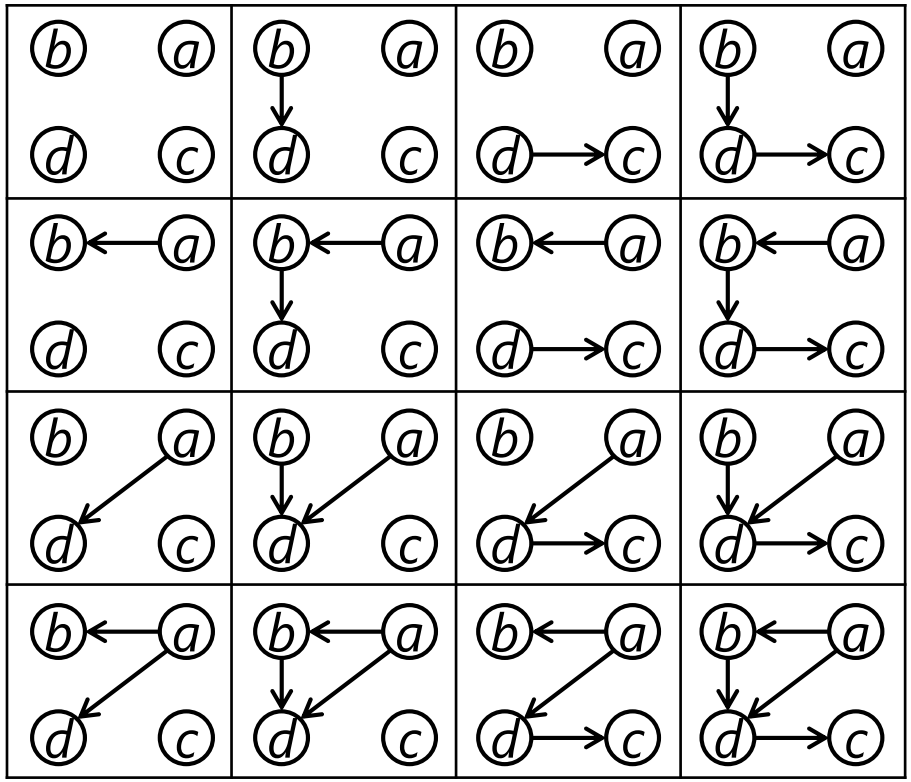
拡散モデル 独立カスケード [Goldenberg-Libai-Muller. *Market. Lett.'01*]

グラフ $G = (V, E)$

辺確率 $p: E \to [0,1]$



uvを確率p_{uv}で残す



$2^{|E|}$ 通りのランダムグラフ

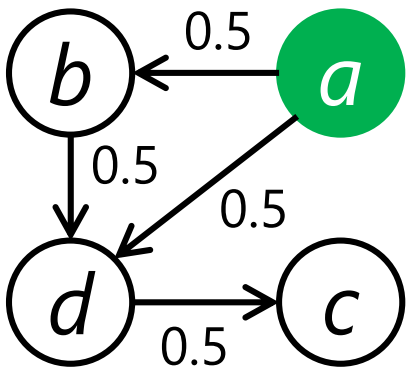
この例では等確率で発生

定式化

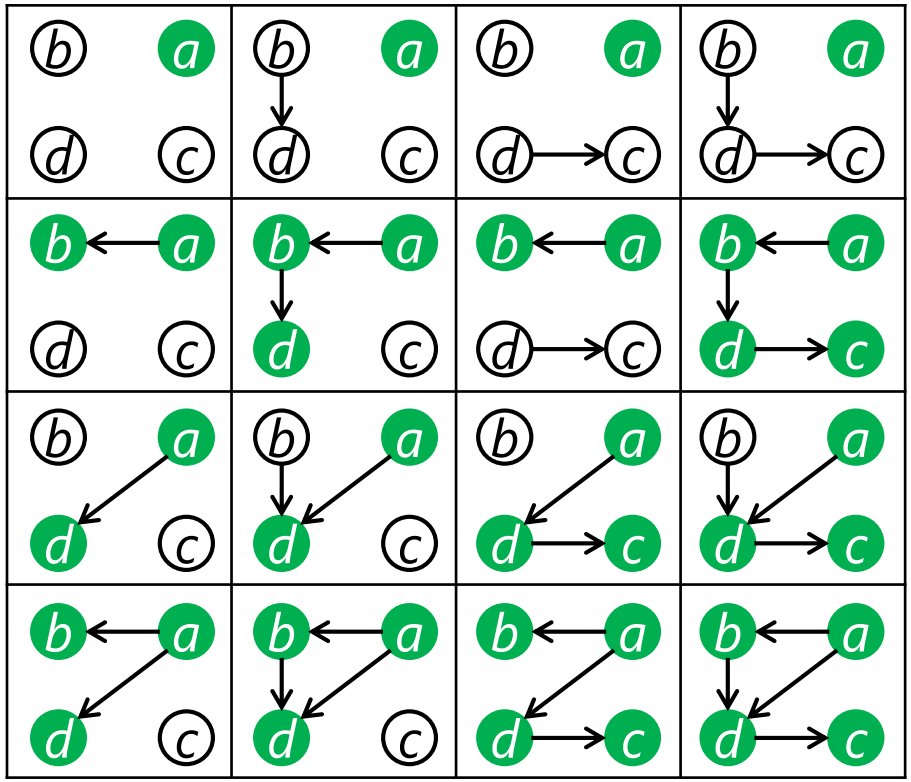
拡散モデル 独立カスケード [Goldenberg-Libai-Muller. *Market. Lett.*'01]

グラフ $G = (V, E)$

辺確率 $p: E \to [0,1]$



uvを確率p_{uv}で残す



R_aの分布

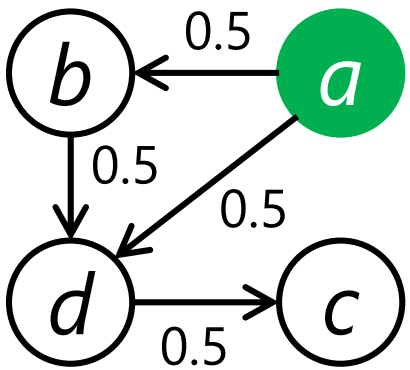
R_A = Aから**到達可能**な頂点数

これがAの拡散のサイズ

定式化

拡散モデル 独立カスケード [Goldenberg-Libai-Muller. *Market. Lett.*'01]

グラフ $G = (V, E)$
辺確率 $p: E \to [0,1]$



uvを確率p_{uv}で残す

1	1	1	1
2	3	2	4
2	2	3	3
3	3	4	4

R_aの分布

R_A = Aから**到達可能**な頂点数

$\mathbf{E}[R_a] = 2.4375$ 全体の平均

$\mathrm{CVaR}_{0.25}[R_a] = 1$ **最悪時25%**の平均

定式化

k頂点ポートフォリオ

$\binom{|V|}{k}$次元ベクトル $\boldsymbol{\pi}$ $s.t.\ \|\boldsymbol{\pi}\|_1 = 1$

例 : $k = 1$, $\pi_a = \pi_b = 0.5$

$\langle \boldsymbol{\pi}, \mathbf{R} \rangle = \pi_a R_a + \pi_b R_b$

1	**1.5**	**1**	2
1.5	2.5	1.5	3.5
1.5	2	2	3
2	2.5	2.5	3.5

R_a

1	**1**	**1**	**1**
2	3	2	4
2	2	3	3
3	3	4	4

R_b

1	2	1	3
1	2	1	3
1	2	1	3
1	2	1	3

$$\underbrace{\mathrm{CVaR}_{0.25}[\pi_a R_a + \pi_b R_b]}_{\mathbf{1.25}} \geq \underbrace{\pi_a \mathrm{CVaR}_{0.25}[R_a]}_{0.5\times\mathbf{1}} + \underbrace{\pi_b \mathrm{CVaR}_{0.25}[R_b]}_{0.5\times\mathbf{1}}$$

優加法性

定式化

問題定義

入力 : $G = (V, E)$, p, 整数 k, 有意水準 α

出力 : k頂点ポートフォリオ $\boldsymbol{\pi}$

$$\boxed{\max_{\boldsymbol{\pi}} \ \mathrm{CVaR}_{\alpha}[\langle \boldsymbol{\pi}, \mathbf{R} \rangle]}$$

$$\| \quad \sum_{A:|A|=k} \pi_A R_A$$

ランダムグラフ上で
Aから到達可能な頂点数

⚠ $\boldsymbol{\pi}$ & $\mathbf{R}$ は $\binom{|V|}{k}$次元

既存手法は適用困難

提案アルゴリズム
概要

CVaR 最大化
元の問題

↓

巨大線形計画
複雑な制約

↓

制約の凸結合
指数個の変数

標準的変形 [Rockafellar-Uryasev. *J. Risk'00*]
サンプリング近似 標本数のバウンドが必要

乗算型重み更新 [Arora-Hazan-Kale. '12]
最適化、機械学習、ゲーム理論等に出現
感謝祭 SeasonII [Hatano-Yoshida. *AAAI'15*]
AdaBoost [Freund-Schapire. *J. Comput. Syst. Sci.'97*]

実は 貪欲算法で近似的に解ける

提案アルゴリズム

第一段階：標準的近似

$$\max_{\boldsymbol{\pi}} \ \mathrm{CVaR}_{\alpha}[\langle \boldsymbol{\pi}, \mathbf{R} \rangle]$$

最適化問題としてCVaRを記述

[Rockafellar-Uryasev. *J. Risk'00*]

$\mathbf{R}^1, \ldots, \mathbf{R}^s$ を $\mathbf{R}$ の分布から標本

$$\max_{\boldsymbol{\pi},\tau} \ \tau - \frac{1}{\alpha s} \sum_{1 \le i \le s} \max\left\{\tau - \left\langle \boldsymbol{\pi}, \mathbf{R}^i \right\rangle, 0\right\}$$

実行可能性問題“$\square \ge \gamma$?”を
繰り返し解き、 γで二分法

提案アルゴリズム

“$\square \geq \gamma$?”判定の難しさ

“$\square \geq \gamma$?” ＝ 巨大線形計画 $\square$ の実行可能性

$$\mathbf{x} = \begin{bmatrix} \tau \\ \mathbf{y} \\ \boldsymbol{\pi} \end{bmatrix}$$

CVaR記述のための補助変数
max関数除去の補助変数
k頂点ポートフォリオ

$$\boxed{\exists ?\, \mathbf{x} \in \mathbf{P}_\gamma \qquad \mathbf{A}\mathbf{x} \geq \mathbf{b}}$$

≈

#変数 ≈ $\binom{|V|}{k}$
#制約 ＝ s

ポートフォリオを構築し
“複数の劣モジュラ関数が同時に閾値以上”

$$\boxed{\begin{array}{c} y_1 + \langle \boldsymbol{\pi}, \mathbf{R}^1 \rangle \geq \tau \\ \vdots \\ y_s + \langle \boldsymbol{\pi}, \mathbf{R}^s \rangle \geq \tau \end{array}}$$

これは難しい

提案アルゴリズム

巨大線形計画 □ の実行可能性の近似判定

乗算型重み更新 [Arora-Hazan-Kale. '12]

① 凸結合 □ を $\mathbf{p} = \mathbf{p}_1, \ldots, \mathbf{p}_T$ について解く

② 平均の解 ≈ □ の解

$$\exists ?\, \mathbf{x} \in \mathbf{P}_\gamma \qquad \langle \mathbf{p}, \mathbf{A}\mathbf{x} \rangle \geq \langle \mathbf{p}, \mathbf{b} \rangle$$

≈

#変数 ≈ $\binom{|V|}{k}$

#制約 = 1

ポートフォリオを構築し

“単一の劣モジュラ関数が閾値以上”

$$\langle \mathbf{p}, \mathbf{y} \rangle + \langle \boldsymbol{\pi}, \mathbf{p} \odot \mathbf{R}^{\cdot} \rangle \geq \tau$$

発見：貪欲算法の解A^*にだけ投資 $(\pi_{A^*} = 1)$

すると、$\langle \mathbf{p}, \mathbf{A}\mathbf{x} \rangle \geq \langle \mathbf{p}, \mathbf{b} \rangle - \mathrm{e}^{-1}$

提案アルゴリズム

まとめ：精度と時間

$$\mathrm{CVaR}_\alpha[\langle \boldsymbol{\pi}, \mathbf{R} \rangle] \geq \max_{\boldsymbol{\pi}^*} \mathrm{CVaR}_\alpha[\langle \boldsymbol{\pi}^*, \mathbf{R} \rangle] - |V|\mathrm{e}^{-1} - |V|\epsilon$$

最適値　　付加誤差

$O(\epsilon^{-6} k^2 |V||E| \,\mathrm{polylog})$ 時間

厳密 CVaR 最適化

≈ ランダムグラフ標本　$s = O(k\epsilon^{-4}\,\mathrm{polylog})$ サンプル

経験 CVaR 最適化

↓ 二分法 $\Theta(\log \epsilon^{-1})$回

巨大線形計画

↓ 乗算型重み更新　$T = O(\epsilon^{-2}\,\mathrm{polylog})$ ステップ

制約の凸結合　貪欲算法 $O(k|V||E|)$ 時間

実験

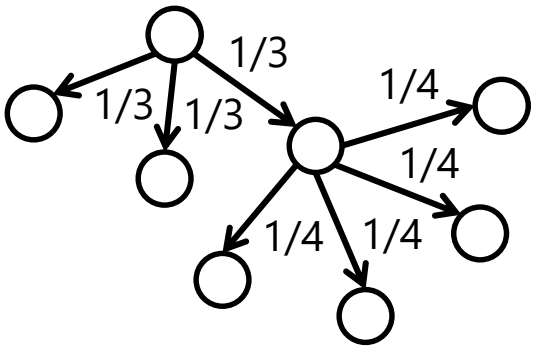
設定

- ネットワークデータ (その他の結果は論文に記載)

  Physicians friend network ($|V| = 117$ & $|E| = 542$)

  [Koblenz Network Collection]

  $p_{uv} = (u\text{の出次数})^{-1}$



- パラメタ : $\epsilon = 0.4$

- ベースライン : 単一k頂点集合を出力

Greedy	影響最大化の標準的解法 [Kempe-Kleinberg-Tardos. *KDD'03*]
Degree	次数順にk頂点選択
Random	無作為にk頂点選択

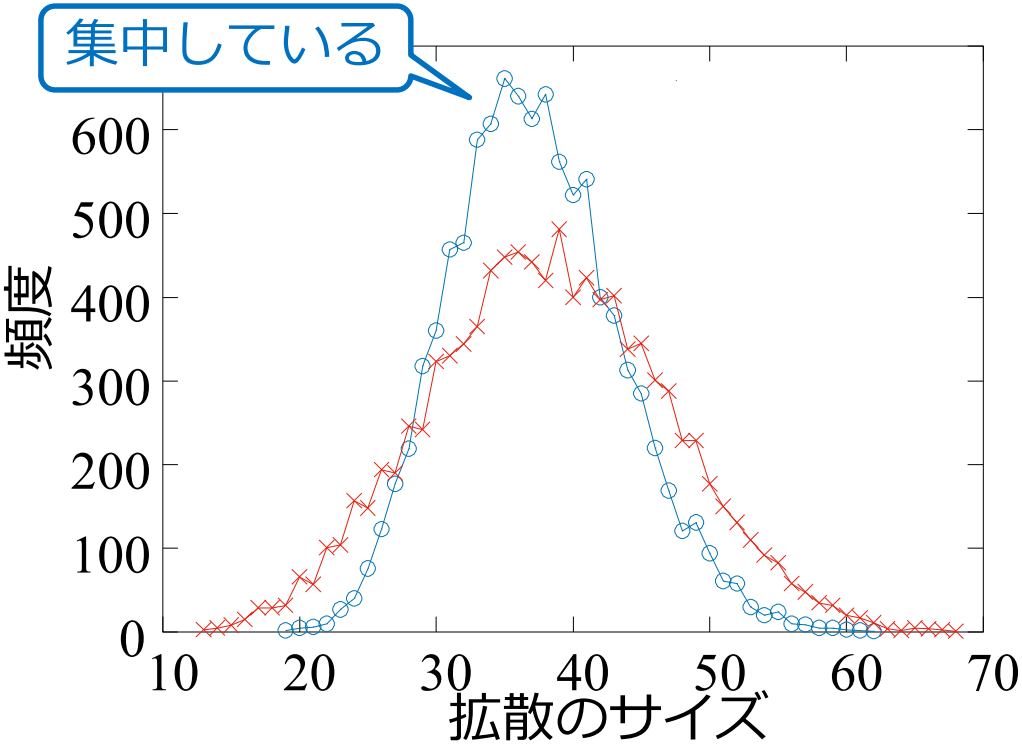
- 実験環境 : Intel Xeon E5-2670 2.60GHz CPU, 512GB RAM

実験

k = 10 & α = 0.01 の結果

	CVaR at α	期待値	実行時間
本研究	**23.6**	37.7	157秒
Greedy	16.9	**38.2**	100ミリ秒
Degree	14.2	30.8	0.2ミリ秒
Random	15.1	31.0	0.2ミリ秒

匹敵



π は非常に疎!!

(**π**の非ゼロ要素数) = **42**

π の次元 = $\binom{|V|}{10}$ ≈ $\mathbf{10^{14}}$

A	π_A
25,42,67,81,85,94,103,106,111,112	3/47
7,20,21,48,75,98,104,111,112,113	2/47
0,29,43,52,92,97,107,108,113,116	2/47
25,38,69,71,81,103,105,110,112,116	2/47
⋮	⋮

まとめ

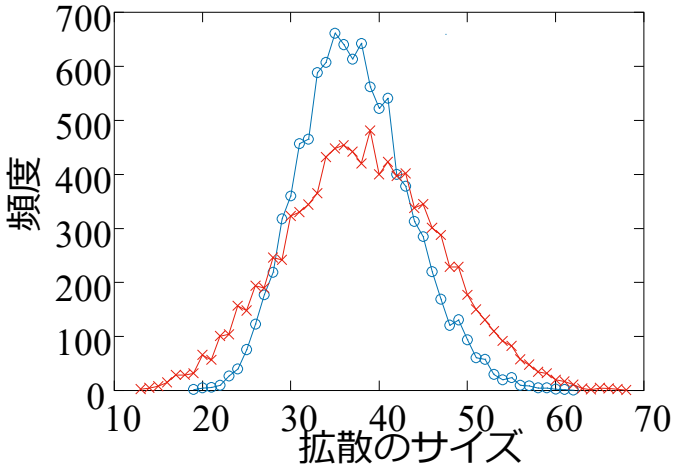
ポートフォリオ最適化による
影響最大化の低リスク戦略の獲得

Conditional Value at Risk
を

$\pi = 0.2\,\{a, b\} + 0.3\,\{a, c\} + 0.5\,\{d, e\}$

で最適化

すると



今後の発展

- 高速化：巨大線形計画をより速く解ける？
- CVaRの効率的近似
- 他のリスク尺度：Value at Risk、下方部分積率、…

感謝祭の関連研究

2014

影響最大化の高速化技法
大坂-秋葉-吉田-河原林 *AAAI'14*

2015

分散制約最適化の新解法
波多野-**吉田** *AAAI'15*

2016

動的グラフの実時間解析
大坂-秋葉-吉田-河原林 *VLDB'16*

共通利用技術
乗算型重み更新

2017

巨大グラフの小型化
大坂-**薗部**-藤田-河原林 *SIGMOD'17*
明日

ポートフォリオ最適化
によるリスク回避
大坂-吉田 *WWW'17*

影響拡散の厳密計算
前原-鈴木-石畠 *WWW'17*
明日

WWW'17 Social 2 session

Why ϵ^{-6}?

- #steps T in multiplicative weights alg.
  - $O(\epsilon^{-2})$ [Arora-Hazan-Kale. '12]
- #samplings = #graphs = $O(\epsilon^{-4})$
  - Bound the error for the all portfolios
  - #portfolios $|L|$ to be considered = $\binom{\binom{|V|}{k}+T-1}{T}$
  - $O\left(\frac{1}{\epsilon^2}\log|L|\right)$

What actually is $\mathbf{Ax} \geq \mathbf{b}$ & $\langle \mathbf{p}, \mathbf{Ax} \rangle \geq \langle \mathbf{p}, \mathbf{b} \rangle$?

- $\mathbf{Ax} \geq \mathbf{b} \Leftrightarrow$

$$y_i + \sum_{A:|A|=k} \pi_A X_A^i \geq \tau \quad (1 \leq i \leq s)$$

- $\langle \mathbf{p}, \mathbf{Ax} \rangle \geq \langle \mathbf{p}, \mathbf{b} \rangle \Leftrightarrow$

$$\sum_i p_i y_i + \sum_{A:|A|=k} \pi_A \boxed{\sum_i p_i X_A^i} \geq \tau$$

- $\mathbf{P}_\gamma$ requires $\tau - \frac{1}{\alpha s} \sum_{1 \leq i \leq s} y_i \geq \gamma$

Submodular in A

$$\mathbf{x} = \begin{bmatrix} \tau \\ \mathbf{y} \\ \boldsymbol{\pi} \end{bmatrix}$$

auxiliary variable required for expressing CVaR
auxiliary variables for removing $\max$ functions
k-vertex portfolio

Our algorithm

Linear programming formulation

$$\mathbf{x} = \begin{bmatrix} \tau \\ \mathbf{y} \\ \boldsymbol{\pi} \end{bmatrix}$$

- τ: auxiliary variable required for expressing CVaR
- $\mathbf{y}$: auxiliary variables for removing $\max$ functions
- $\boldsymbol{\pi}$: k-vertex portfolio

Polyhedron for "$\square \geq \gamma$?" s constraints on $\mathbf{y}$

$$\exists ?\, \mathbf{x} \in \mathbf{P}_\gamma \quad \mathbf{A}\mathbf{x} \geq \mathbf{b}$$

Convex combination One constraint on $\mathbf{y}$

$$\text{For } \mathbf{p} \in [0,1]^s \quad \exists ?\, \mathbf{x} \in \mathbf{P}_\gamma \quad \langle \mathbf{p}, \mathbf{A}\mathbf{x} \rangle \geq \langle \mathbf{p}, \mathbf{b} \rangle$$

Our finding:

☺ $\square$ can be approximated in poly-time

Key : Original influence maximization [Kempe-Kleinberg-Tardos. '03]

+ Multiplicative weights alg. [Arora-Hazan-Kale. '12]

☺ We can approximately check $\square$

Our algorithm

Approximately checking $\square$ a multiplicative weights algorithm

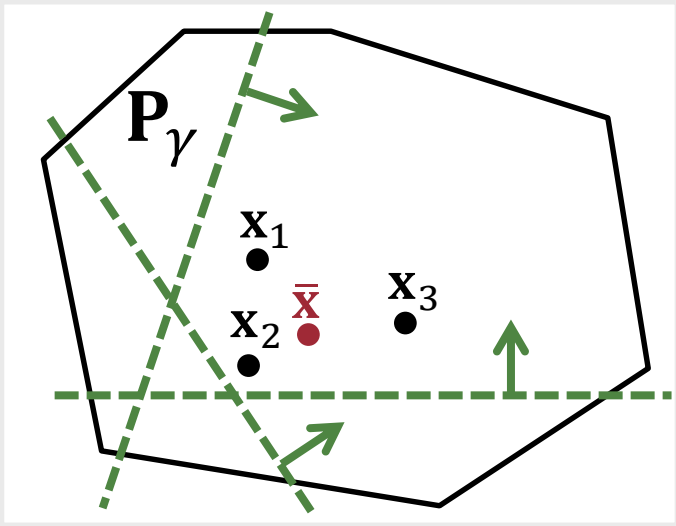
[Arora-Hazan-Kale. '12]

$$\exists ?\, \mathbf{x} \in \mathbf{P}_\gamma \quad \langle \mathbf{p}, \mathbf{A}\mathbf{x} \rangle \geq \langle \mathbf{p}, \mathbf{b} \rangle$$

Generate $\mathbf{p}_1$, solve for $\mathbf{p}_1$, & obtain $\mathbf{x}_1$
Generate $\mathbf{p}_2$, solve for $\mathbf{p}_2$, & obtain $\mathbf{x}_2$
Generate $\mathbf{p}_3$, solve for $\mathbf{p}_3$, & obtain $\mathbf{x}_3$

$\vdots$

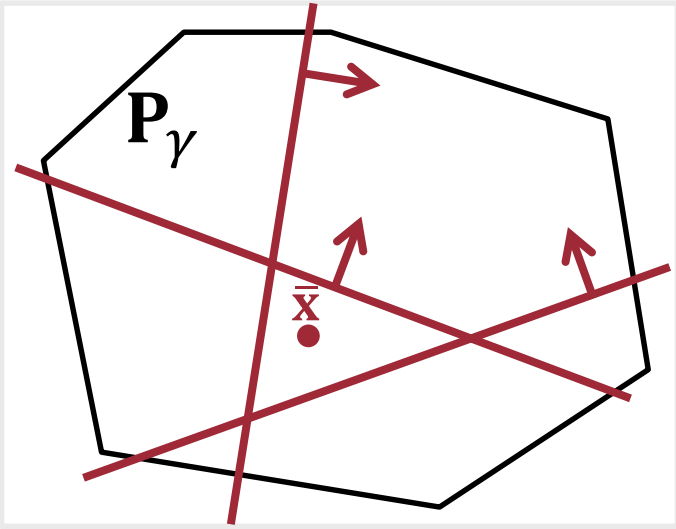
$\bar{\mathbf{x}} = \text{average}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \ldots)$



$$\exists ?\, \mathbf{x} \in \mathbf{P}_\gamma \quad \mathbf{A}\mathbf{x} \geq \mathbf{b}$$

$$\bar{\mathbf{x}} \in \mathbf{P}_\gamma \;\&\; \mathbf{A}\bar{\mathbf{x}} \geq \mathbf{b} - \mathbf{1}\epsilon$$

[Arora-Hazan-Kale. '12]



Our algorithm

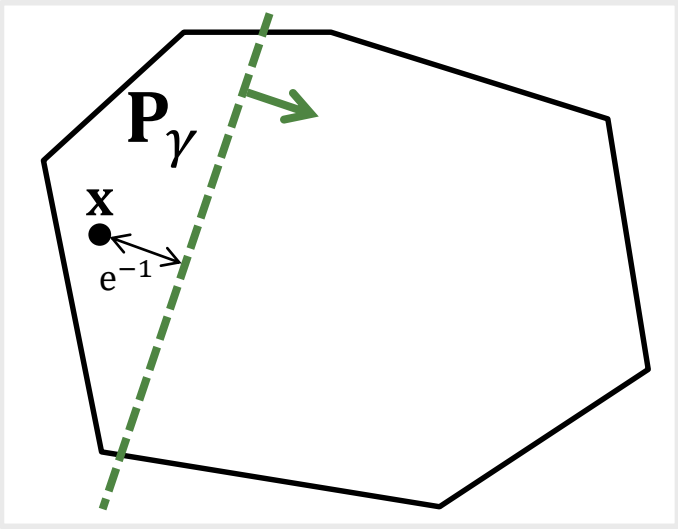
Approximately checking a greedy algorithm

$$\exists ?\, \mathbf{x} \in \mathbf{P}_\gamma \quad \langle \mathbf{p}, \mathbf{A}\mathbf{x} \rangle \geq \langle \mathbf{p}, \mathbf{b} \rangle$$

≈

“ A single submodular function exceeds a threshold ”

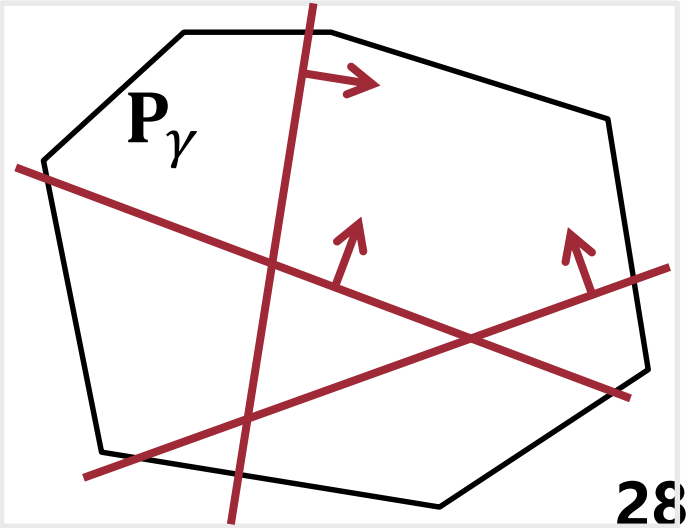
The greedy algorithm achieves $\langle \mathbf{p}, \mathbf{A}\mathbf{x} \rangle \geq \langle \mathbf{p}, \mathbf{b} \rangle - \mathrm{e}^{-1}$



$$\exists ?\, \mathbf{x} \in \mathbf{P}_\gamma \quad \mathbf{A}\mathbf{x} \geq \mathbf{b}$$

≈

“ Multiple submodular functions exceed a threshold simultaneously ”



Our algorithm

So, what we can do finally?

Check $\boxed{\exists ?\, \mathbf{x} \in \mathbf{P}_\gamma \quad \mathbf{A}\mathbf{x} \geq \mathbf{b} - \mathrm{e}^{-1} - \epsilon}$ me

$$(\#\ \mathbf{p}\text{'s}) \cdot (\text{Time of greedy alg.}) = \mathrm{O}(\epsilon^{-2} k |V||E| s \log s)$$

Properly setting s & # bisection searches

See our paper for details

W.h.p., our algorithm produces $\boldsymbol{\pi}$ *s.t.*

$$\mathrm{CVaR}_\alpha(\langle \boldsymbol{\pi}, \mathbf{X} \rangle) \geq \max_{\boldsymbol{\pi}^*} \mathrm{CVaR}_\alpha\left(\langle \boldsymbol{\pi}^*, \mathbf{X} \rangle\right) - \mathrm{e}^{-1} - \epsilon$$

in time $\widetilde{\mathrm{O}}(\epsilon^{-6} k^2 |V||E|)$

NOTE : $\widetilde{\mathrm{O}}(f) = \mathrm{O}(f \log^c f)$

Properties of CVaR

CVaR has

- Monotonicity
- Translation invariance
- Concavity

Running time

Physicians dataset $|V|$ = 117 & $|E|$ = 542

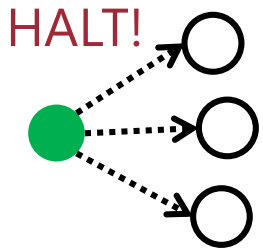
	k=1	k=10
This work	1.8s	157.2s
Greedy	0.1s	0.1s
Degree	0.2ms	0.2ms
Random	0.2ms	0.2ms

Advogato dataset $|V|$ = 5,042 & $|E|$ = 78,454

	k=1	k=10
This work	185.9s	14,094.8s
Greedy	2.8s	3.5s
Degree	0.4ms	0.4ms
Random	0.1ms	0.1ms

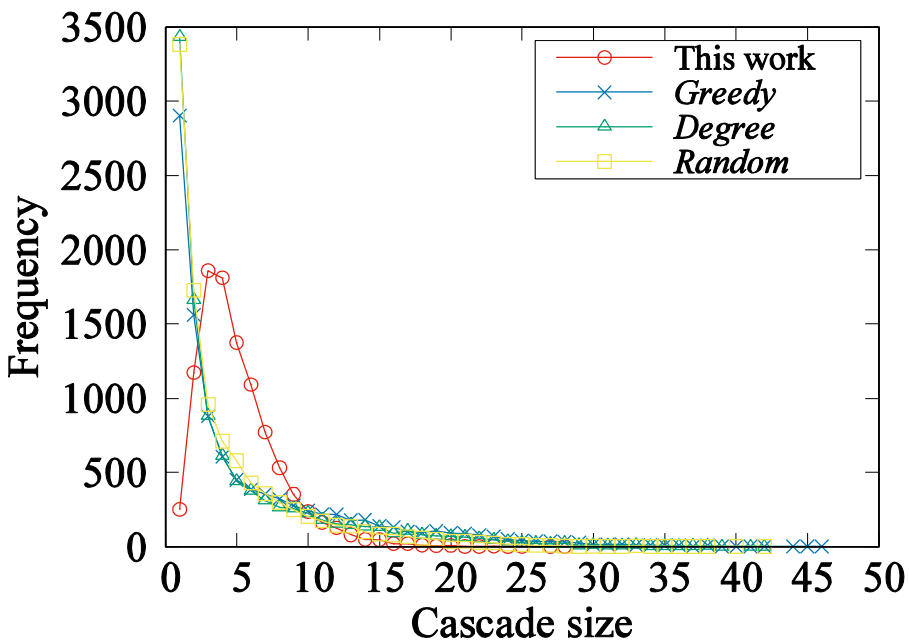
Experiments

Results for $\boldsymbol{k} = \mathbf{1}$ & $\alpha = 0.01$



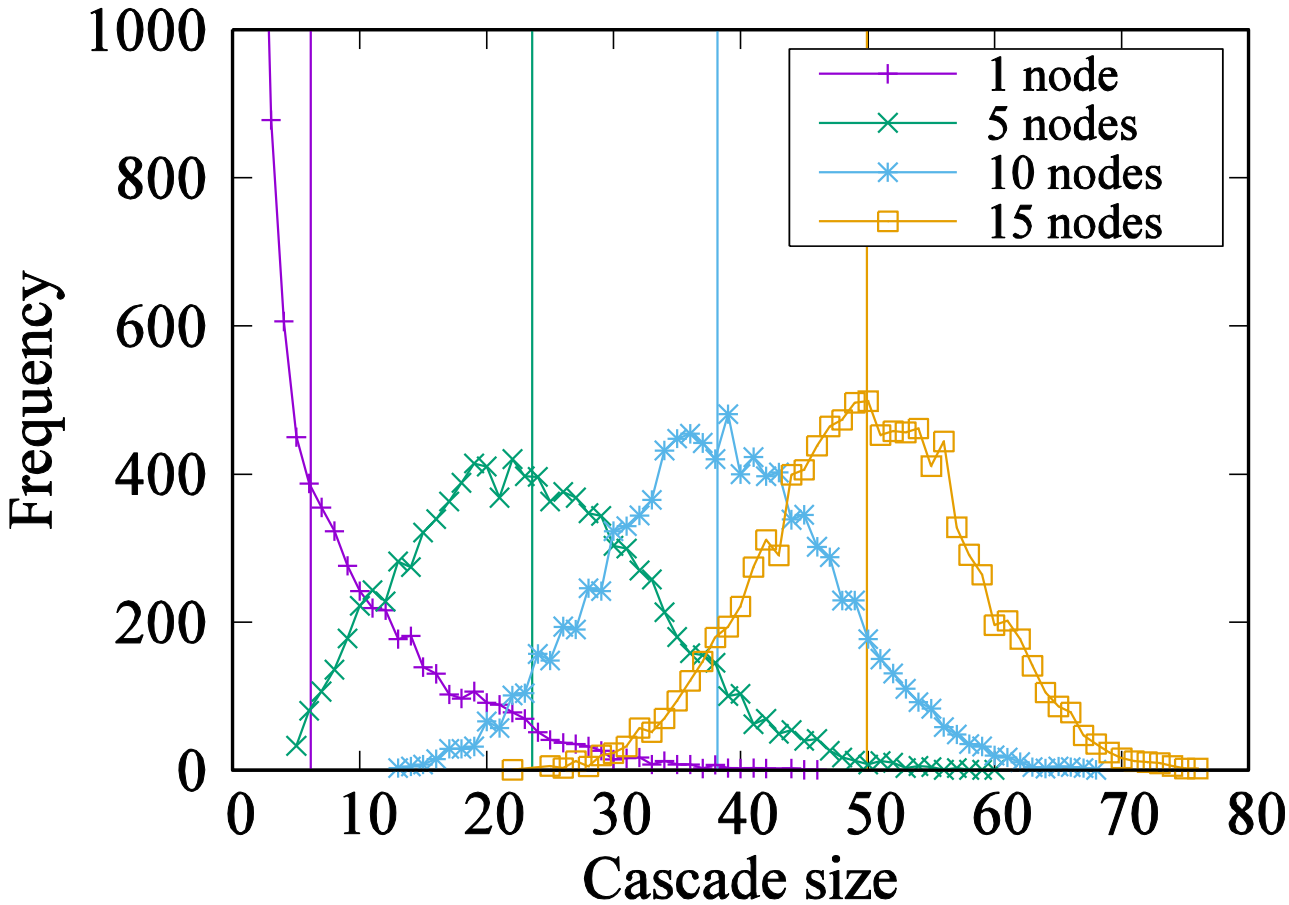
	CVaR at α	Expectation	Runtime
This work	**1.5**	5.6	1.8s
Greedy	1.0	**6.2**	0.1s
Degree	1.0	5.2	0.2ms
Random	1.0	4.5	0.2ms

Comparable

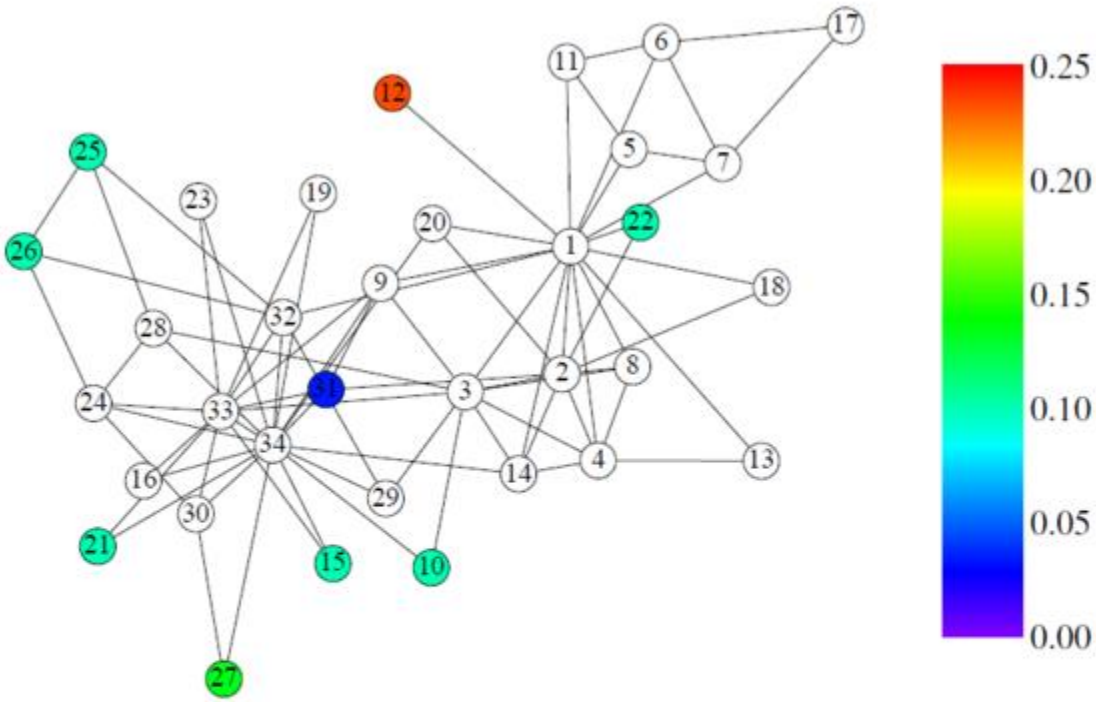


(# non-zeros in $\boldsymbol{\pi}$) = 10

dim. of $\boldsymbol{\pi}$ = $\binom{|V|}{1}$ = 117



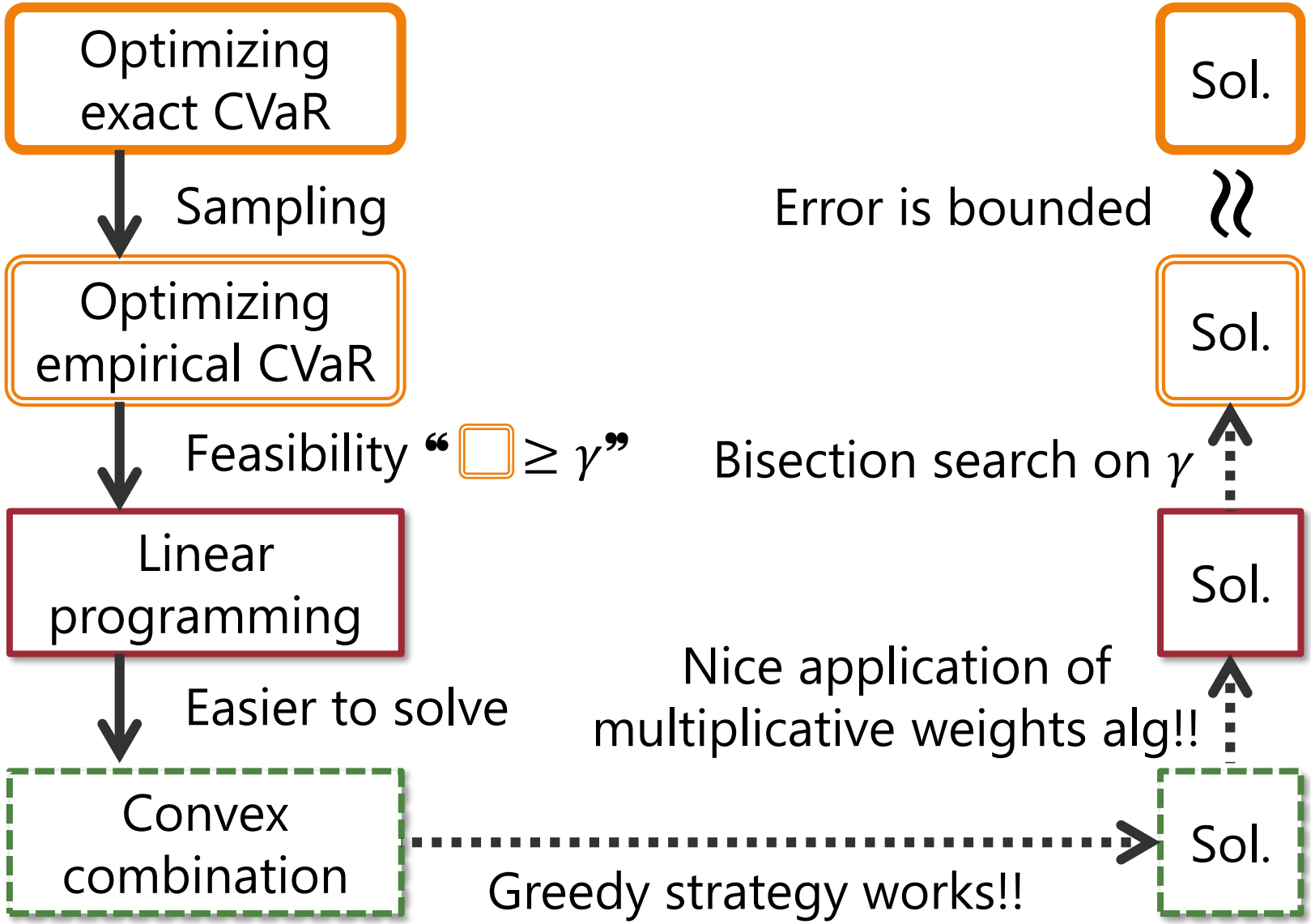
Histogram of cascade sizes for seed sets selected by the standard greedy algorithm in the independent cascade model on Physicians. Vertical lines correspond to the mean values.



v	π_v
12	0.233
27	0.133
10	0.100
15	0.100
21	0.100
22	0.100
25	0.100
26	0.100
31	0.033

Figure 3: Visualization of Karate network. Each vertex is colored according to its portfolio weight ($\alpha = 0.01$).

Summary


Optimizing exact CVaR
Sampling
Optimizing empirical CVaR
Feasibility "□ ≥ γ"
Linear programming
Easier to solve
Convex combination
Greedy strategy works!!
Sol.
Nice application of multiplicative weights alg!!
Sol.
Bisection search on γ
Sol.
Error is bounded
Sol.