2017/4/6 WWW'17@Perth, Australia

Portfolio Optimization for Influence Spread

Naoto Ohsaka (UTokyo)

Yuichi Yoshida (NII & PFI)

Kawarabayashi Large Graph Project

ERATO

$$\pi = 0.2 \begin{smallmatrix} a \\ b \end{smallmatrix} + 0.3 \begin{smallmatrix} a \\ c \end{smallmatrix} + 0.5 \begin{smallmatrix} d \\ e \end{smallmatrix}$$

Influence maximization

Find the most influential group from a social network

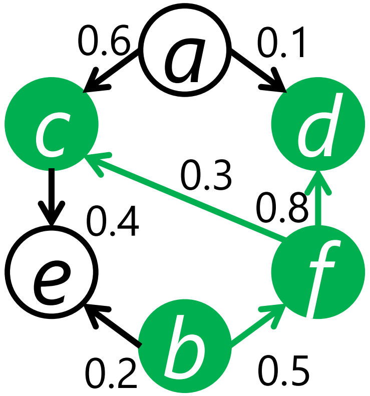
Motivated by viral marketing [Domingos-Richardson. KDD'01]

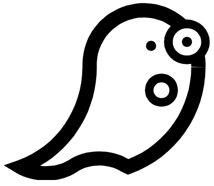
Discrete optimization problem under stochastic models [Kempe-Kleinberg-Tardos. KDD'03]

$$\max_{A:|A|=k} \mathbf{E}[\text{cascade size trigger by } A]$$

\# vertices influenced

Greedy strategy is $1 - \mathrm{e}^{-1} \approx 63\%$-approx.
[Nemhauser-Wolsey-Fisher. Math. Program.'78]
Due to monotonicity & submodularity
[Kempe-Kleinberg-Tardos. KDD'03]

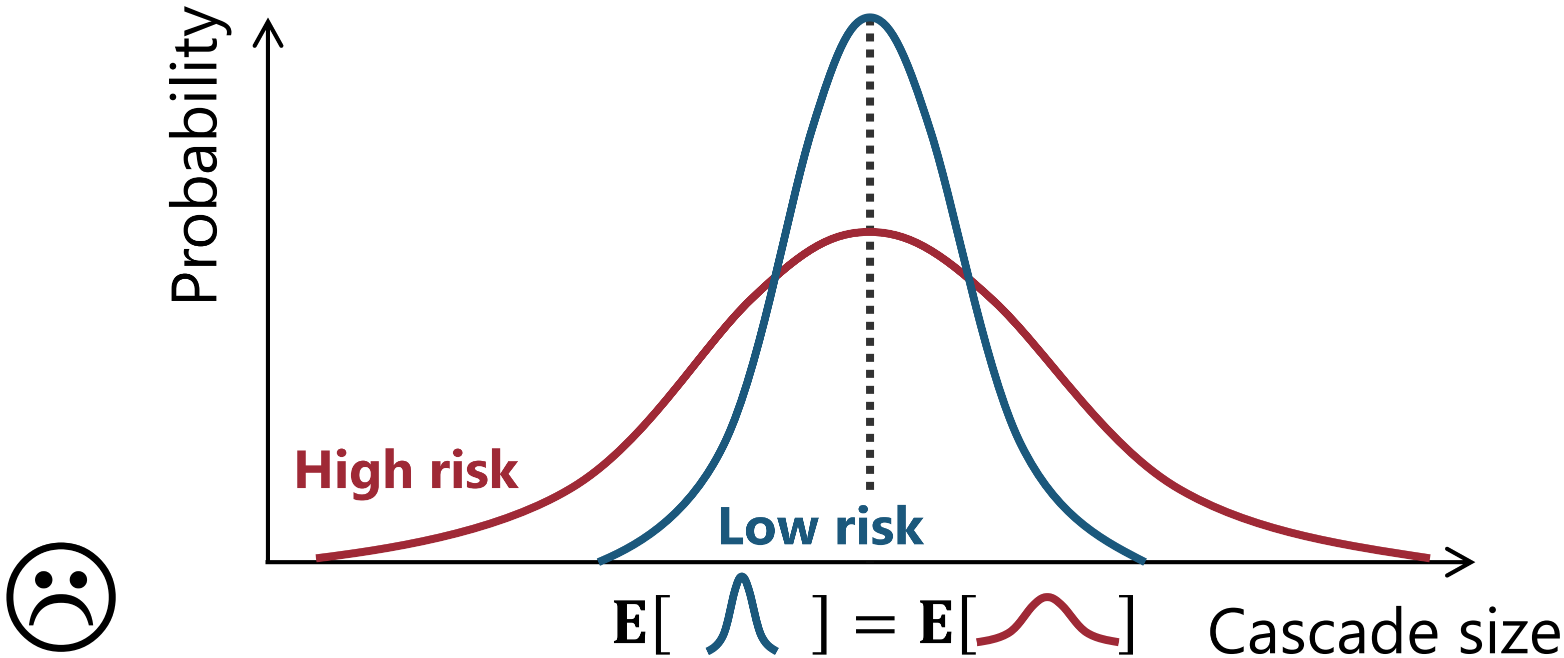


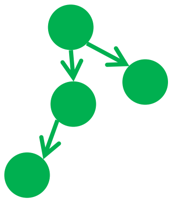


Is **maximizing expectation** enough?

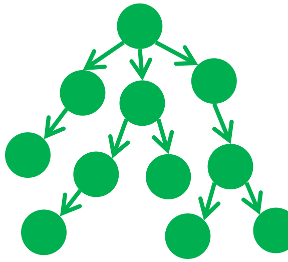
Risk of having small cascades

Q. Which is better,  or  ?





Need a *good risk measure* for finding a low-risk strategy



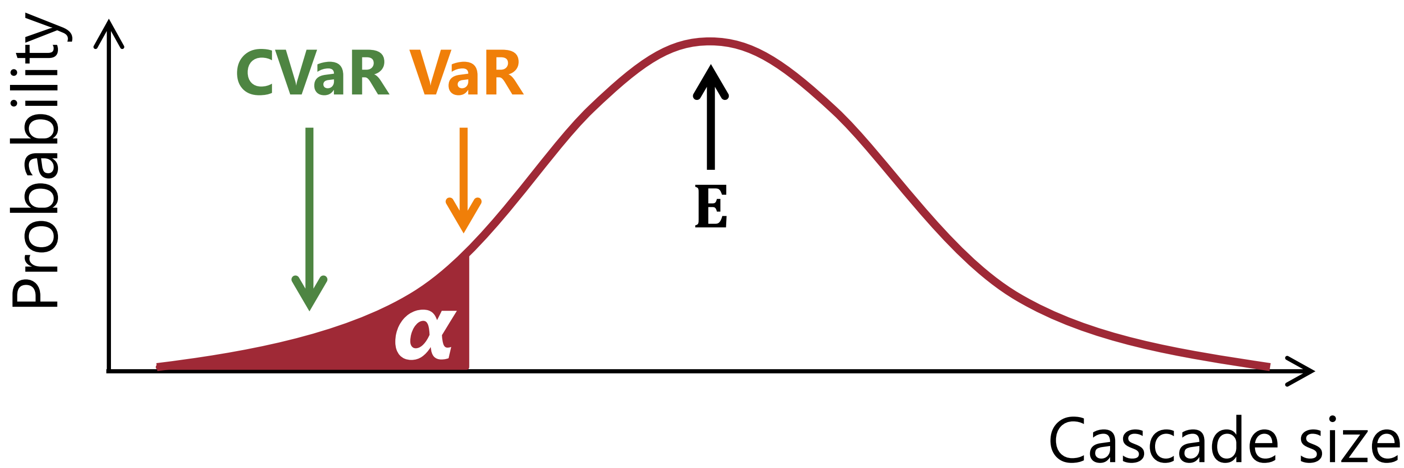
Popular downside risk measures
in finance economics & actuarial science

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

Conditional Value at Risk at α ($CVaR_\alpha$)

≒ expectation in the worst α-fraction of cases

α : significance level (typically 0.01 or 0.05)



CVaR is appropriate in practice & theory

coherence, convex/concave, continuous

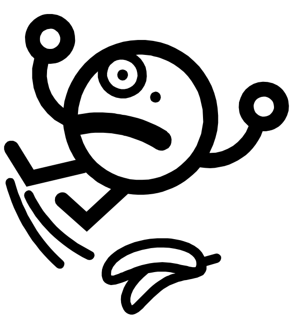
Optimizing CVaR

Much effort in continuous optimization
[Rockafellar-Uryasev. J. Risk'00] [Rockafellar-Uryasev. J. Bank. Financ.'02]

Solve $\max_{A:|A|=k} \mathrm{CVaR}_\alpha[\text{cascade size triggered by } A]$?

NOT submodular

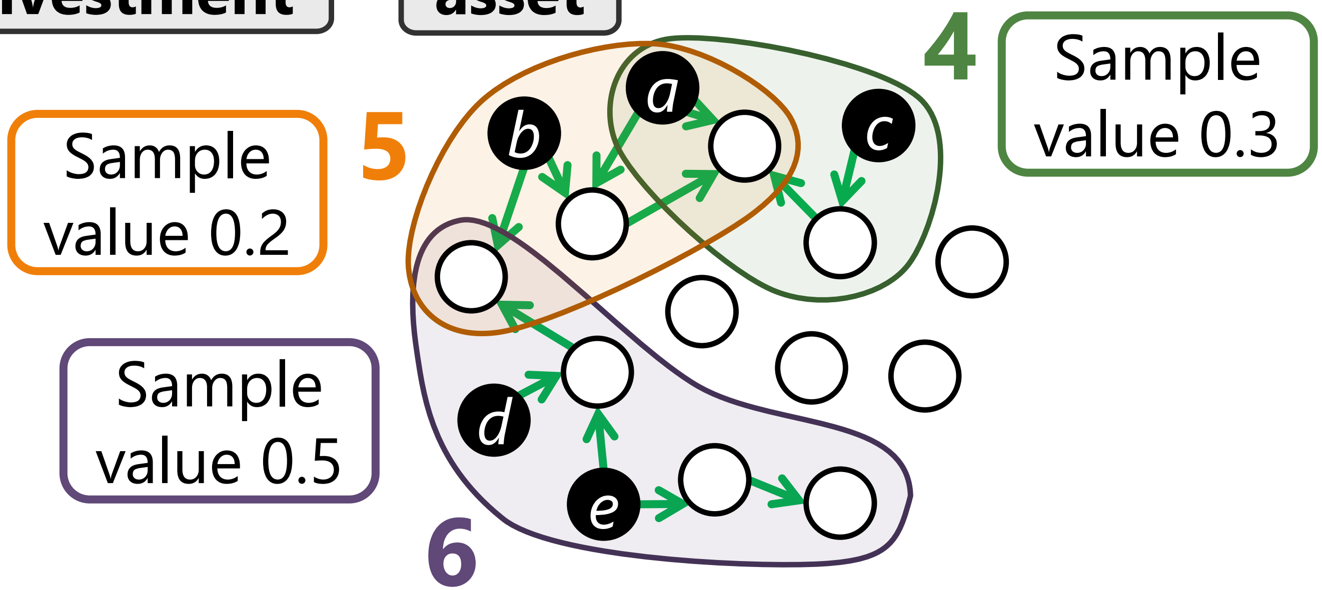
Poly-time approximation is impossible
under some assumption [Maehara. Oper. Res. Lett.'15]



Our approach is portfolio optimization

$$\pi = 0.2 \begin{matrix} a \\ b \end{matrix} + 0.3 \begin{matrix} a \\ c \end{matrix} + 0.5 \begin{matrix} d \\ e \end{matrix}$$

investment **asset**



Return = $0.2 \cdot 5 + 0.3 \cdot 4 + 0.5 \cdot 6 = \mathbf{5.2}$

We found poly-time approximation is possible!!

Our contributions

① Formulation

Portfolio optimization for maximizing CVaR

② Algorithm

Constant additive error in poly-time

③ Experiments

Our portfolio outperforms baselines

Related work

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

- ▶ Find smallest A s.t. VaR $\geq$ thld.

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

- ▶ Find A s.t. (# vertices influenced by A *w.h.p.*) $\geq$ thld.
- ▶ No approx. guarantee

Robust influence maximization

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

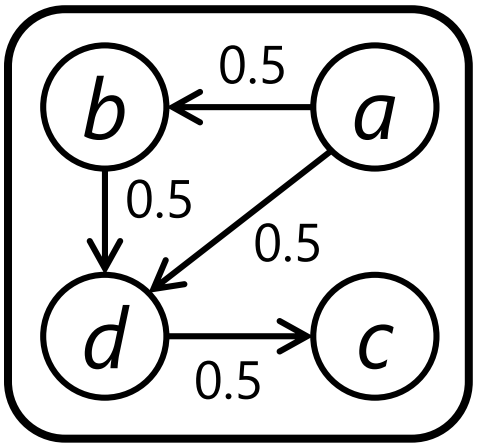
- ▶ Model parameters are noisy or uncertain
- ▶ Robustness is measured by $\mathbf{E}$[cascade size]

These are single set selection problems

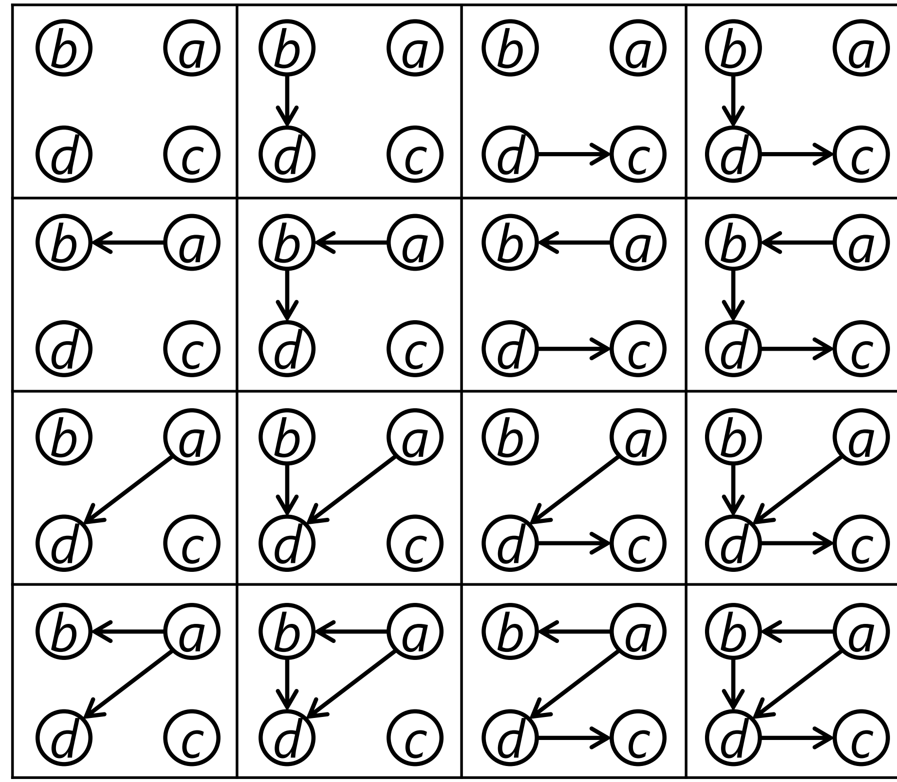
Formulation

Diffusion process [Goldenberg-Libai-Muller. Market. Lett.'01]

Graph $G = (V, E)$
Edge prob. $p: E \to [0,1]$



uv lives w.p. p_{uv}
$2^{|E|}$ outcomes



A influences v in the diffusion process ⟺ A can reach v in the random graph

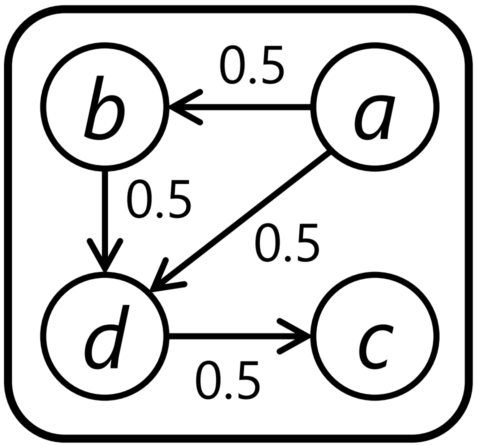
X_A = r.v. for # vertices reachable from A in the random graph

Formulation

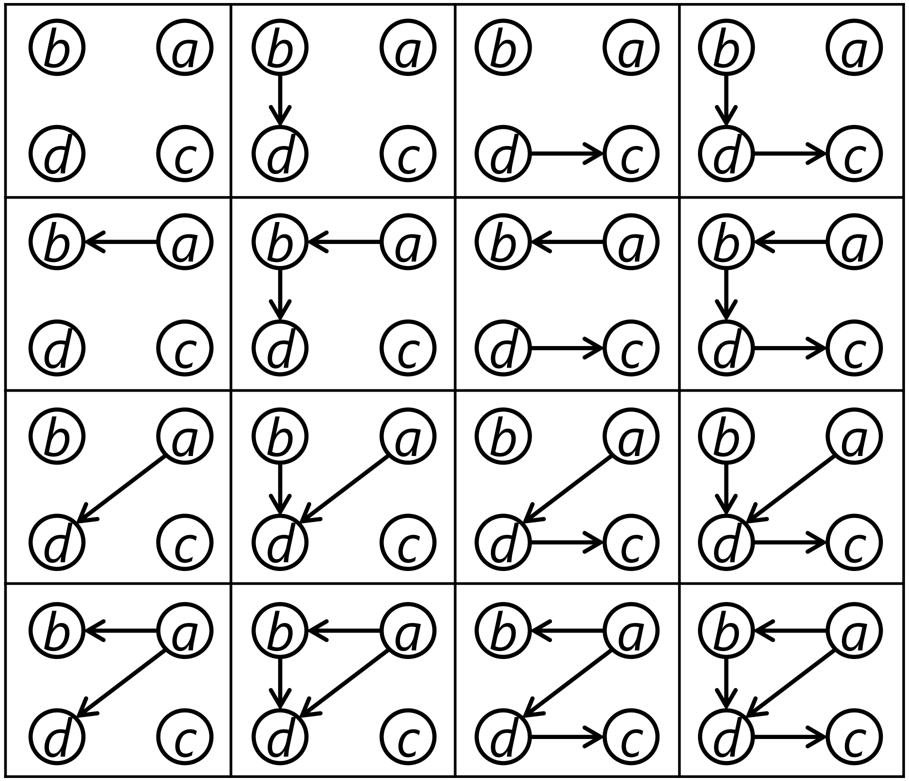
k-vertex **portfolio**, $\mathbf{E}[\cdot]$ and $\mathrm{CVaR}_\alpha[\cdot]$

$\boldsymbol{\pi}$: $\binom{|V|}{k}$-dim. vector
s.t. $\|\boldsymbol{\pi}\|_1 = 1$

E.g., $k = 1$, $\pi_a = \pi_b = 0.5$



uv lives w.p. p_{uv}
$2^{|E|}$ outcomes



Dist. of $\langle \boldsymbol{\pi}, \mathbf{X} \rangle = \pi_a X_a + \pi_b X_b$

$$\mathbf{E}[\pi_a X_a + \pi_b X_b] = \mathbf{2.25}$$

$$\mathrm{CVaR}_{0.25}[\pi_a X_a + \pi_b X_b] = \underline{\mathbf{1.25}}$$ worst 0.25-fraction

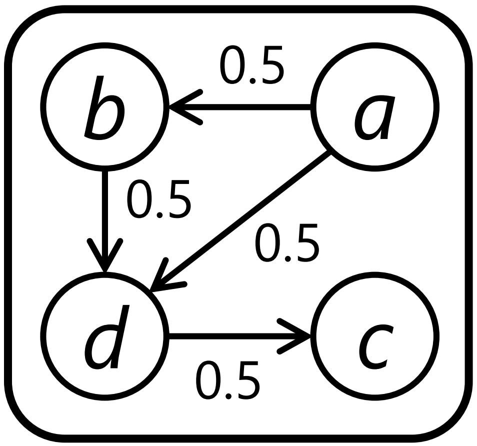
Formulation

k-vertex **portfolio**, $\mathbf{E}[\cdot]$ and $\mathrm{CVaR}_\alpha[\cdot]$

$\boldsymbol{\pi}$: $\binom{|V|}{k}$-dim. vector

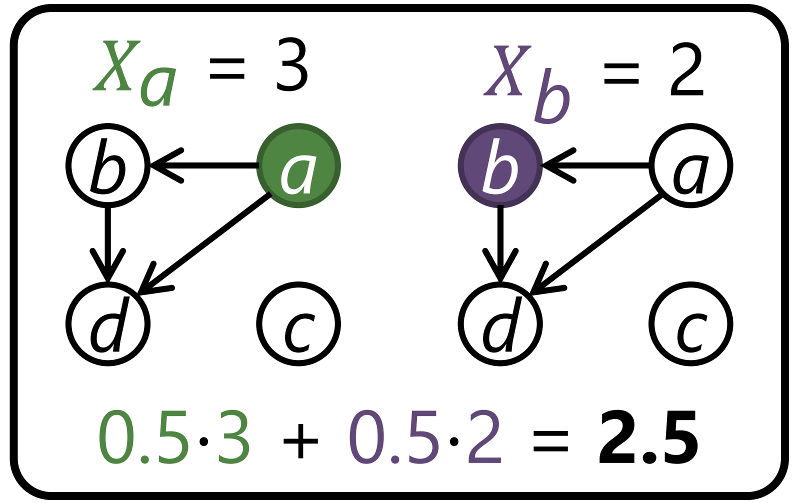
s.t. $\|\boldsymbol{\pi}\|_1 = 1$

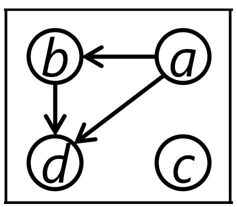
E.g., $k = 1$, $\pi_a = \pi_b = 0.5$



uv lives w.p. p_{uv}

$2^{|E|}$ outcomes





Dist. of $\langle \boldsymbol{\pi}, \mathbf{X} \rangle = \pi_a X_a + \pi_b X_b$

$$\mathbf{E}[\pi_a X_a + \pi_b X_b] = \mathbf{2.25}$$

$$\mathrm{CVaR}_{0.25}[\pi_a X_a + \pi_b X_b] = \underline{\mathbf{1.25}}$$ worst 0.25-fraction

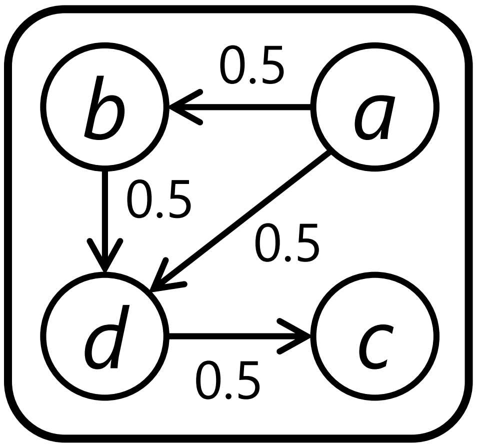
Formulation

k-vertex **portfolio**, $\mathbf{E}[\cdot]$ and $\mathrm{CVaR}_\alpha[\cdot]$

$\boldsymbol{\pi}$: $\binom{|V|}{k}$-dim. vector

s.t. $\|\boldsymbol{\pi}\|_1 = 1$

E.g., $k = 1$, $\pi_a = \pi_b = 0.5$



uv lives w.p. p_{uv}

$2^{|E|}$ outcomes

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**

Dist. of $\langle \boldsymbol{\pi}, \mathbf{X} \rangle = \pi_a X_a + \pi_b X_b$

$$\mathbf{E}[\pi_a X_a + \pi_b X_b] = \mathbf{2.25}$$

$$\mathrm{CVaR}_{0.25}[\pi_a X_a + \pi_b X_b] = \mathbf{1.25}$$ worst 0.25-fraction

Formulation

Our problem definition

Input : $G = (V, E)$, p, integer k, significance level α

Output : k-vertex portfolio $\boldsymbol{\pi}$

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

$$= \sum_{A:|A|=k} \pi_A X_A$$

vertices reachable from A in the random graph

⚠ $\boldsymbol{\pi}$ & $\mathbf{X}$ are $\binom{|V|}{k}$-dimensional

Existing methods cannot be applied

Our algorithm

Main idea

Standard approximation

CVaR optimization ➔ BIG linear programming

Multiplicative weights algorithm [Arora-Hazan-Kale. '12]

Used in optimization, machine learning, game theory, …

E.g., our group's application [Hatano-Yoshida. AAAI'15]

Requires an oracle for

a convex combination of the constraints

↑

Greedy strategy solves approximately!

Our algorithm

Standard approximation as a first step

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

Write CVaR as an optimization problem
[Rockafellar-Uryasev. J. Risk'00]

Sampling $\mathbf{X}^1, \dots, \mathbf{X}^s$ from the dist. of $\mathbf{X}$

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

Solve the feasibility problem "$\square \ge \gamma$?"
to perform the bisection search on γ

Our algorithm

Difficulty of checking "$\square \geq \gamma$?"

"$\square \geq \gamma$?" = Feasibility of a BIG linear programming

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

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

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

$\wr\wr$

" Multiple submodular functions exceed a threshold simultaneously "

\# variables $\approx \binom{|V|}{k}$

\# constraints $= s$

HARD to satisfy!!

Our algorithm

Our solution for checking the BIG LP □

Multiplicative weights algorithm [Arora-Hazan-Kale. '12]

① Solve the *convex combination* □ for $\mathbf{p} = \mathbf{p}_1, \dots, \mathbf{p}_T$

② The average solution ≈ A solution of □

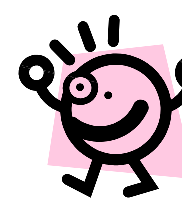
$$\exists ?\, \mathbf{x} \in \mathbf{P}_\gamma \qquad \langle \mathbf{p}, \mathbf{A}\mathbf{x} \rangle \geq \langle \mathbf{p}, \mathbf{b} \rangle$$

≀

Still looks difficult, but ... " A single submodular function exceeds a threshold "

We can assume $\boldsymbol{\pi}$ is sparse ➔ Greedy works!!

Constant additive error ($-\mathrm{e}^{-1}$) in poly-time



Our algorithm

Summary : Time & Quality

Exact CVaR optimization

≀ Error is bounded

Empirical CVaR optimization

↓ Bisection search

BIG linear programming

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

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

↓ Multiplicative weights!

Convex combination

Greedy works!

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

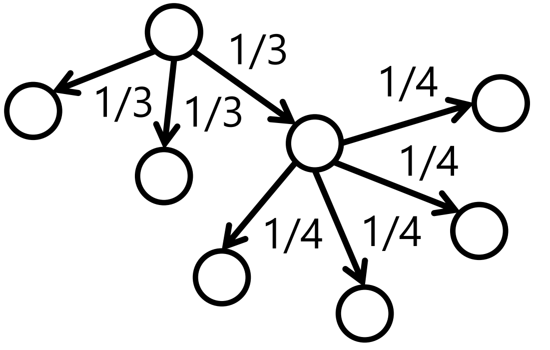
optimum value — additive error

Experiments

Settings

- Test data (see our paper for results on other data)

  Physicians friend network ($|V| = 117$ & $|E| = 542$)
  from [Koblenz Network Collection]
  $p_{uv} = (\text{out}-\text{degree of } u)^{-1}$



- Parameter : $\epsilon = 0.4$

- Baselines : produce a *single* vertex set

Greedy	a standard greedy algorithm for influence maximization
Degree	select k vertices in degree decreasing order
Random	select k vertices uniformly at random

- Environment : Intel Xeon E5-2670 2.60GHz CPU, 512GB RAM

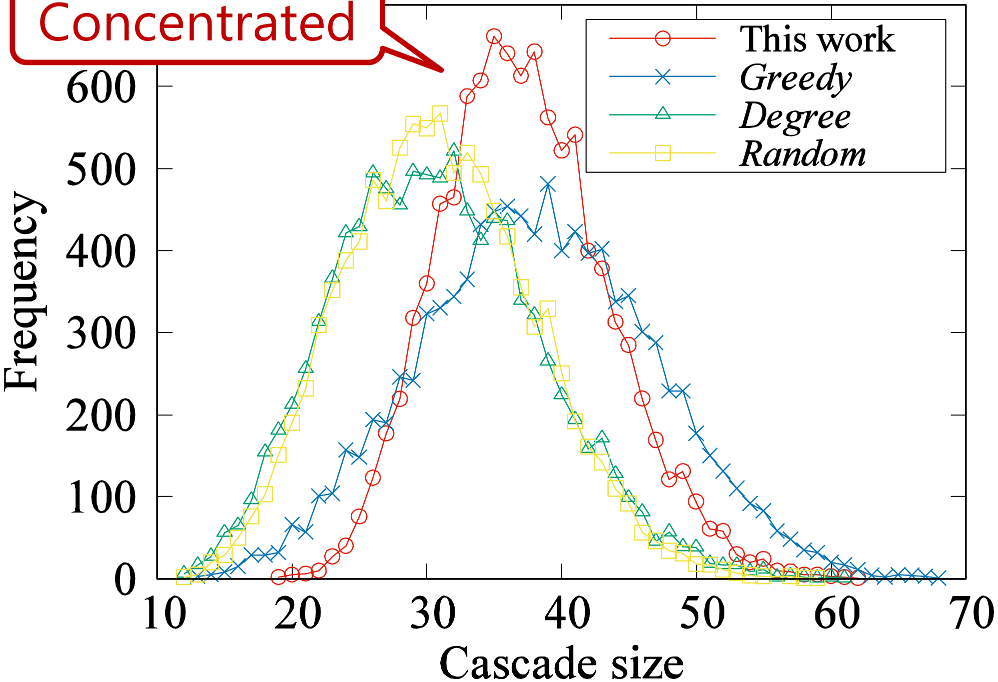
Experiments

Results for $k = 10$ & $\alpha = 0.01$

	CVaR at α	Expectation	Runtime
This work	**23.6**	37.7	157.2s
Greedy	16.9	**38.2**	0.1s
Degree	14.2	30.8	0.2ms
Random	15.1	31.0	0.2ms

Comparable

Concentrated



π is extremely sparse!!

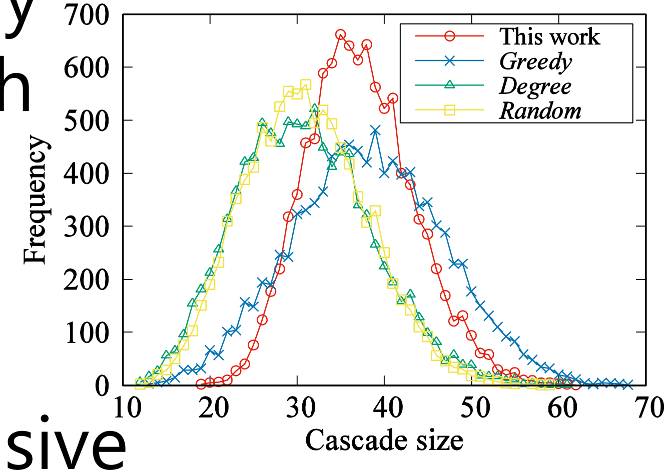
(# non-zeros in **π**) = 42

dim. of **π** = $\binom{|V|}{10} \approx 10^{14}$

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
⋮	⋮

Conclusion

Succeed to obtain a low-risk strategy by a portfolio optimization approach



Future study Ⅰ : **Speed-up**

$\tilde{O}(\epsilon^{-6}k^2|V||E|)$ time is still expensive

Can we solve the BIG LP in a different way?

Future study Ⅱ : **Exact computation of CVaR**

Is it possible to extend [Maehara-Suzuki-Ishihata. WWW'17] ?

Future study Ⅲ : **Other risk measures**

Value at Risk, Lower partial moment, ...