Monotone k-Submodular Function Maximization with Size Constraints

Naoto Ohsaka (UTokyo)
Joint work with Yuichi Yoshida (NII & PFI)

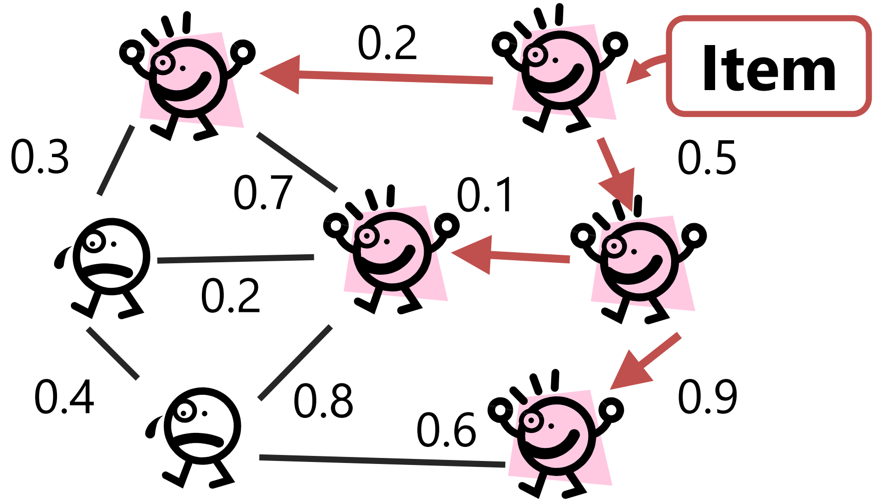
My research interests:

Graph algorithms in practice & theory

Work in [Ohsaka,Akiba,Yoshida,Kawarabayashi. AAAI'14]

Fast algorithm for influence maximization

[Kempe,Kleinberg,Tardos. KDD'03]



Q. How to distribute free items to B people to maximize the spread of influence?

(e.g., rumors & product adoptions)

Instance of monotone submodular maximization

Monotone submodular maximization

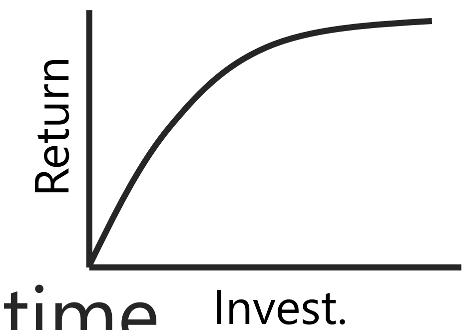
Monotonicity:

$$f(S) \leq f(T) \text{ for } S \subseteq T$$

Submodularity (diminishing return):

$$f(S+e) - f(S) \geq f(T+e) - f(T)$$
$$\text{for } S \subseteq T \text{ \& } e \notin T$$

Goal: $\max f(S)\ s.t.\ |S| = B$



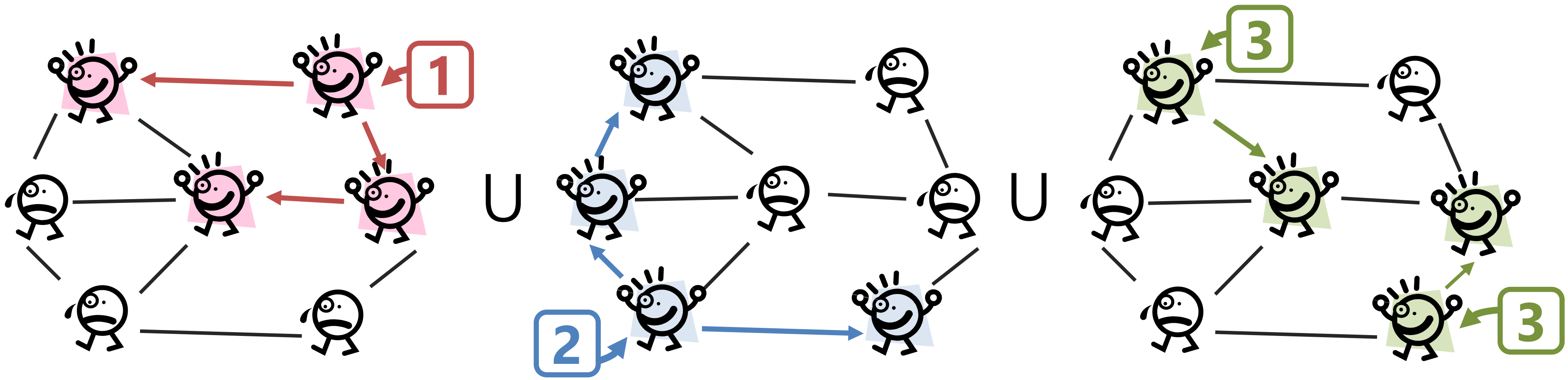
Naturally arise & 0.63-approx. in poly time
[Nemhauser,Wolsey,Fisher. Math. Program. '78]

Document summarization [Lin,Bilmes. ACL-HLT'11]
⇨ Network inference [Gomez-Rodriguez,Leskovec,Krause. KDD'10]
Sensor placement [Krause,Singh,Guestrin. J. Mach. Learn. Res.'08]

Motivation of this research

We often have **multiple** kinds of items 1, 2, 3

Each has a different topic ⇨ different cascades



Q. How to distribute $\boldsymbol{k}$ **kinds** of items to B people?

Can be modeled as $\boldsymbol{k}$**-submodular functions**

Our contributions

Presented in [Ohsaka,Yoshida. NIPS'15]

Monotone k-submodular function maximization under

Total size constraint	**Individual size constraint**
#1 + #2 + #3 = B	#1 = B_1 , #2 = B_2, #3 = B_3

Approximation algorithms

Total size constraint	Individual size constraint
Approx. ratio = 1/2	Approx. ratio = 1/3
# function eval. =	# function eval. =
Greedy: Quadratic	Greedy: Quadratic
Randomized: Almost linear	Randomized: Almost linear

Experimental evaluations

Total size constraint	Individual size constraint
Influence maximization with k topics	Sensor placement with kinds of sensors

Related work

Theoretical results under **NO** constraint
[Iwata,Tanigawa,Yoshida. SODA'16]

Non-monotone: 1/2-approx. alg.

Monotone: $\left(\frac{k}{2k-1}\right)$-approx. alg.

$\left(\frac{k+1}{2k}+\epsilon\right)$-approx. needs exponentially many queries

Application of bi(2-)submodular functions
[Singh,Guillory,Bilmes. AISTATS'12]

Sensor placement & feature selection

Function form

We consider a function of form

$$f:\{0,1,\ldots,k\}^V \to \mathbb{R}$$
$$|V| = n$$

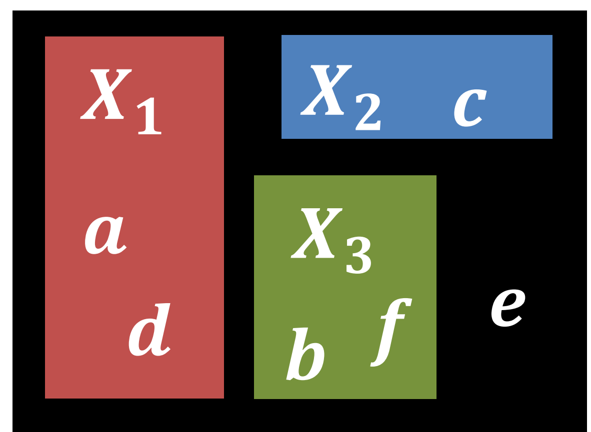
e	a	b	c	d	e	f
$x(e)$	1	3	2	1	0	3

one-to-one correspondence

Another interpretation

$$f:\underline{(k+1)^V} \to \mathbb{R}$$

the family of
k disjoint subsets of V



Characterization of k-submodular functions

[Huber,Kolmogorov. ISCO'12] [Ward,Živný. ACM Trans. Algor. '15]

f is ***monotone k-submodular*** iff ① & ②

① Monotone

$$\Delta_{e,i} f(\boldsymbol{x}) \geq 0 \text{ for } \boldsymbol{x} \text{ s.t. } \boldsymbol{x}(e) = 0$$

$$\boxed{0 \mid 1 \mid \boxed{1} \mid 3} - \boxed{0 \mid 1 \mid 0 \mid 3} \geq 0$$

② Orthant submodular

$$\Delta_{e,i} f(\boldsymbol{x}) \geq \Delta_{e,i} f(\boldsymbol{y}) \text{ for } \boldsymbol{x} \preccurlyeq \boldsymbol{y} \text{ s.t. } \boldsymbol{y}(e) = 0$$

$$\boxed{0 \mid 1 \mid \boxed{2} \mid 0} - \underbrace{\boxed{0 \mid 1 \mid 0 \mid 0}}_{\boldsymbol{x}} \geq \boxed{2 \mid 1 \mid \boxed{2} \mid 3} - \underbrace{\boxed{2 \mid 1 \mid 0 \mid 3}}_{\boldsymbol{y}}$$

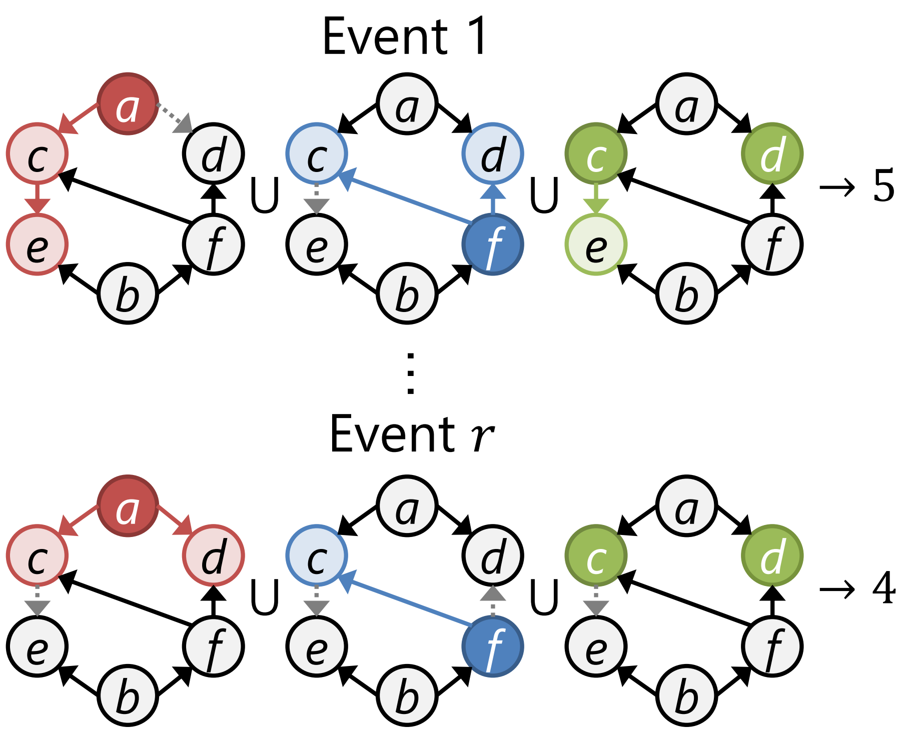
$\Delta_{e,i} f(\boldsymbol{x})$ = change of f by assigning i to e^{th} element

Modeling influence maximization with k topics

We have k graphs $G_1, \ldots, G_k$ and a *strategy* $\boldsymbol{s} \in (k+1)^V$

a	*b*	*c*	*d*	*e*	*f*
1	0	3	3	0	2

Influence from v ($\boldsymbol{s}(v) = i$) *stochastically* spreads in G_i



$f(\boldsymbol{s}) :=$
Exp. # vertices influenced in one of the k graphs

f is
Monotone & k-submodular

Monotone submodular maximization by greedy algorithm with lazy evaluations

[Nemhauser,Wolsey,Fisher. Math. Program. '78] [Minoux. '78]

$$\begin{array}{l} \textbf{for } j = 1 \textbf{ to } B \\ \quad e_j \leftarrow \underset{e}{\operatorname{argmax}} \Delta_e(S_{j-1}) := f(S_{j-1} + e) - f(S_{j-1}) \\ \quad S_j \leftarrow S_{j-1} + e_j \end{array}$$

j	S_j	$\Delta_a(S_{j-1})$	$\Delta_b(S_{j-1})$	$\Delta_c(S_{j-1})$	$\Delta_d(S_{j-1})$	$\Delta_e(S_{j-1})$
1	$\{b\}$	**7**	$\boxed{\mathbf{10}}$	**3**	**8**	**6**
2	$\{b, e\}$	—	—	$\le$ **3**	**2**	$\boxed{\mathbf{5}}$
3		—	—	$\le$ **3**	$\le$ **2**	—

Greedy algorithm for the total size constraint

Constraint: $\#1 + \#2 + \#3 + \cdots + \#\mathbf{k} = B$

k-Greedy-TS (*total size*)

$\boldsymbol{s}_0 \leftarrow \mathbf{0}$
for $j = 1$ **to** B
$\quad (e_j, i_j) \leftarrow \operatorname*{argmax}_{e:\boldsymbol{s}_{j-1}(e)=0,\, i} \Delta_{e,i} f(\boldsymbol{s}_{j-1})$
$\quad \boldsymbol{s}_j \leftarrow$ assign i_j to the e_jst element in $\boldsymbol{s}_{j-1}$

- # function eval. $= \boldsymbol{O(knB)}$
- Approx. ratio $= \mathbf{1/2}$

Proof sketch

		a	b	c	d	e	f
optimal	$\boldsymbol{o}_0 = \boldsymbol{o}^*$	2	0	4	2	3	0
	$\boldsymbol{o}_1$	3					
Swap	$\boldsymbol{o}_2$	3	1				
	$\boldsymbol{o}_3$	3	1	4			
output	$\boldsymbol{o}_4 = \boldsymbol{s}_4$	3	1	4	1	0	0
	$\boldsymbol{s}_3$	3	1	4	0	0	0
Greedy	$\boldsymbol{s}_2$	3	1	0	0	0	0
	$\boldsymbol{s}_1$	3	0	0	0	0	0
zero	$\boldsymbol{s}_0 = \boldsymbol{0}$	0	0	0	0	0	0

?

Construct a sequence from $\boldsymbol{o}^*$ to $\boldsymbol{s}_B$ preserving
$(\#\text{nonzero in } \boldsymbol{o}_i) = B$

$$f(\boldsymbol{s}_j) - f(\boldsymbol{s}_{j-1}) \geq f(\boldsymbol{o}_{j-1}) - f(\boldsymbol{o}_j)$$

Proof sketch

optimal	$\boldsymbol{o}_0 = \boldsymbol{o}^*$
Swap	$\boldsymbol{o}_1$
	$\boldsymbol{o}_2$
	$\boldsymbol{o}_3$
output	$\boldsymbol{o}_4 = \boldsymbol{s}_4$
Greedy	$\boldsymbol{s}_3$
	$\boldsymbol{s}_2$
	$\boldsymbol{s}_1$
zero	$\boldsymbol{s}_0 = \boldsymbol{0}$

$$f(\boldsymbol{o}^*) - f(\boldsymbol{s})$$

$$\rotatebox{90}{$\geq$}$$

$$f(\boldsymbol{s}) - f(\boldsymbol{0})$$

$$\rotatebox{90}{$\geq$}$$

$$f(\boldsymbol{s})$$

$$\boxed{\boldsymbol{f(s) \geq \frac{1}{2} f(o^*)}}$$

Speeding-up by random sampling

— $O(knB)$ may be quadratic (e.g., $B = n/2$)

Can be reduced?

Combining **random sampling**

[Mirzasoleiman,Badanidiyuru,Karbasi,Vondrák,Krause. AAAI'15]

At step j:

Looking at ~~all elements~~ a single **random** element

$W.p. \geq \frac{B-j+1}{n-j+1}, \quad f(\boldsymbol{s}_j) - f(\boldsymbol{s}_{j-1}) \geq f(\boldsymbol{o}_{j-1}) - f(\boldsymbol{o}_j)$

Looking more elements gains success probability

Almost linear-time algorithm for the total size constraint

k-Stochastic-Greedy-TS

$\boldsymbol{s}_0 \leftarrow \mathbf{0}$
for $j = 1$ **to** B
    $R_j \leftarrow$ a random subset of size $\min\left\{\frac{n-j+1}{B-j+1}\log\frac{B}{\delta}, n\right\}$
    $(e_j, i_j) \leftarrow \operatorname*{argmax}_{e\in R_j:\boldsymbol{s}_{j-1}(e)=0,i} \Delta_{e,i} f(\boldsymbol{s}_{j-1})$
    $\boldsymbol{s}_j \leftarrow$ assign i_j to the e_jst element in $\boldsymbol{s}_{j-1}$

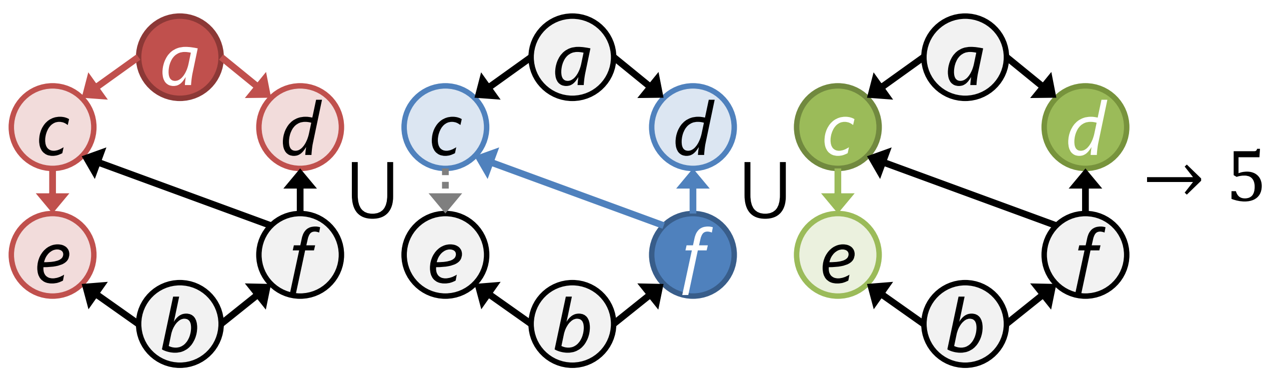
- # function eval. $= \boldsymbol{O(kn\log(B)\cdot\log(B/\delta))}$
- Approx. ratio $= \mathbf{1/2}$ ***w.p.*** $\mathbf{1-\delta}$

Influence maximization with k topics

Given: k graphs $G_1, \dots, G_k$ & a budget B

Goal: $\max f(\boldsymbol{s})$ *s.t.* #**1** + #**2** + #**3** + ⋯ + #**k** = B

a	b	c	d	e	f
1	0	3	3	0	2



$f(\boldsymbol{s}) :=$ Exp. # vertices influenced in one of the k graphs

Exact computation of $f(\cdot)$ is #P-hard [Chen,Wang,Wang. KDD'10]
Averaging 100 simulations

Settings

Digg dataset (a social news website)
http://www.isi.edu/~lerman/downloads/digg2009.html

- A friendship network

  $n = 3{,}522$ vertices & $90{,}244$ edges
- A log of user votes for stories

Learning parameters

- Using the method by [Barbieri,Bonchi,Manco. ICDM'12]
- Edge probabilities in $k = 10$ graphs

Algorithms

$\boldsymbol{k}$-**Greedy-TS** w/ lazy evaluations

$\boldsymbol{k}$-**Stochastic-Greedy-TS** $(\delta = 0.1)$ w/ lazy evaluations

Baselines

Single$(\boldsymbol{i})$

Greedy strategy that considers only the i^{th} topic

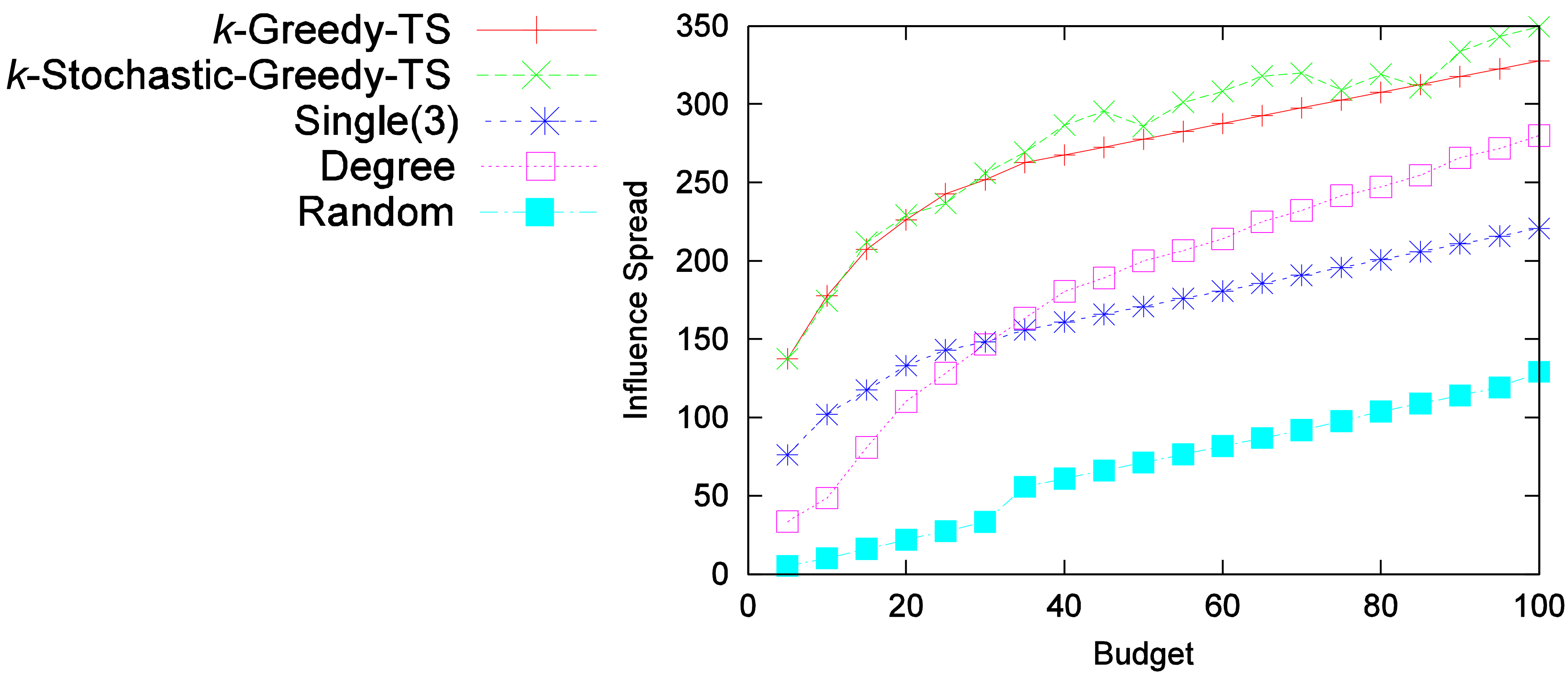
Degree

Select vertices of *high degree* & assign *random* topics

Random

Select vertices *randomly* & assign *random* topics

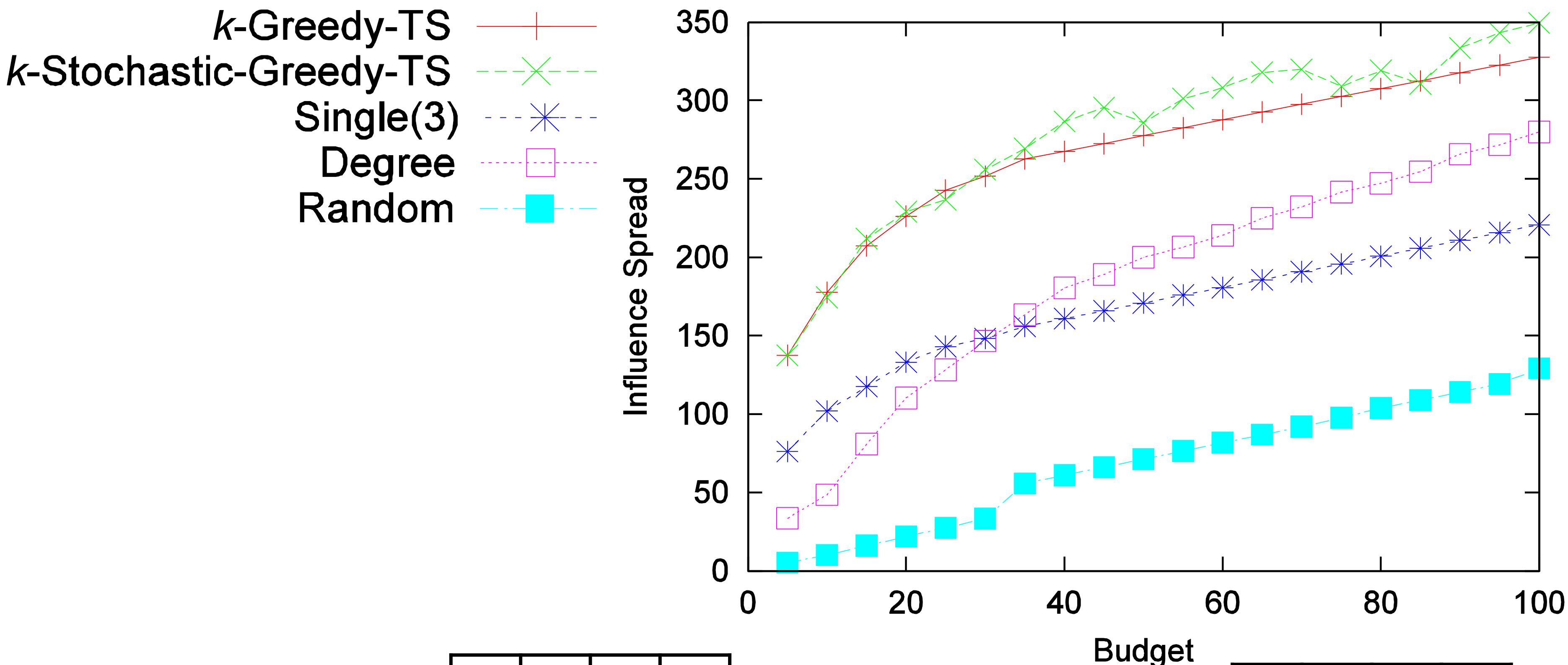
Result: Objective function value



Our methods outperformed baselines

Environment: Intel Xeon E5-2690 2.90GHz CPU with 256GB memory

Result: Objective function value



3	3	0	3

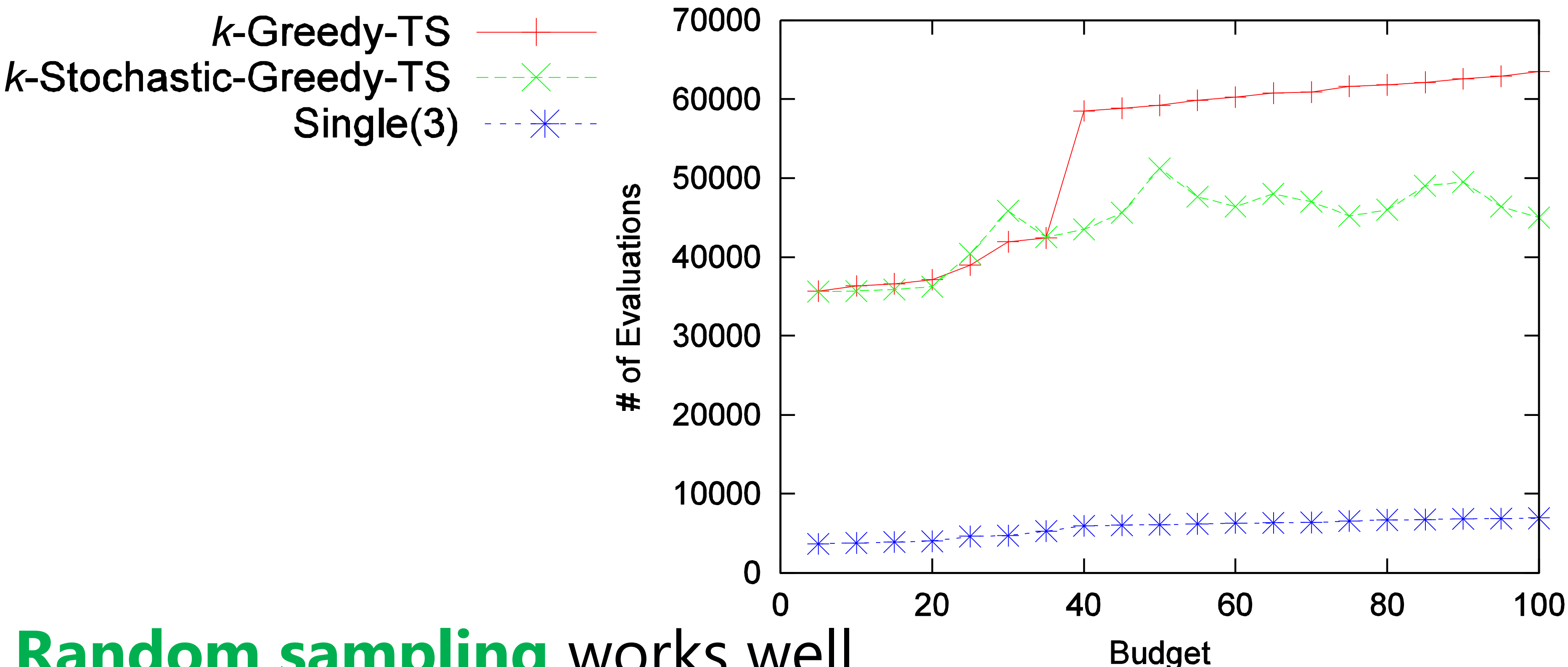
0	1	5	3

Single(3) was worse than **Degree**

(Effectiveness of assigning multiple kinds of items)

Environment: Intel Xeon E5-2690 2.90GHz CPU with 256GB memory

Result: # function evaluations



Random sampling works well

k-Greedy-TS rapidly grows

∵ (e, i) gives similar marginal gains at 40th iteration

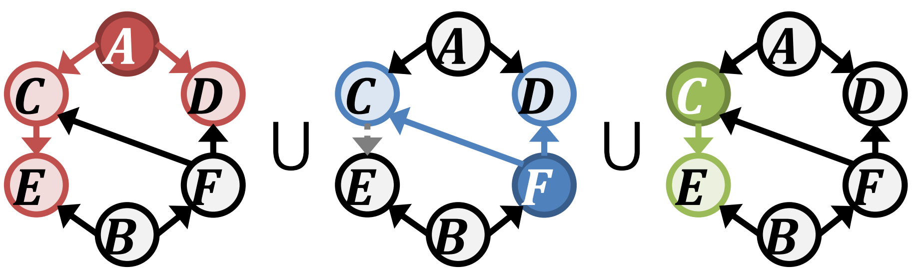
⇨ we need to check $\Delta_{e,i} f(\boldsymbol{s})$ for most of the (e, i)'s

Environment: Intel Xeon E5-2690 2.90GHz CPU with 256GB memory

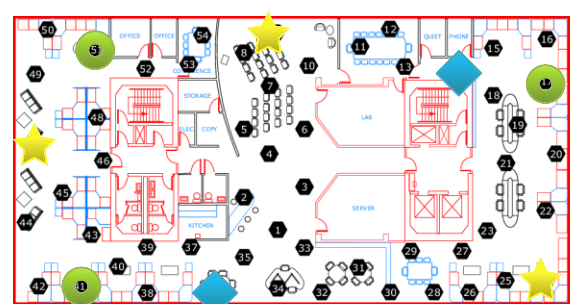
Summary

Monotone k-submodular function maximization

Constraint	Approx. ratio	# function eval.
Total size $\#1 + \#2 + \#3 = B$	$1/2$	$O(knB)$
	$1/2$ *w.p.* $1-\delta$	$\tilde{O}(kn)$
Individual size $\#1 = B_1$, $\#2 = B_2$, $\#3 = B_3$	$1/3$	$O(kn\sum_i B_i)$
	$1/3$ w.p. $1-\delta$	$\tilde{O}(k^2 n)$



Influence maximization with k topics



Sensor placement with k kinds of sensors