Subset Selection by Pareto Optimization: Theories and Practical Algorithms

Chao Qian

UBRI, School of Computer Science and Technology
University of Science and Technology of China

http://staff.ustc.edu.cn/~chaoqian/
Email: chaoqian@ustc.edu.cn

Outline

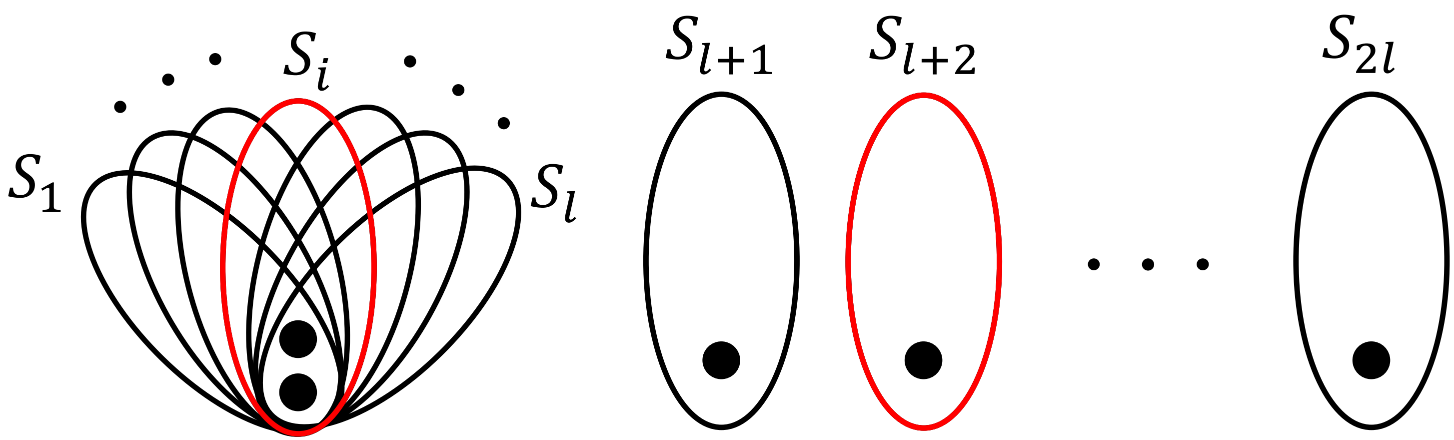
- **Introduction**
- Pareto optimization for subset selection
- Pareto optimization for large-scale subset selection
- Pareto optimization for noisy subset selection
- Conclusion

Maximum coverage

Maximum coverage [Feige, JACM'98] : select at most B sets from n given sets to make the union maximal

Formally stated: given a ground set U, a collection $V = \{S_1, \dots, S_n\}$ of subsets of U and a budget B, it is to find a subset $X \subseteq V$ such that

$$max_{X \subseteq V} \quad f(X) = |\cup_{S_i \in X} S_i| \quad s.t. \quad |X| \le B.$$



Sparse regression

Sparse regression [Tropp, TIT'04] : find a sparse approximation solution to the linear regression problem

Formally stated: given all observation variables $V = \{v_1, \ldots, v_n\}$, a predictor variable z and a budget B, it is to find a subset $X \subseteq V$ such that

$$max_{X \subseteq V} \quad R^2_{z,X} = \frac{\mathrm{Var}(z) - \mathrm{MSE}_{z,X}}{\mathrm{Var}(z)} \quad s.t. \quad |X| \leq B.$$

	Corr.	Dis.	LR	…	…	AIC.	BIC	RF.
x1	0.28	0.46	1	…	…	0.22	0.63	1
x2	0.31	0.59	0.64	…	…	0.58	0.56	1
x3	0.11	0.02	0.53	…	…	0.43	0.01	1
x4	0.1	0.1	0.64	…	…	0.73	0.92	1
x5	0.02	0.15	0.33	…	…	0.56	0.36	0.78
x6	0.36	0.02	0.01	…	…	0.32	0.02	0.22
x7	0.2	0.2	0.21	…	…	0.21	0.02	0.11
x8	0.1	0.03	0.32	…	…	0.33	0.51	0.44
x9	0.32	0.1	0.2	…	…	0.06	0.66	0
x10	0.24	0	0.02	…	…	0.6	0.03	0.33
x11	0.12	0.45	0.44	…	…	0.64	0.45	1
x12	0.36	0.58	0.12	…	…	0.73	0.58	0.67
x13	0.2	0.02	0.24	…	…	0.34	0.02	0.89
x14	0.24	0.92	0.33	…	…	0.24	0.93	0.56

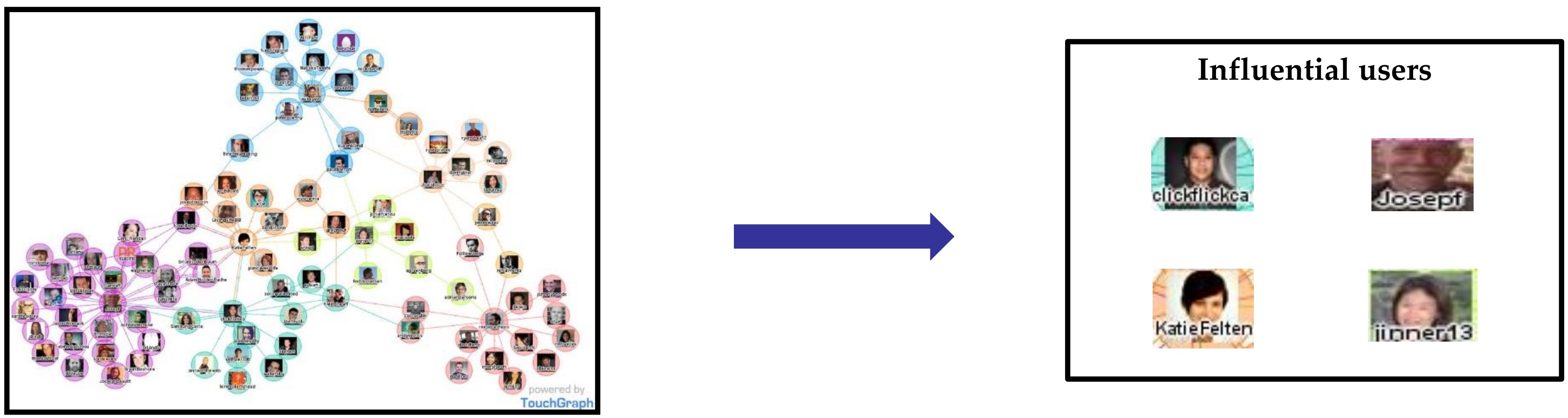
	Corr.	Dis.	LR	…	…	AIC.	BIC	RF.
x1	0.28	0.46	1	…	…	0.22	0.63	1
x2	0.31	0.59	0.64	…	…	0.58	0.56	1
x3	0.11	0.02	0.53	…	…	0.43	0.01	1
x4	0.1	0.1	0.64	…	…	0.73	0.92	1
x5	0.02	0.15	0.33	…	…	0.56	0.36	0.78
x6	0.36	0.02	0.01	…	…	0.32	0.02	0.22
x7	0.2	0.2	0.21	…	…	0.21	0.02	0.11
x8	0.1	0.03	0.32	…	…	0.33	0.51	0.44
x9	0.32	0.1	0.2	…	…	0.06	0.66	0
x10	0.24	0	0.02	…	…	0.6	0.03	0.33
x11	0.12	0.45	0.44	…	…	0.64	0.45	1
x12	0.36	0.58	0.12	…	…	0.73	0.58	0.67
x13	0.2	0.02	0.24	…	…	0.34	0.02	0.89
x14	0.24	0.92	0.33	…	…	0.24	0.93	0.56

Influence maximization

Influence maximization [Kempe et al., KDD'03] : select a subset of users from a social network to maximize its influence spread

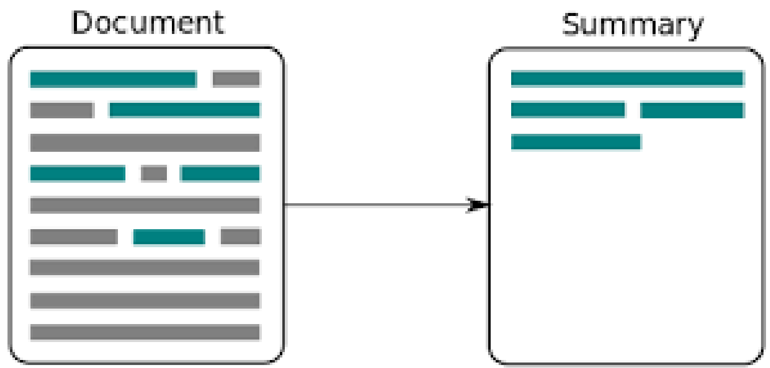
Formally stated: given a directed graph $G = (V, E)$ with $V = \{v_1, \ldots, v_n\}$, edge probabilities $p_{u,v}$ $((u, v) \in E)$ and a budget B, it is to find a subset $X \subseteq V$ such that

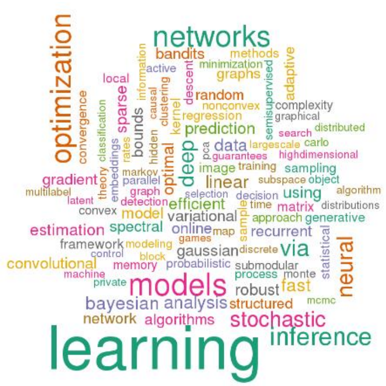
$$max_{X \subseteq V} \quad f(X) = \sum_{i=1}^{n} p(X \to v_i) \quad s.t. \quad |X| \le B.$$



Document summarization

Document summarization [Lin & Bilmes, ACL'11]: select a few sentences to best summarize the documents

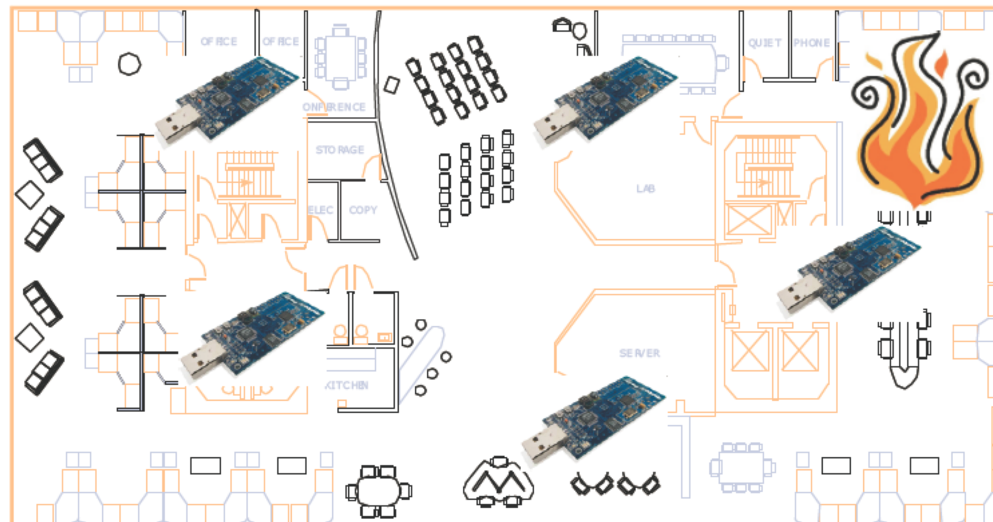




Sensor placement

Sensor placement [Krause & Guestrin, IJCAI'09 Tutorial] : select a few places to install sensors such that the information gathered is maximized

Water contamination detection



Fire detection

Subset selection

Subset selection is to select a subset of size B from a total set of n items for optimizing some objective function

Formally stated: given all items $V=\{v_1,\ldots,v_n\}$, an objective function $f: 2^V \to \mathrm{R}$ and a budget B, it is to find a subset $X \subseteq V$ such that

$$max_{X\subseteq V} \quad f(X) \quad s.t. \quad |X| \leq B.$$

Application	$\boldsymbol{v_i}$	$\boldsymbol{f}$
maximum coverage	a set of elements	size of the union
sparse reg		MSE of prediction
influence max		influence spread
document summarization	a sentence	summary quality
sensor placement	a place to install a sensor	entropy

Many applications, but NP-hard in general!

Subset selection - submodular

Subset selection: given all items $V = \{v_1, \dots, v_n\}$, an objective function $f: 2^V \to \mathrm{R}$ and a budget B, it is to find a subset $X \subseteq V$ such that

$$max_{X \subseteq V} \quad f(X) \quad s.t. \quad |X| \leq B.$$

Monotone: for any $X \subseteq Y \subseteq V$, $f(X) \leq f(Y)$

Submodular [Nemhauser et al., MP'78] : satisfy the natural diminishing returns property, i.e., for any $X \subseteq Y \subseteq V$, $v \notin Y$,

Discrete analogue of convexity!

$$f(X \cup \{v\}) - f(X) \geq f(Y \cup \{v\}) - f(Y)$$

Submodular ratio [Zhang & Vorobeychi, AAAI'16] :

$$\gamma_f = \min_{X \subseteq Y, v \notin Y} \frac{f(X \cup \{v\}) - f(X)}{f(Y \cup \{v\}) - f(Y)}$$

The optimal approximation guarantee [Nemhauser & Wolsey, MOR'78] :

$1 - 1/e \approx 0.632$ by the greedy algorithm

Variants of subset selection

- Monotone set function maximization with size constraints

$$max_{X\subseteq V} \quad f(X) \quad s.t. \quad |X| \le B$$

$1-1/e^{\gamma}$

[Das & Kempe, ICML'11]

- Monotone set function maximization with general constraints

$$|X| \le B \;\rightarrow\; c(X) \le B$$

$(\gamma/2)\,(1-1/e^{\gamma})$

[Zhang & Vorobeychik, AAAI'16]

- Monotone multiset function maximization with size constraints

X: a subset → a multiset

$(1-1/e)/2$

[Soma et al., ICML'14]

- Monotone k-submodular function maximization with size constraints

X: a subset → k subsets

$1/2$

[Ohsaka & Yoshida, NIPS'15]

- Monotone sequence function maximization with size constraints

X: a subset → a sequence

$1-e^{-1/(2\Delta)}$

[Tschiatschek et al, AAAI'17]

- Ratio optimization of monotone functions

$$min_{X\subseteq V} \quad f(X)/g(X)$$

$$\frac{|X^*|}{(1+(|X^*|-1)(1-\kappa))\gamma}$$

[Bai et al., ICML'16]

Previous approaches

- Greedy algorithms

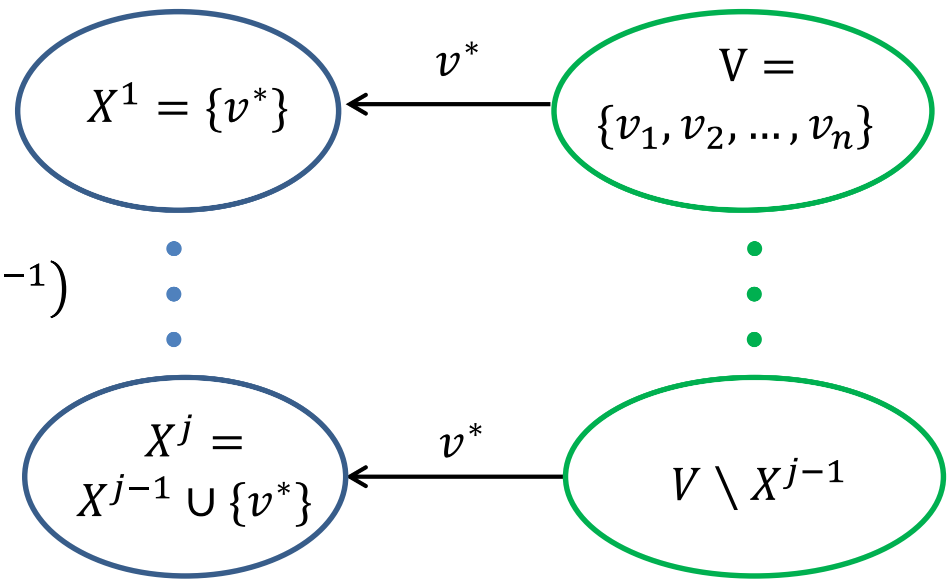
  Process: iteratively select one item that makes some criterion currently optimized

$$max_{X \subseteq V} \quad f(X) \quad s.t. \quad |X| \le B$$

Iteration 1:

$v^* =$
$\arg max_{v \in V \setminus X^{j-1}} f\left(X^{j-1} \cup \{v\}\right) - f\left(X^{j-1}\right)$

Iteration j:



Previous approaches

- Greedy algorithms

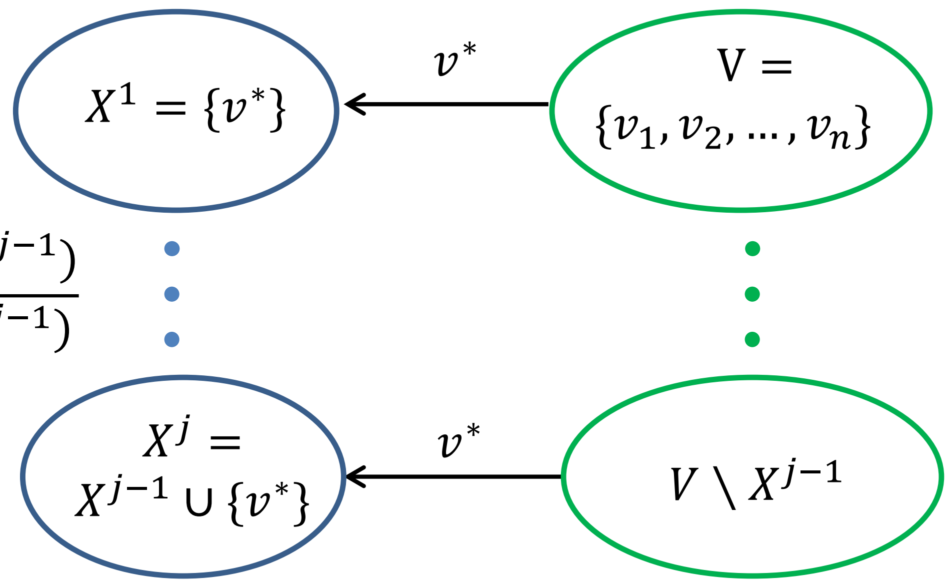
Process: iteratively select one item that makes some criterion currently optimized

$$max_{X \subseteq V} \quad f(X) \quad s.t. \quad c(X) \leq B$$

Iteration 1:

$$v^* = \arg max_{v \in V \setminus X^{j-1}} \frac{f\left(X^{j-1} \cup \{v\}\right) - f\left(X^{j-1}\right)}{c\left(X^{j-1} \cup \{v\}\right) - c\left(X^{j-1}\right)}$$

Iteration j:



Previous approaches

- Greedy algorithms

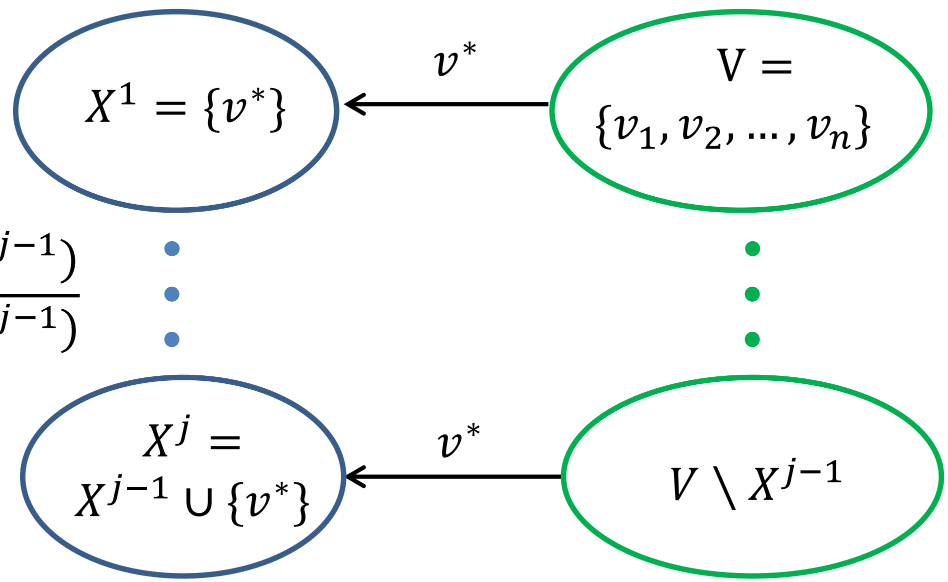
Process: iteratively select one item that makes some criterion currently optimized

$$min_{X \subseteq V} \quad f(X)/g(X)$$

Iteration 1:

$$v^* = \arg min_{v \in V \setminus X^{j-1}} \frac{f\left(X^{j-1} \cup \{v\}\right) - f(X^{j-1})}{g(X^{j-1} \cup \{v\}) - g(X^{j-1})}$$

Iteration j:



Previous approaches

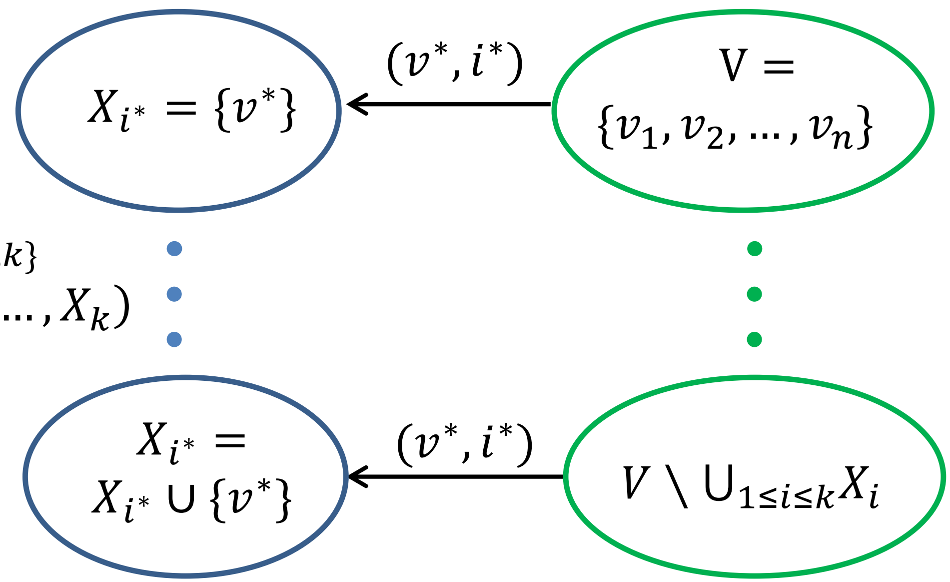
- Greedy algorithms

Process: iteratively select one item that makes some criterion currently optimized

$$max_{X_1,X_2,\dots,X_k \subseteq V} \quad f(X_1,X_2,\dots,X_k) \;\; s.t. \quad |\cup_{1\le i\le k} X_i| \le B$$

Iteration 1:

Iteration j:



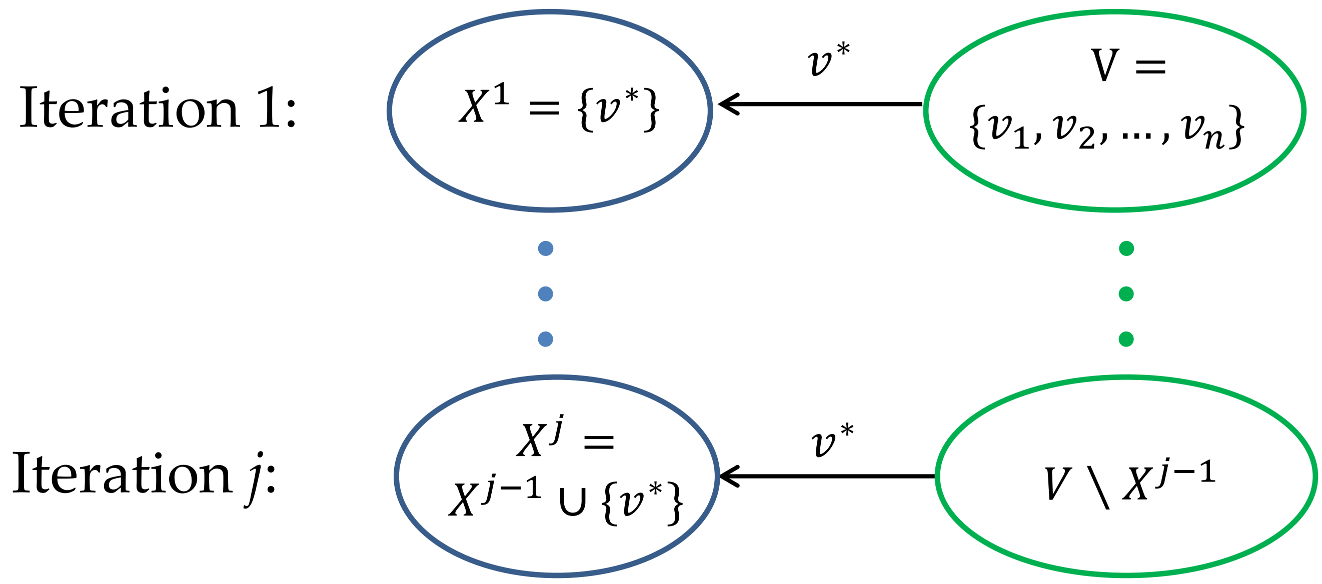
$$(v^*, i^*) = \arg max_{v \in V\setminus \cup_i X_i,\, i\in\{1,2,\dots,k\}}$$
$$f(X_1,\dots,X_i \cup \{v\},\dots,X_k) - f(X_1,\dots,X_i,\dots,X_k)$$

Previous approaches

- Greedy algorithms

Process: iteratively select one item that makes some criterion currently optimized

Weakness: get stuck in local optima due to the greedy behavior



Previous approaches (con't)

- Relaxation methods

Process: relax the original problem, then find the optimal solutions to the relaxed problem

Weakness: the optimal solution of the relaxed problem may be distant to the true optimum

$$max_{X \subseteq V} \quad f(X) \quad s.t. \quad |X| \le B$$

$$\Updownarrow$$

$$max_{w \in R^n} \quad g(w) \quad s.t. \quad |w|_0 \le B \qquad \text{non-convex}$$

$$max_{w \in R^n} \quad g(w) \quad s.t. \quad |w|_1 \le B \qquad \text{convex}$$

Motivation

Subset selection: $max_{x\in\{0,1\}^n}\ f(x)\quad s.t.\ \ |x| \le B$

a subset $X \subseteq V$

Two conflicting objectives:

1. Optimize the criterion $max_{x\in\{0,1\}^n}\ f(x)$
2. Keep the size small $min_{x\in\{0,1\}^n}\ max\{|x|-B, 0\}$

Why not directly optimize the bi-objective formulation?

$$min_{x\in\{0,1\}^n}\,(-f(x), |x|)$$

Outline

- Introduction
- **Pareto optimization for subset selection**
- Pareto optimization for large-scale subset selection
- Pareto optimization for noisy subset selection
- Conclusion

Pareto optimization

The basic idea:

$$max_{x\in\{0,1\}^n}\ f(x)\quad s.t.\quad |x|\le B$$

$$max_{x\in\{0,1\}^n}\ f(x)\quad s.t.\quad c(x)\le B$$

$$max_{x\in\{0,1,\dots,k\}^n}\ f(x)\quad s.t.\quad |x|\le B$$

$$min_{x\in\{0,1\}^n}\ f(x)/g(x)$$

$\Rightarrow$ bi-objective optimization

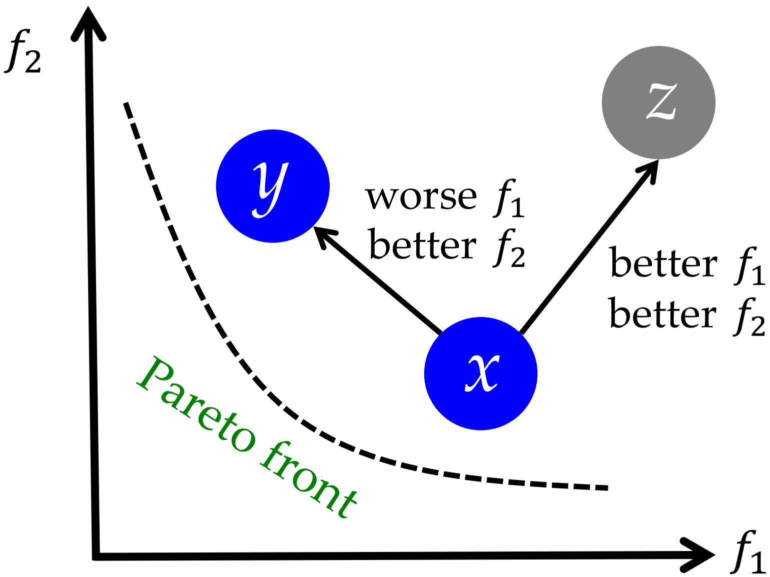
$$min_x\ (f_1(x), f_2(x))$$

x dominates z :

$$f_1(x) < f_1(z) \wedge f_2(x) < f_2(z)$$

x is incomparable with y :

$$f_1(x) > f_1(y) \wedge f_2(x) < f_2(y)$$



Pareto optimization

The basic idea:

$$max_{x\in\{0,1\}^n}\ f(x) \quad s.t. \quad |x| \le B$$

$$max_{x\in\{0,1\}^n}\ f(x) \quad s.t. \quad c(x) \le B$$

$$max_{x\in\{0,1,\dots,k\}^n}\ f(x) \quad s.t. \quad |x| \le B$$

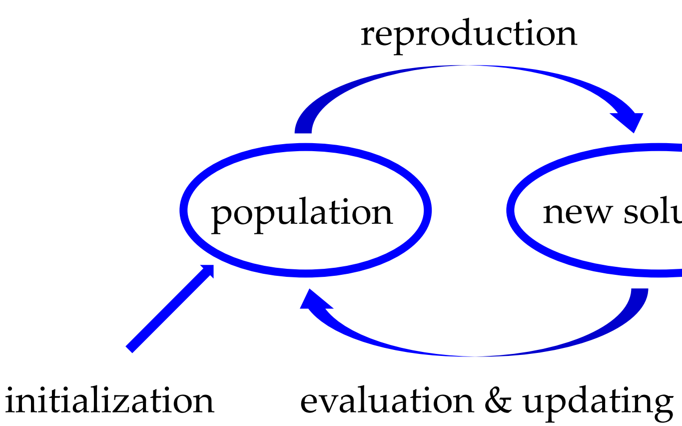
$$min_{x\in\{0,1\}^n}\ f(x)/g(x)$$

How to transform?

bi-objective optimization

$$min_x\ (f_1(x), f_2(x))$$

A simple multi-objective evolutionary algorithm [Laumanns et al., TEvC'04]



Initialization: put a random or special solution into the population P

Reproduction: pick a solution randomly from P, and randomly change it (e.g., flip each bit of $x \in \{0,1\}^n$ with prob. $1/n$)

Evaluation & Updating: if the new solution is not dominated, put it into P and weed out bad solutions

Output: select the best solution w.r.t. the original problem

Pareto optimization vs Greedy algorithms

Greedy algorithms:

- Produce a new solution by adding a single item (single-bit forward search: $0 \rightarrow 1$)
- Maintain only one solution

Pareto optimization:

- Produce a new solution by flipping each bit of a solution with prob. $1/n$ (single-bit forward search, backward search, multi-bit search)
- Maintain several non-dominated solutions due to bi-objective optimization

Pareto optimization may have a better ability of avoiding local optima!

Monotone set function maximization with size constraints

The POSS approach [Qian, Yu and Zhou, NIPS'15]

Transformation:

$$max_{x\in\{0,1\}^n} \ f(x) \quad s.t. \quad |x| \le B \qquad \text{original}$$

$$\Downarrow$$

$$min_{x\in\{0,1\}^n} \ (-f(x), |x|) \qquad \text{bi-objective}$$

Algorithm 1 POSS

Input: all variables $V = \{X_1, \ldots, X_n\}$, a given objective f and an integer parameter $k \in [1, n]$
Parameter: the number of iterations T
Output: a subset of V with at most k variables
Process:
1: Let $s = \{0\}^n$ and $P = \{s\}$.
2: Let $t = 0$.
3: **while** $t < T$ **do**
4: Select s from P uniformly at random.
5: Generate s' by flipping each bit of s with prob. $\frac{1}{n}$.
6: Evaluate $f_1(s')$ and $f_2(s')$.
7: **if** $\nexists z \in P$ such that $z \prec s'$ **then**
8: $Q = \{z \in P \mid s' \preceq z\}$.
9: $P = (P \setminus Q) \cup \{s'\}$.
10: **end if**
11: $t = t + 1$.
12: **end while**
13: **return** $\arg\min_{s \in P, |s| \le k} f_1(s)$

Initialization: put the special solution $\{0\}^n$ into the population P

Reproduction: pick a solution x randomly from P, and flip each bit of x with prob. $1/n$ to produce a new solution

Evaluation & Updating: if the new solution is not dominated, put it into P and weed out bad solutions

Output: select the best feasible solution

Theoretical analysis

POSS can achieve the same general approximation guarantee as the greedy algorithm

Theorem 1. For monotone set function maximization with cardinality constraints, POSS using $\boxed{E[T] \leq 2eB^2n}$ finds a solution x with $|x| \leq B$ and $\boxed{f(x) \geq (1-e^{-\gamma}) \cdot OPT}$.

the expected number of iterations

the best known polynomial-time approximation ratio,
previously obtained by the greedy algorithm [Das & Kempe, ICML'11]

Proof

Lemma 1. For any $X \subseteq V$, there exists one item $\hat{v} \in V \setminus X$ such that

$$f(X \cup \{\hat{v}\}) - f(X) \geq \frac{\boxed{\gamma}}{B}(\boxed{OPT} - f(X))$$

submodularity ratio [Das & Kempe, ICML'11]

the optimal function value

Roughly speaking, the improvement by adding a specific item is proportional to the current distance to the optimum

Proof

Lemma 1. For any $X \subseteq V$, there exists one item $\hat{v} \in V \setminus X$ such that

$$f(X \cup \{\hat{v}\}) - f(X) \geq \frac{\gamma}{B}(OPT - f(X))$$

$\{0,1\}^n$

Main idea:

- consider a solution x with $|x| \leq i$ and $f(x) \geq \left(1 - \left(1 - \frac{\gamma}{B}\right)^i\right) \cdot OPT$

$i = 0$ $\longrightarrow$ $i = B$

initial solution $00 \ldots 0$

$$1 - \left(1 - \frac{\gamma}{B}\right)^B \geq 1 - e^{-\gamma}$$

the desired approximation bound

Proof

Lemma 1. For any $X \subseteq V$, there exists one item $\hat{v} \in V \setminus X$ such that

$$f(X \cup \{\hat{v}\}) - f(X) \geq \frac{\gamma}{B}(OPT - f(X))$$

$\{0,1\}^n$

Main idea:

- consider a solution x with $|x| \leq i$ and $f(x) \geq \left(1 - \left(1 - \frac{\gamma}{B}\right)^i\right) \cdot OPT$
- in each iteration of POSS:
  - select x from the population P, the probability: $1/|P|$
  - flip one specific 0-bit of x to 1-bit, the probability: $\frac{1}{n}\left(1 - \frac{1}{n}\right)^{n-1} \geq \frac{1}{en}$

$$|x'| = |x| + 1 \leq i + 1 \text{ and } f(x') \geq \left(1 - \left(1 - \frac{\gamma}{B}\right)^{i+1}\right) \cdot OPT$$

$i \longrightarrow i + 1$ the probability: $\frac{1}{|P|} \cdot \frac{1}{en}$

Proof

Lemma 1. For any $X \subseteq V$, there exists one item $\hat{v} \in V \setminus X$ such that

$$f(X \cup \{\hat{v}\}) - f(X) \geq \frac{\gamma}{B}(OPT - f(X))$$

$\{0,1\}^n$

Main idea:

- consider a solution x with $|x| \leq i$ and $f(x) \geq \left(1 - \left(1 - \frac{\gamma}{B}\right)^i\right) \cdot OPT$
- in each iteration of POSS:

$i \longrightarrow i+1$ the probability: $\boxed{\frac{1}{|P|} \cdot \frac{1}{en}} \xrightarrow{|P| \leq 2B} \frac{1}{2eBn}$

$i \longrightarrow i+1$ the expected number of iterations: $2eBn$

$i = 0 \longrightarrow B$ the expected number of iterations: $B \cdot 2eBn$

Theoretical analysis

POSS can achieve the same general approximation guarantee as the greedy algorithm

Theorem 1. For monotone set function maximization with cardinality constraints, POSS using $E[T] \leq 2eB^2n$ finds a solution x with $|x| \leq B$ and $\boxed{f(x) \geq (1 - e^{-\gamma}) \cdot OPT}$.

the best known polynomial-time approximation ratio, previously obtained by the greedy algorithm [Das & Kempe, ICML'11]

POSS can do better than the greedy algorithm in cases

[Das & Kempe, STOC'08]

Theorem 2. For the Exponential Decay subclass of sparse regression, POSS using $E[T] = O(B^2(n-B)n\log n)$ finds an optimal solution, while the greedy algorithm cannot.

Sparse regression

Sparse regression [Tropp, TIT'04] : find a sparse approximation solution to the linear regression problem

Formally stated: given all observation variables $V = \{v_1, \ldots, v_n\}$, a predictor variable z and a budget B, it is to find a subset $X \subseteq V$ such that

$$max_{X \subseteq V} \quad R^2_{z,X} = \frac{\mathrm{Var}(z) - \mathrm{MSE}_{z,X}}{\mathrm{Var}(z)} \quad s.t. \quad |X| \leq B.$$

	Corr.	Dis.	LR	···	···	AIC.	BIC	RF.
x1	0.28	0.46	1	···	···	0.22	0.63	1
x2	0.31	0.59	0.64	···	···	0.58	0.56	1
x3	0.11	0.02	0.53	···	···	0.43	0.01	1
x4	0.1	0.1	0.64	···	···	0.73	0.92	1
x5	0.02	0.15	0.33	···	···	0.56	0.36	0.78
x6	0.36	0.02	0.01	···	···	0.32	0.02	0.22
x7	0.2	0.2	0.21	···	···	0.21	0.02	0.11
x8	0.1	0.03	0.32	···	···	0.33	0.51	0.44
x9	0.32	0.1	0.2	···	···	0.06	0.66	0
x10	0.24	0	0.02	···	···	0.6	0.03	0.33
x11	0.12	0.45	0.44	···	···	0.64	0.45	1
x12	0.36	0.58	0.12	···	···	0.73	0.58	0.67
x13	0.2	0.02	0.24	···	···	0.34	0.02	0.89
x14	0.24	0.92	0.33	···	···	0.24	0.93	0.56

$\longrightarrow$

	Corr.	Dis.	LR	···	···	AIC.	BIC	RF.
x1	0.28	0.46	1	···	···	0.22	0.63	1
x2	0.31	0.59	0.64	···	···	0.58	0.56	1
x3	0.11	0.02	0.53	···	···	0.43	0.01	1
x4	0.1	0.1	0.64	···	···	0.73	0.92	1
x5	0.02	0.15	0.33	···	···	0.56	0.36	0.78
x6	0.36	0.02	0.01	···	···	0.32	0.02	0.22
x7	0.2	0.2	0.21	···	···	0.21	0.02	0.11
x8	0.1	0.03	0.32	···	···	0.33	0.51	0.44
x9	0.32	0.1	0.2	···	···	0.06	0.66	0
x10	0.24	0	0.02	···	···	0.6	0.03	0.33
x11	0.12	0.45	0.44	···	···	0.64	0.45	1
x12	0.36	0.58	0.12	···	···	0.73	0.58	0.67
x13	0.2	0.02	0.24	···	···	0.34	0.02	0.89
x14	0.24	0.92	0.33	···	···	0.24	0.93	0.56

Experimental results - R^2 values

the size constraint: $B = 8$

the number of iterations of POSS: $2eB^2n$

exhaustive search

greedy algorithms

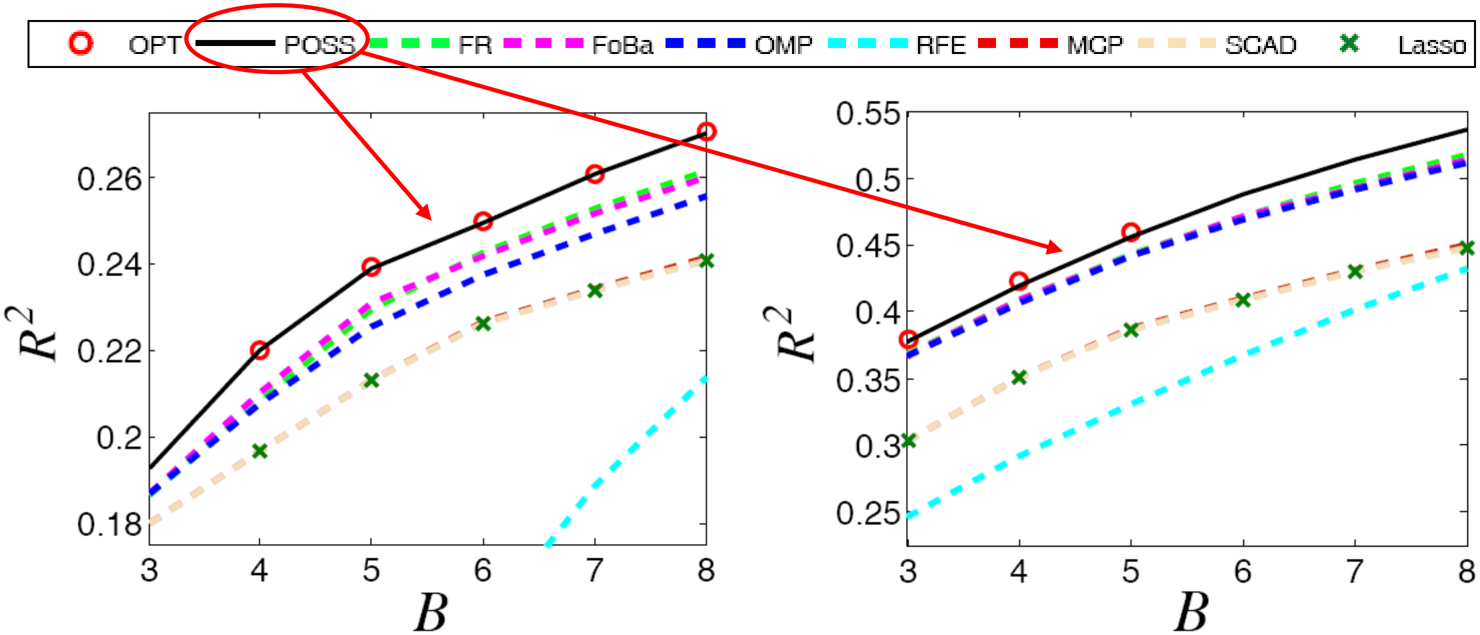
relaxation methods

Data set	OPT	POSS	FR	FoBa	OMP	RFE	MCP
housing	.7437±.0297	.7437±.0297	.7429±.0300•	.7423±.0301•	.7415±.0300•	.7388±.0304•	.7354±.0297•
eunite2001	.8484±.0132	.8482±.0132	.8348±.0143•	.8442±.0144•	.8349±.0150•	.8424±.0153•	.8320±.0150•
svmguide3	.2705±.0255	.2701±.0257	.2615±.0260•	.2601±.0279•	.2557±.0270•	.2136±.0325•	.2397±.0237•
ionosphere	.5995±.0326	.5990±.0329	.5920±.0352•	.5929±.0346•	.5921±.0353•	.5832±.0415•	.5740±.0348•
sonar	–	.5365±.0410	.5171±.0440•	.5138±.0432•	.5112±.0425•	.4321±.0636•	.4496±.0482•
triazines	–	.4301±.0603	.4150±.0592•	.4107±.0600•	.4073±.0591•	.3615±.0712•	.3793±.0584•
coil2000	–	.0627±.0076	.0624±.0076•	.0619±.0075•	.0619±.0075•	.0363±.0141•	.0570±.0075•
mushrooms	–	.9912±.0020	.9909±.0021•	.9909±.0022•	.9909±.0022•	.6813±.1294•	.8652±.0474•
clean1	–	.4368±.0300	.4169±.0299•	.4145±.0309•	.4132±.0315•	.1596±.0562•	.3563±.0364•
w5a	–	.3376±.0267	.3319±.0247•	.3341±.0258•	.3313±.0246•	.3342±.0276•	.2694±.0385•
gisette	–	.7265±.0098	.7001±.0116•	.6747±.0145•	.6731±.0134•	.5360±.0318•	.5709±.0123•
farm-ads	–	.4217±.0100	.4196±.0101•	.4170±.0113•	.4170±.0113•	–	.3771±.0110•
POSS: win/tie/loss		–	12/0/0	12/0/0	12/0/0	11/0/0	12/0/0

POSS is significantly better than all the compared methods on all data sets

Experimental results - R^2 values

different size constraints: $B = 3 \rightarrow 8$

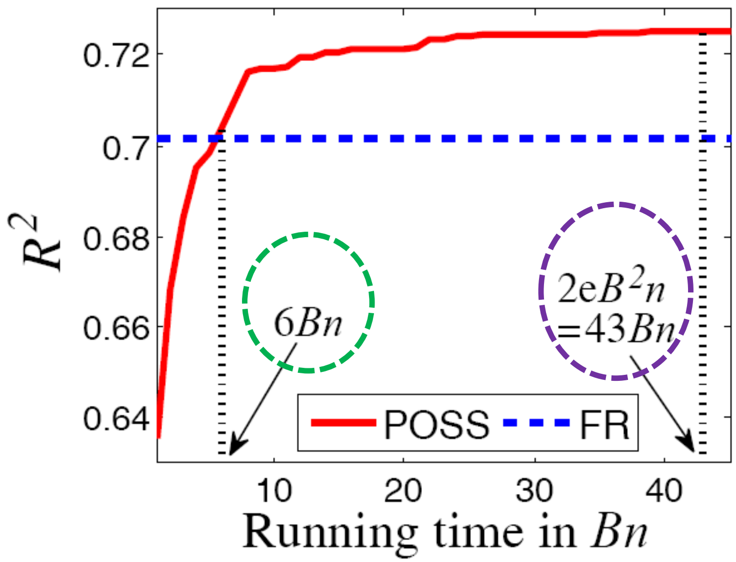


(a) on *svmguide3* (b) on *sonar*

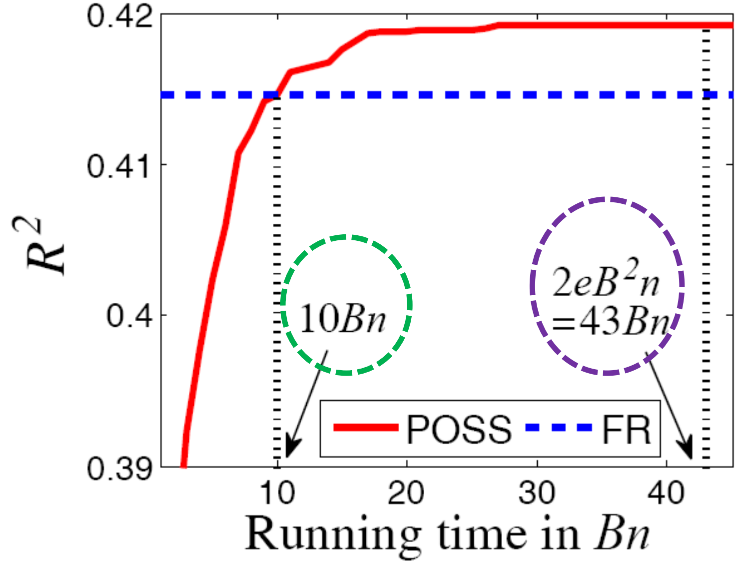
POSS tightly follows OPT, and has a clear advantage over the rest methods

Experimental results – running time

OPT: n^B/B^B greedy methods (FR): Bn POSS: $2eB^2n$



(a) on *gisette*



(b) on *farm-ads*

theoretical running time

POSS can be much more efficient in practice than in theoretical analysis

Monotone set function maximization with general constraints

The POMC approach [Qian, Shi, Yu and Tang, IJCAI'17]

Transformation:

$$\max_{x \in \{0,1\}^n} \; f(x) \quad s.t. \quad c(x) \le B \qquad \text{original}$$

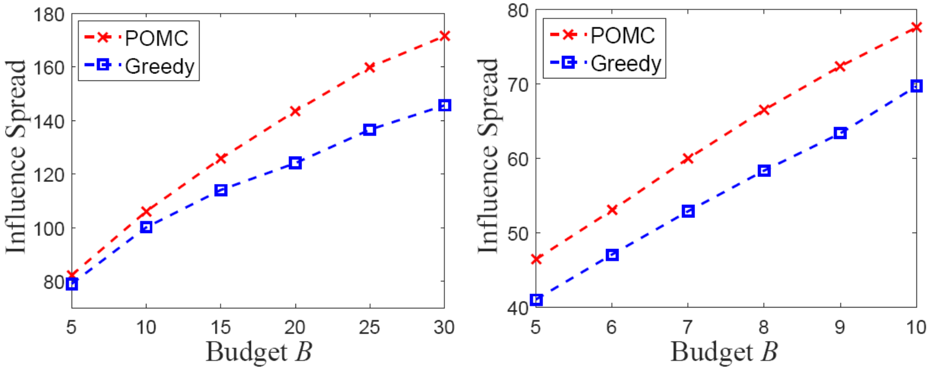
$$\Downarrow$$

$$\min_{x \in \{0,1\}^n} \; (-f(x), c(x)) \qquad \text{bi-objective}$$

Theory: POMC can achieve the same approximation guarantee $(\gamma/2)(1-e^{-\gamma})$ as the greedy algorithm [Zhang & Vorobeychik, AAAI'16]

Application:

influence maximization



(a) (Digg, cardinality) (b) (Synthetic, routing)

Monotone multiset function maximization with size constraints

The POMS approach [Qian, Zhang, Tang and Yao, AAAI'18]

Transformation:

$$\max_{x \in \mathbb{Z}_+^n} \; f(x) \quad s.t. \quad |x| \leq B \qquad \text{original}$$

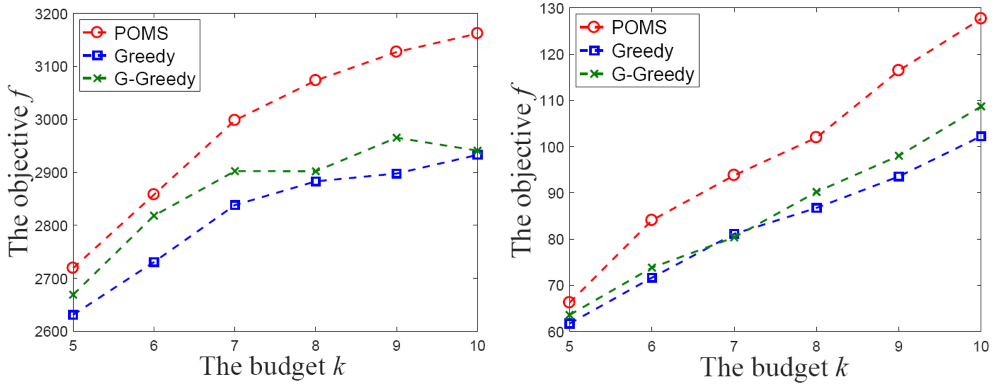
$$\Downarrow$$

$$\min_{x \in \mathbb{Z}_+^n} \; (-f(x), |x|) \qquad \text{bi-objective}$$

Theory: POMS can achieve the same approximation guarantee $(1-1/e)/2$ as the greedy algorithm [Soma et al., ICML'14]

Application:

generalized influence maximization



(a) *ego-Facebook* (b) *Weibo*

Monotone k-submodular function maximization with size constraints

The MOMS approach [Qian, Shi, Tang and Zhou, TEvC in press]

$$max_{x\in\{0,1,\ldots,k\}^n} \; f(x) \;\; s.t. \; |x| \le B$$ original

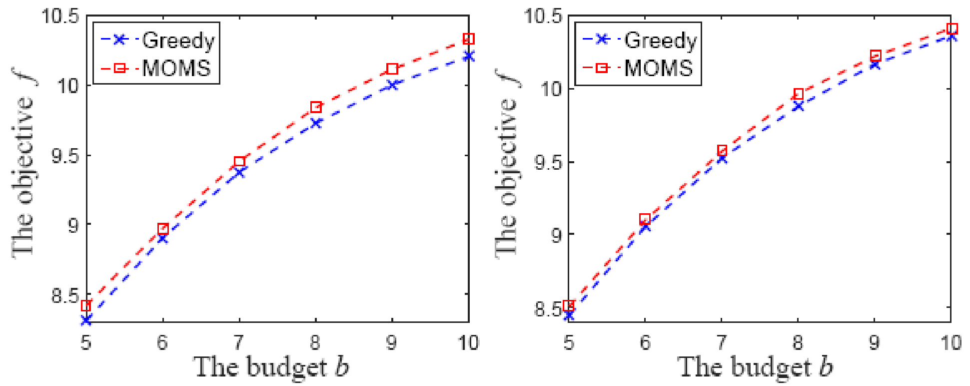
Transformation: ⇩

$$min_{x\in\{0,1,\ldots,k\}^n} \; (-f(x), |x|)$$ bi-objective

Theory: MOMS can achieve the same approximation guarantee 1/2 as the greedy algorithm [Ohsaka & Yoshida, NIPS'15]

Application:

sensor placement



(c) $k = 3$ (d) $k = 4$

Monotone sequence function maximization with size constraints

The POSeqSel approach [Qian, Feng and Tang, IJCAI'18]

Transformation:

$$max_{x \in \mathcal{S}} \ f(x) \quad s.t. \quad |x| \leq B$$ original

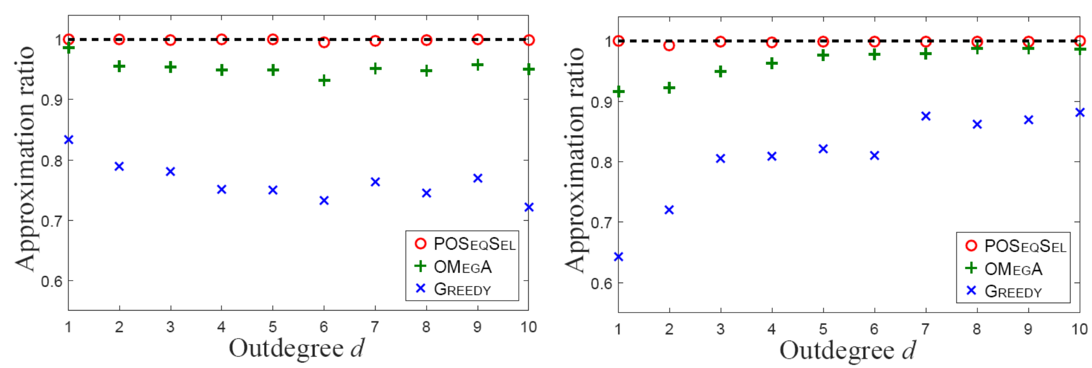
$$\Downarrow$$

$$min_{x \in \mathcal{S}} \ (-f(x), |x|)$$ bi-objective

Theory: POSeqSel can achieve the approximation guarantee $1 - e^{-1/2}$ better than the greedy algorithm [Tschiatschek et al., AAAI'17]

Application:

movie recommendation



(a) modular h (b) submodular h

Ratio optimization of monotone functions

The PORM approach [Qian, Shi, Yu, Tang and Zhou, IJCAI'17]

Transformation:

$$min_{x\in\{0,1\}^n}\ f(x)/g(x) \qquad \text{original}$$

$$\Downarrow$$

$$min_{x\in\{0,1\}^n}\ (f(x),-g(x)) \qquad \text{bi-objective}$$

Theory: PORM can achieve the same approximation guarantee $\frac{|X^*|}{(1+(|X^*|-1)(1-\kappa))\gamma}$ as the greedy algorithm [Bai et al., ICML'16]

Application:

F-measure maximization in information retrieval

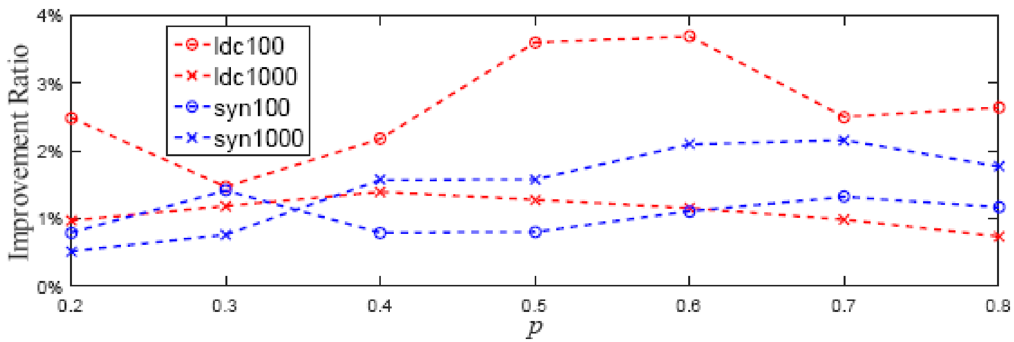


Figure 1: Ratio of improvement of PORM to GreedRatio.

Pareto optimization for subset selection

achieve superior performance on diverse variants of subset selection both theoretically and empirically

The running time (e.g., $2eB^2n$) for achieving a good solution

unsatisfactory when the problem size (e.g., B and n) is large

A sequential algorithm that cannot be readily parallelized

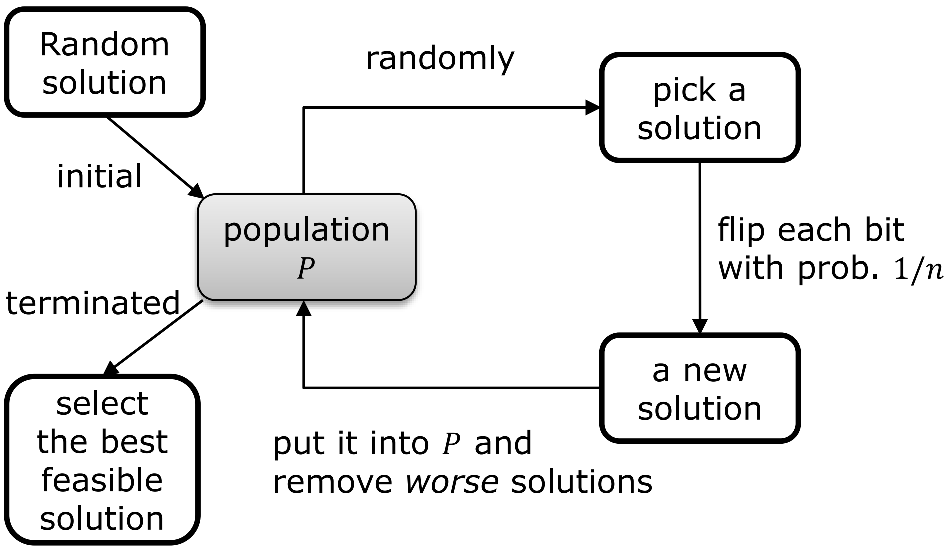
restrict the application to large-scale real-world problems

Can we make the Pareto optimization method parallelizable?

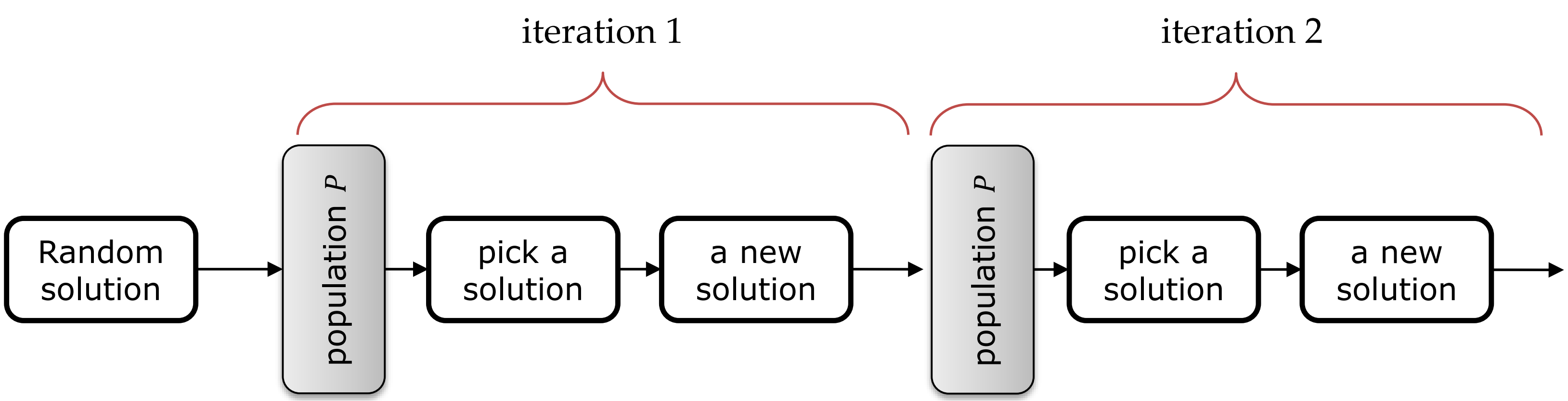
Outline

- Introduction
- Pareto optimization for subset selection
- **Pareto optimization for large-scale subset selection**
- Pareto optimization for noisy subset selection
- Conclusion

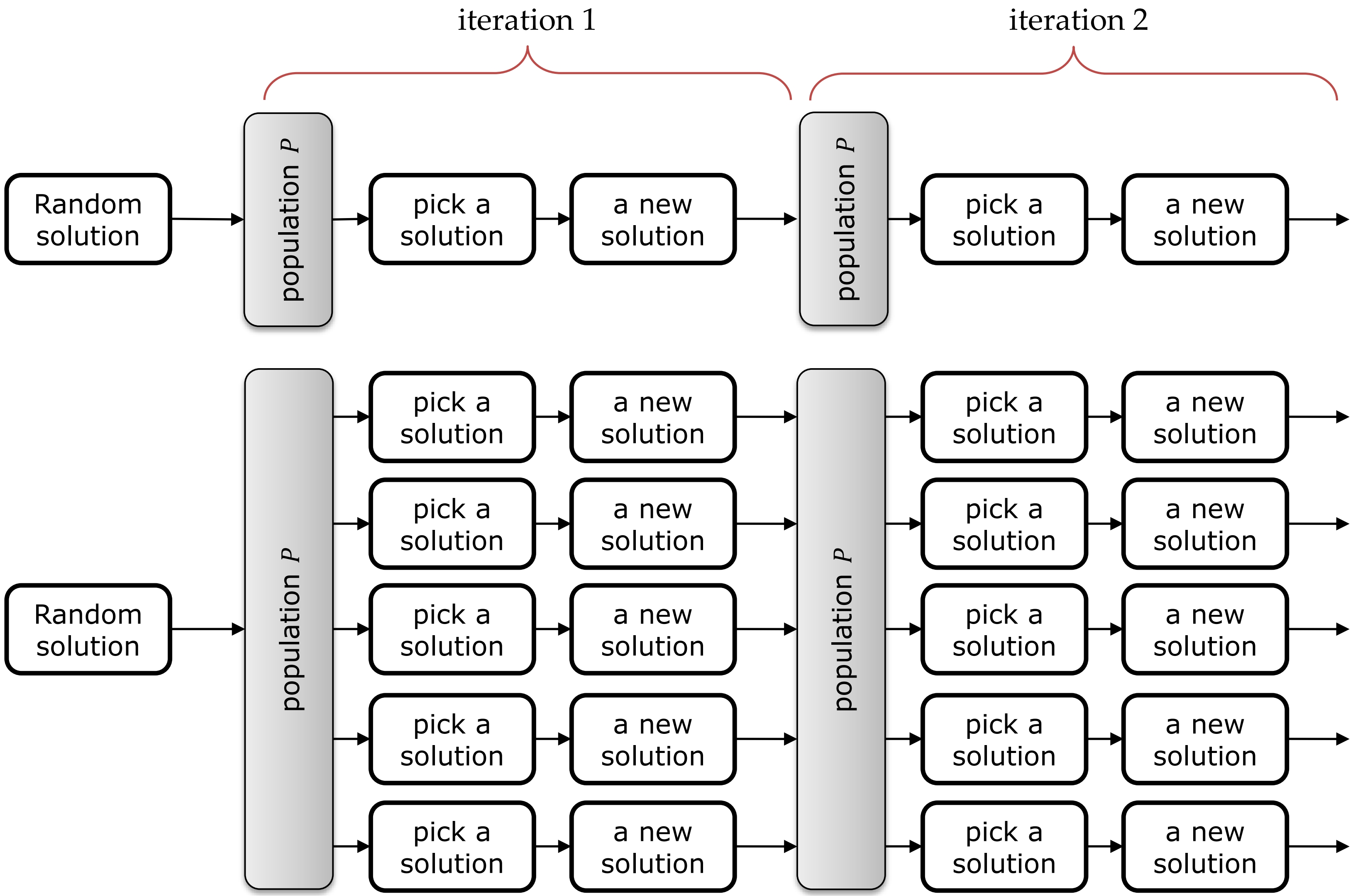
Pareto optimization



1. randomly generate a solution,
   and put it into the population P;
2. loop
   | 2.1 pick a solution randomly from P;
   | 2.2 randomly change it to make a new one;
   | 2.3 if the new one is not *strictly worse*
   | | 2.3.1 put it into P;
   | | 2.3.2 remove *worse* solutions from P;
3. when terminates, select the best feasible
   solution from P.



Parallel Pareto optimization



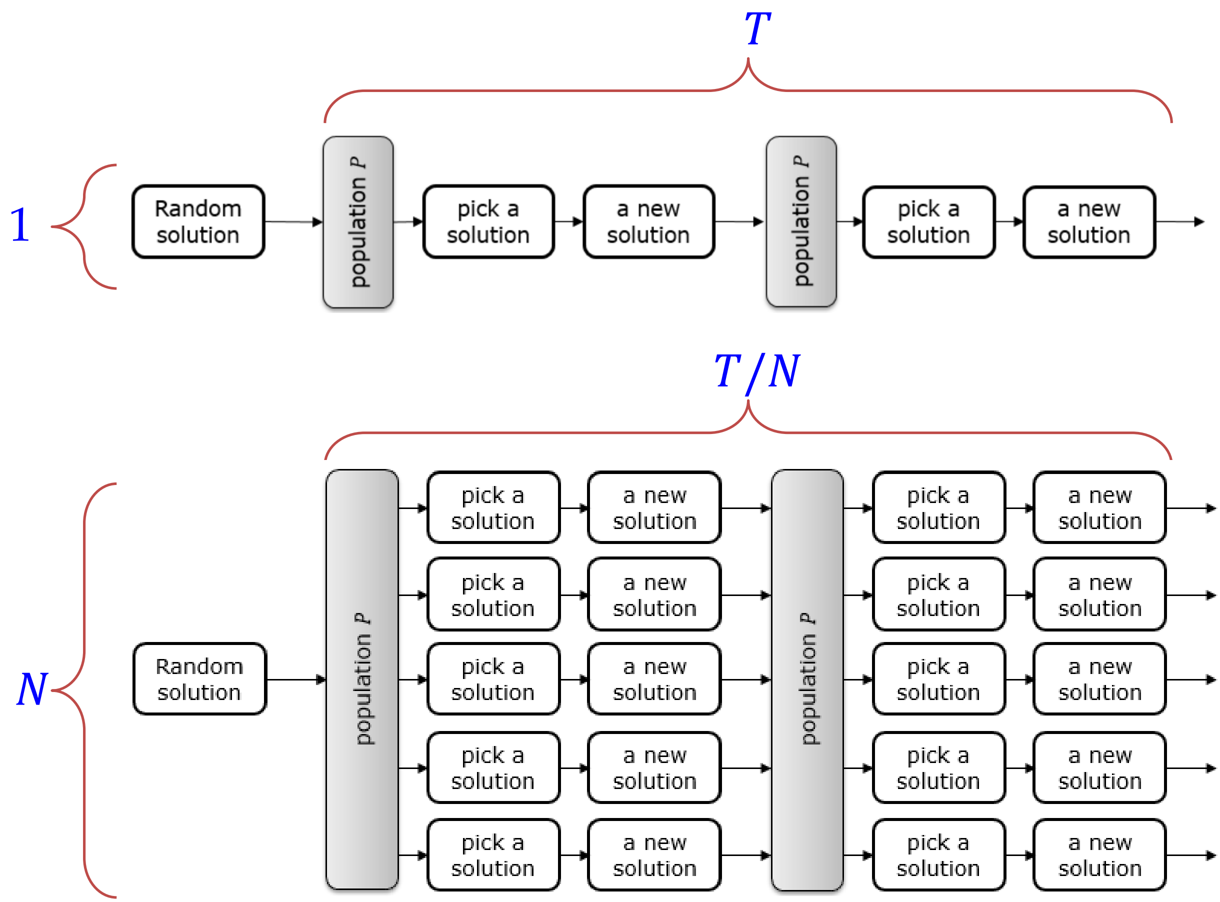
Parallel Pareto optimization

POSS

[Qian et al., NIPS'15]

PPOSS

[Qian et al., IJCAI'16]



Q: the same solution quality? $\frac{1}{en}$ $\longrightarrow$ $1-\left(1-\frac{1}{en}\right)^N \approx \frac{N}{en}$

Theoretical analysis

Theorem 1. For monotone set function maximization with size constraints, the expected number of iterations until PPOSS finds a solution x with $|x| \leq B$ and $\boxed{f(x) \geq (1 - e^{-\gamma}) \cdot OPT}$ is

(1) if $N = o(n)$, then $E[T] \leq 2eB^2n/N$;

(2) if $N = \Omega(n^i)$ for $1 \leq i \leq B$, then $E[T] = O(B^2/i)$;

(3) if $N = \Omega(n^{\min\{3B-1,n\}})$, then $E[T] = O(1)$.

the same approximation bound

- When the number N of processors is less than the number n of items, the number T of iterations can be reduced linearly w.r.t. the number of processors

Theoretical analysis

Theorem 1. For monotone function maximization with cardinality constraints, the expected number of iterations until PPOSS finds a solution x with $|x| \leq B$ and $\boxed{f(x) \geq (1 - e^{-\gamma}) \cdot OPT}$ is

(1) if $N = o(n)$, then $E[T] \leq 2eB^2 n/N$;

(2) if $N = \Omega(n^i)$ for $1 \leq i \leq B$, then $E[T] = O(B^2/i)$;

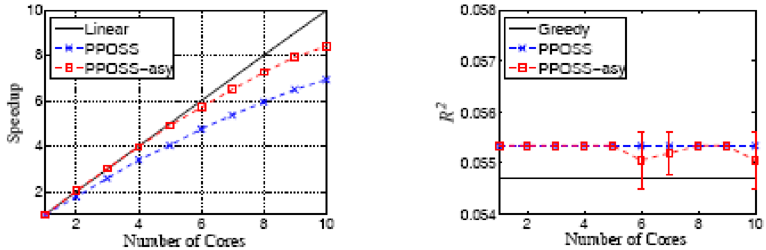
(3) if $N = \Omega(n^{\min\{3B-1,n\}})$, then $E[T] = O(1)$.

the same approximation bound

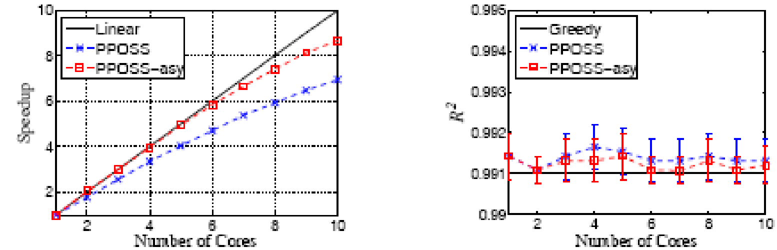
- When the number N of processors is less than the number n of items, the number T of iterations can be reduced linearly w.r.t. the number of processors
- With increasing number N of processors, the number T of iterations can be continuously reduced, eventually to a constant

Experiments on sparse regression

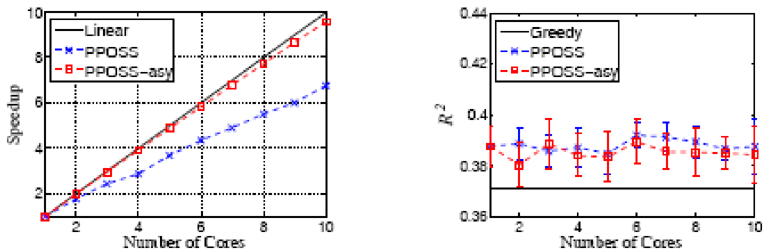
Compare the speedup as well as the solution quality measured by R^2 values with different number of cores



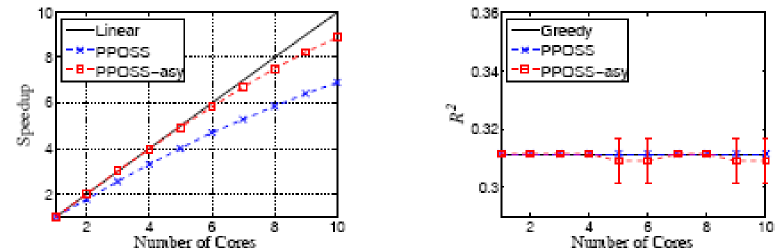
(a) on *coil2000* (9000 instances, 86 features)



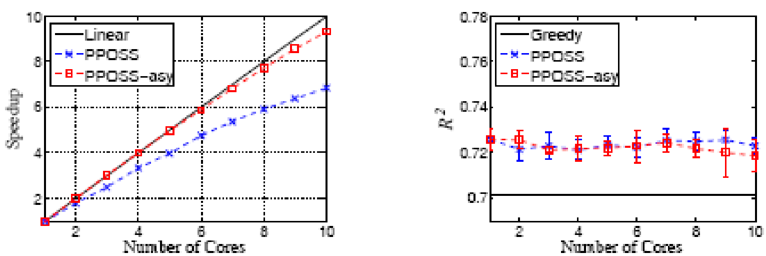
(b) on *mushrooms* (8124 instances, 112 features)



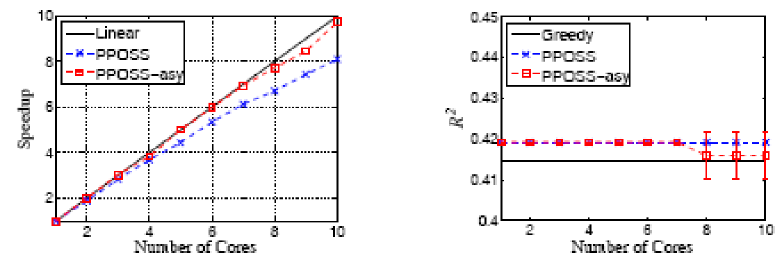
(c) on *clean1* (476 instances, 166 features)



(d) on *w5a* (9888 instances, 300 features)

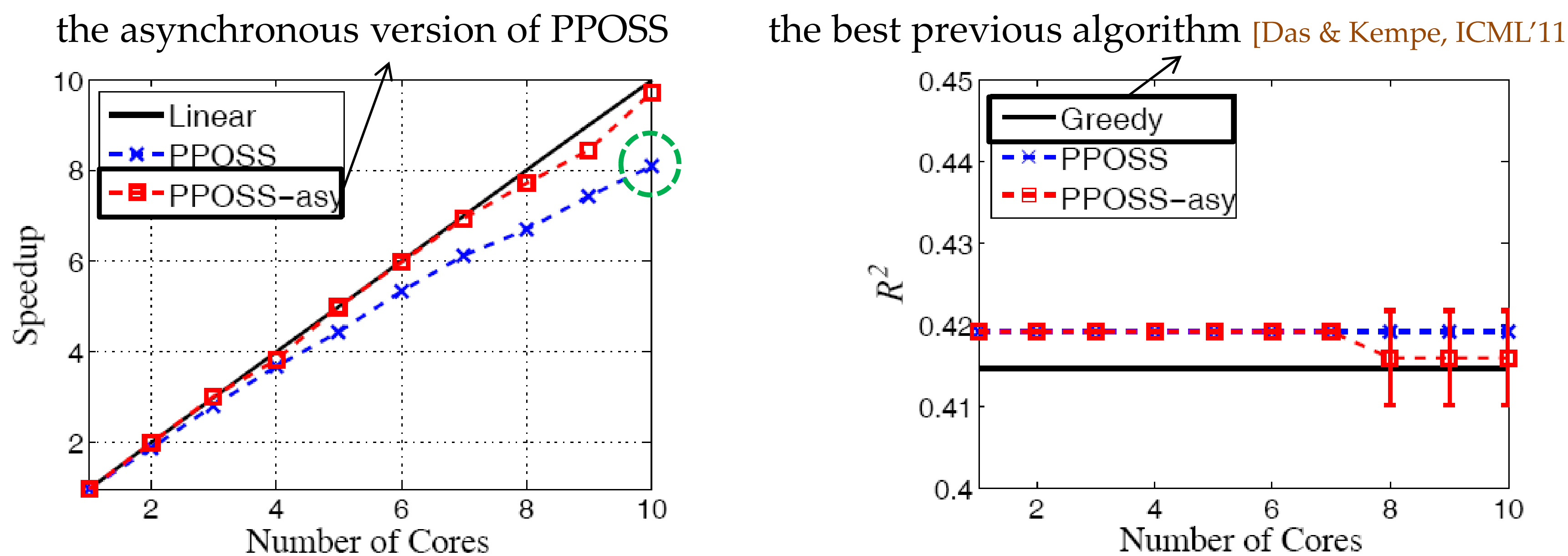


(e) on *gisette* (7000 instances, 5000 features)



(f) on *farm-ads* (4143 instances, 54877 features)

Experiments on sparse regression



(f) on *farm-ads* (4143 instances, 54877 features)

PPOSS (blue line): achieve speedup around 8 when the number of cores is 10; the R^2 values are stable, and better than Greedy

PPOSS-asy (red line): achieve better speedup (avoid the synchronous cost)**;** **the R^2 values are slightly worse** (the noise from asynchronization)

Pareto optimization for subset selection

achieve superior performance on diverse variants of subset selection both theoretically and empirically

Parallel Pareto optimization for subset selection

achieve nearly linear runtime speedup while keeping the solution quality

Require centralized access to the whole data set

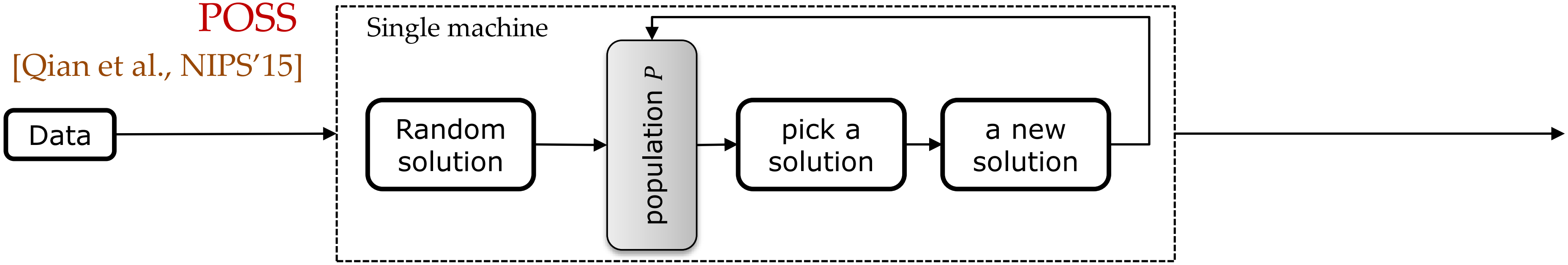
restrict the application to large-scale real-world problems

Can we make the Pareto optimization method distributable?

Distributed Pareto optimization

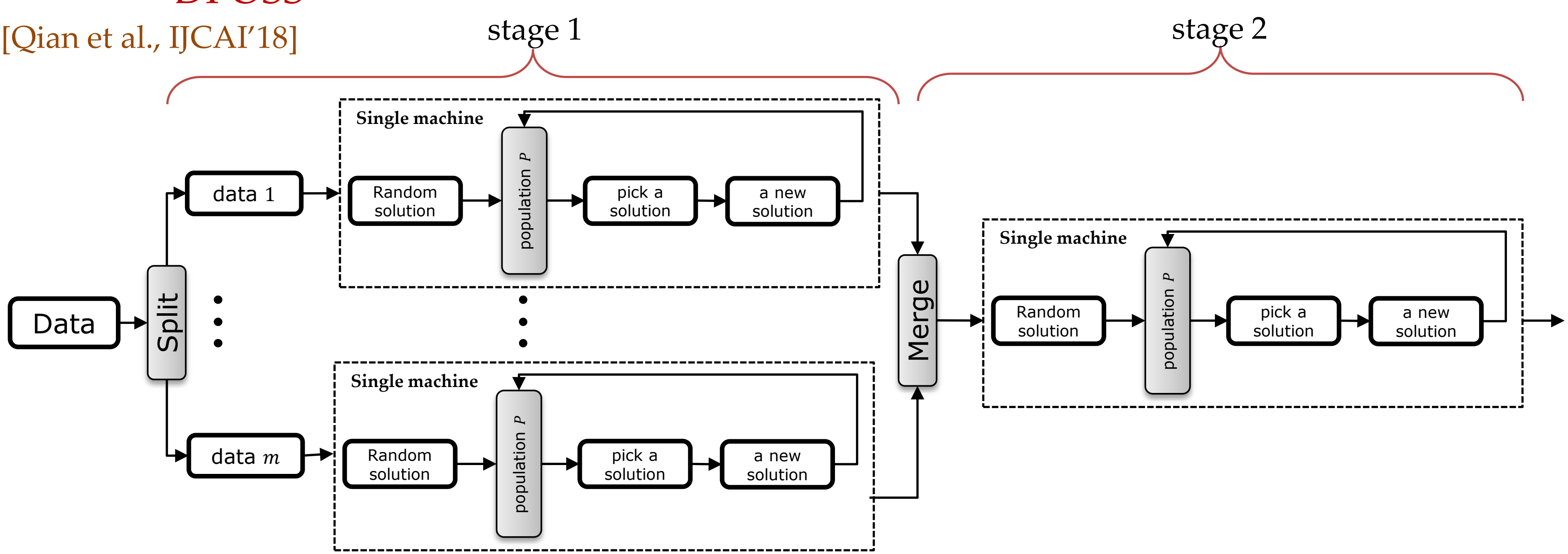
POSS

[Qian et al., NIPS'15]



DPOSS

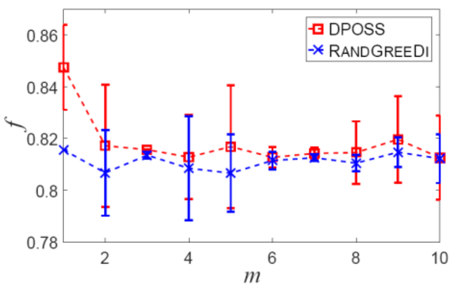
[Qian et al., IJCAI'18]



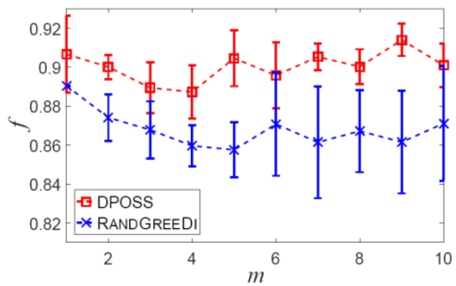
Experiments on sparse regression

Compare DPOSS with the state-of-the-art distributed greedy algorithm RandGreeDi [Mirzasoleiman et al., JMLR'16] **under different number of machines**

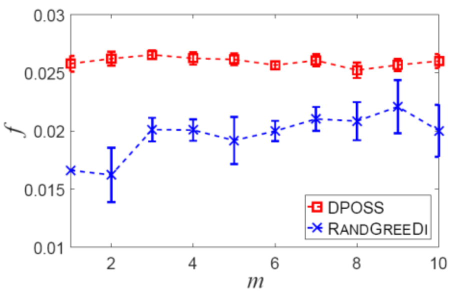
On regular-scale data sets



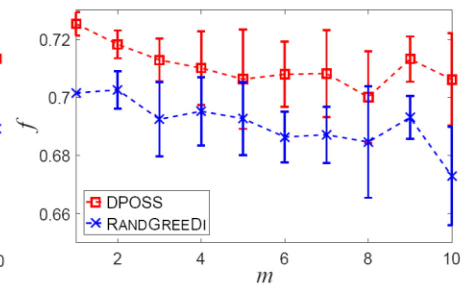
(a) *MicroMass* ($n=1,300$)



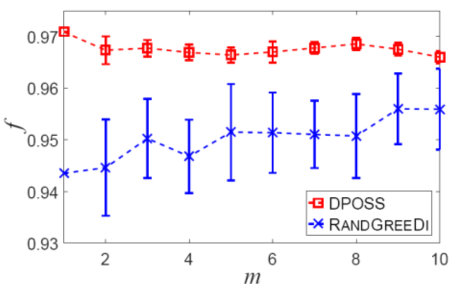
(b) *colon-cancer* ($n=2,000$)



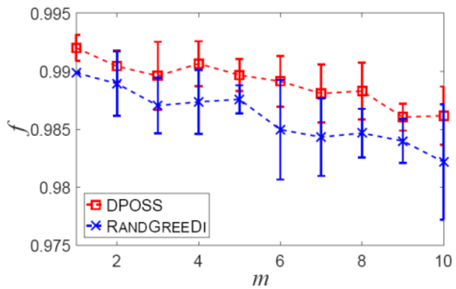
(c) *SVHN* ($n=3,072$)



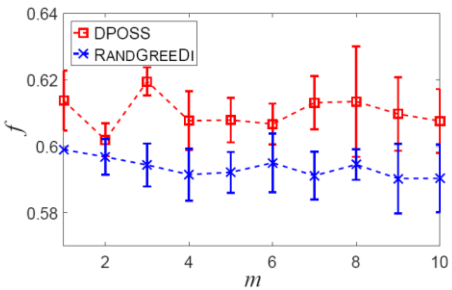
(d) *gisette* ($n=5,000$)



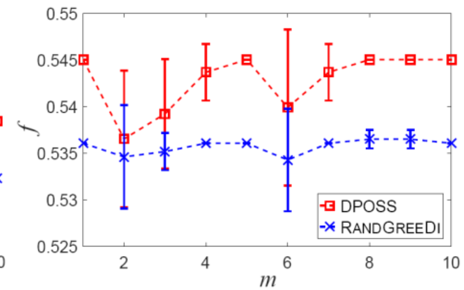
(e) *GHG-Network* ($n=5,232$)



(f) *leukemia* ($n=7,129$)



(g) *Arcene* ($n=10,000$)

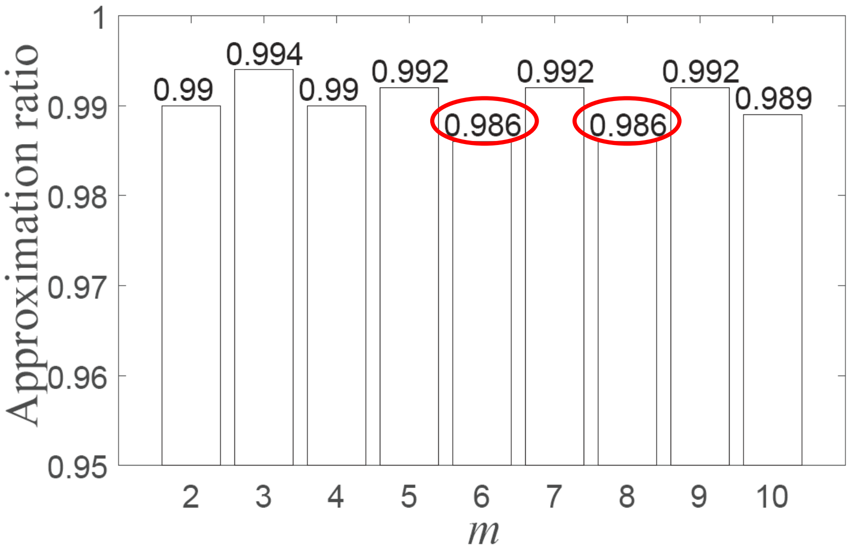


(h) *Dexter* ($n=20,000$)

DPOSS is always better than RandGreeDi

Experiments on sparse regression

On regular-scale data sets



DPOSS is very close to the centralized POSS

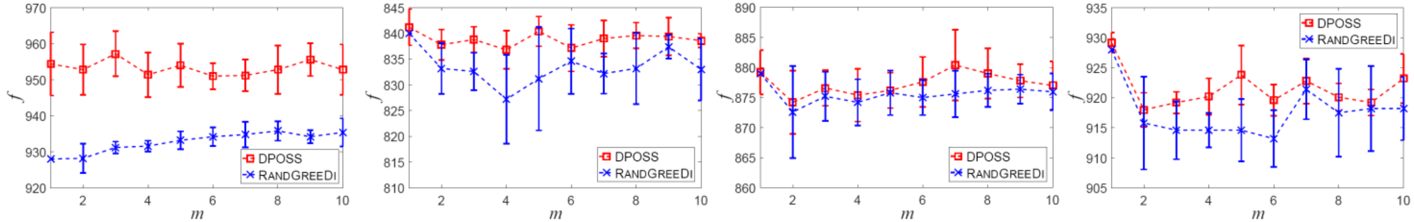
On large-scale data sets

Data set	DPOSS	RandGreeDi
Gas-sensor-flow ($n=120,432$)	$.818 \pm .005$	$.710 \pm .017$
Twin-gas-sensor ($n=480,000$)	$.601 \pm .014$	$.470 \pm .025$
Gas-sensor-sample ($n=1,950,000$)	$.289 \pm .029$	$.245 \pm .018$

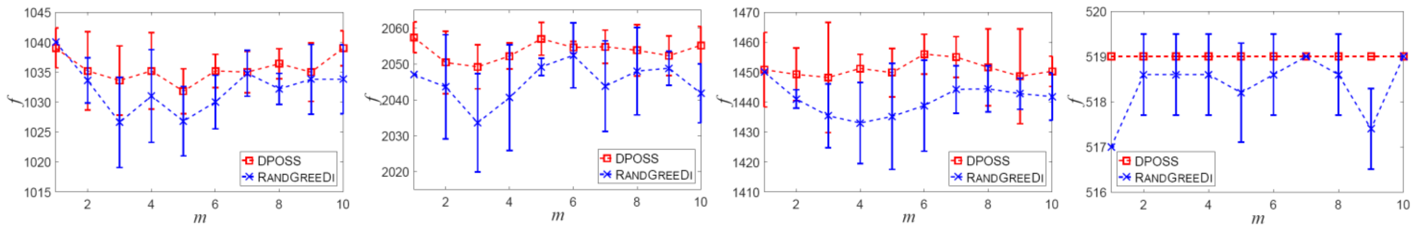
DPOSS is better than RandGreeDi

Experiments on maximum coverage

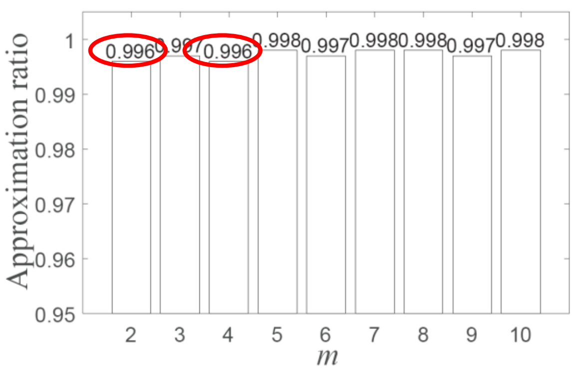
On regular-scale data sets



(a) *hamming* ($n = 1,024$) (b) *frb50-23-mis* ($n = 1,150$) (c) *frb53-24-mis* ($n = 1,272$) (d) *frb56-25-mis* ($n = 1,400$)



(e) *frb59-26-mis* ($n = 1,534$) (f) *frb100-40-mis* ($n = 4,000$) (g) *ca-HepPh* ($n = 12,008$) (h) *retail* ($n = 88,162$)



On large-scale data sets

Data set	DPOSS	RandGreeDi
accident ($n = 340,183$)	175 ± 1	170.6 ± 1.34
kosarak ($n = 990,002$)	9263 ± 0	9263 ± 0

Pareto optimization for subset selection

achieve superior performance on diverse variants of subset selection both theoretically and empirically

Parallel Pareto optimization for subset selection

achieve nearly linear runtime speedup while keeping the solution quality

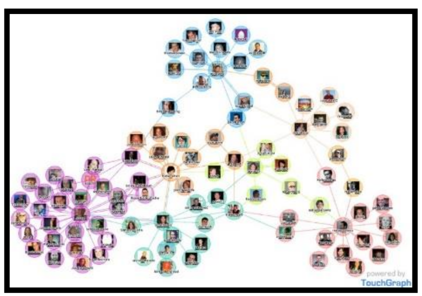
Distributed Pareto optimization for subset selection

achieve very close performance to the centralized algorithm

Noise

Previous analyses often assume that the exact value of the objective function can be accessed

However, in many applications of subset selection, only a noisy value of the objective function can be obtained



Influence maximization

The objective function: the expected number of nodes activated by propagating from X

noise

The average number of active nodes of independent diffusion processes

[Kempe et al., KDD'03]

Noise

Previous analyses often assume that the exact value of the objective function can be accessed

However, in many applications of subset selection, only a noisy value of the objective function can be obtained

	Corr.	Dis.	LR	…	…	AIC.	BIC	RF.
x1	0.28	0.46	1	…	…	0.22	0.63	1
x2	0.31	0.59	0.64	…	…	0.58	0.56	1
x3	0.11	0.02	0.53	…	…	0.43	0.01	1
x4	0.1	0.1	0.64	…	…	0.73	0.92	1
x5	0.02	0.15	0.33	…	…	0.56	0.36	0.78
x6	0.36	0.02	0.01	…	…	0.32	0.02	0.22
x7	0.2	0.2	0.21	…	…	0.21	0.02	0.11
x8	0.1	0.03	0.32	…	…	0.33	0.51	0.44
x9	0.32	0.1	0.2	…	…	0.06	0.66	0
x10	0.24	0	0.02	…	…	0.6	0.03	0.33
x11	0.12	0.45	0.44	…	…	0.64	0.45	1
x12	0.36	0.58	0.12	…	…	0.73	0.58	0.67
x13	0.2	0.02	0.24	…	…	0.34	0.02	0.89
x14	0.24	0.92	0.33	…	…	0.24	0.93	0.56

Sparse regression

	Corr.	Dis.	LR	…	…	AIC.	BIC	RF.
x1	0.28	0.46	1	…	…	0.22	0.63	1
x2	0.31	0.59	0.64	…	…	0.58	0.56	1
x3	0.11	0.02	0.53	…	…	0.43	0.01	1
x4	0.1	0.1	0.64	…	…	0.73	0.92	1
x5	0.02	0.15	0.33	…	…	0.56	0.36	0.78
x6	0.36	0.02	0.01	…	…	0.32	0.02	0.22
x7	0.2	0.2	0.21	…	…	0.21	0.02	0.11
x8	0.1	0.03	0.32	…	…	0.33	0.51	0.44
x9	0.32	0.1	0.2	…	…	0.06	0.66	0
x10	0.24	0	0.02	…	…	0.6	0.03	0.33
x11	0.12	0.45	0.44	…	…	0.64	0.45	1
x12	0.36	0.58	0.12	…	…	0.73	0.58	0.67
x13	0.2	0.02	0.24	…	…	0.34	0.02	0.89
x14	0.24	0.92	0.33	…	…	0.24	0.93	0.56

The objective function: the mean square error of prediction by X

noise

How about the performance for noisy subset selection?

Outline

- Introduction
- Pareto optimization for subset selection
- Pareto optimization for large-scale subset selection
- **Pareto optimization for noisy subset selection**
- Conclusion

Noisy subset selection

Subset selection: given all items $V = \{v_1, \dots, v_n\}$, an objective function $f: 2^V \to R$ and a budget B, it is to find a subset $X \subseteq V$ such that

$$max_{X \subseteq V} \quad f(X) \quad s.t. \quad |X| \leq B.$$

Noise

- Multiplicative: $(1-\epsilon)f(X) \leq F(X) \leq (1+\epsilon)f(X)$
- Additive: $f(X) - \epsilon \leq F(X) \leq f(X) + \epsilon$

Applications: influence maximization, sparse regression

maximizing information gain in graphical models [Chen et al., COLT'15]

crowdsourced image collection summarization [Singla et al., AAAI'16]

Theoretical analysis for greedy algorithms

Multiplicative noise:

$\varepsilon \leq 1/B$ for a constant approximation ratio

$$f(X) \geq \frac{1}{1+\frac{2\epsilon B}{(1-\epsilon)\gamma}}\left(1-\left(\frac{1-\epsilon}{1+\epsilon}\right)^B\left(1-\frac{\gamma}{B}\right)^B\right)\cdot OPT$$

$\gamma \longrightarrow$ submodularity ratio [Das & Kempe, ICML'11]

Additive noise:

$$f(X) \geq \left(1-\left(1-\frac{\gamma}{B}\right)^B\right)\cdot OPT - \left(\frac{2B}{\gamma}-\frac{2B}{\gamma}e^{-\gamma}\right)\epsilon$$

The noiseless approximation guarantee [Das & Kempe, ICML'11]

$$f(X) \geq \left(1-\left(1-\frac{\gamma}{B}\right)^B\right)\cdot OPT \geq (1-e^{-\gamma})\cdot OPT$$

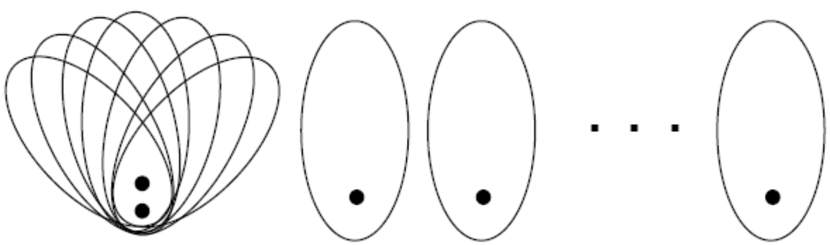
a constant approximation ratio

The performance largely degrades in noisy environments

Theoretical analysis for POSS

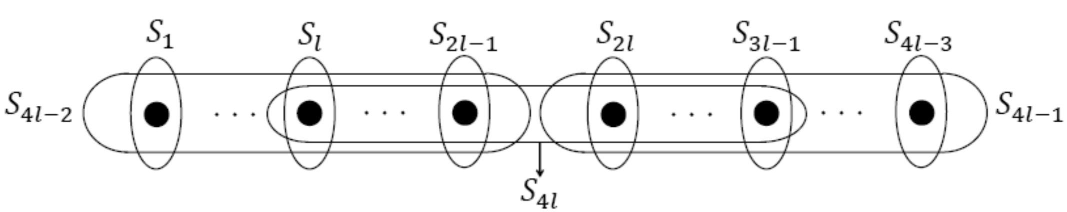
- POSS can generally achieve the same approximation guarantee in both multiplicative and additive noises
- POSS has a better ability of avoiding the misleading search direction led by noise

Maximum coverage



Greedy: very bad approximation
[Hassidim & Singer, COLT'17]

POSS: find the optimal solution through multi-bit search



POSS: find the optimal solution through backward search

PONSS

In our previous work, threshold selection was theoretically shown to be tolerant to noise [Qian et al., ECJ'18]

$$f(x) \geq f(y) \longrightarrow f(x) \geq f(y) + \epsilon$$

Exponentially decrease the running time

POSS

"better"

$$x \preccurlyeq y \Leftrightarrow \begin{cases} f(x) \geq f(y) \\ |x| \leq |y| \end{cases}$$

PONSS [Qian et al., NIPS'17]

Multiplicative:

$$x \preccurlyeq y \Leftrightarrow \begin{cases} f(x) \geq \dfrac{1+\epsilon}{1-\epsilon} f(y) \\ |x| \leq |y| \end{cases}$$

Additive:

$$x \preccurlyeq y \Leftrightarrow \begin{cases} f(x) \geq f(y) + 2\epsilon \\ |x| \leq |y| \end{cases}$$

Theoretical analysis

Multiplicative noise:

$\gamma = 1$ (submodular), ϵ is a constant

PONSS $$f(X) \geq \frac{1-\epsilon}{1+\epsilon}\left(1-\left(1-\frac{\gamma}{B}\right)^B\right)\cdot OPT$$ a constant approximation ratio

$\geq$ significantly better

POSS & Greedy $$f(X) \geq \frac{1}{1+\frac{2\epsilon B}{(1-\epsilon)\gamma}}\left(1-\left(\frac{1-\epsilon}{1+\epsilon}\right)^B\left(1-\frac{\gamma}{B}\right)^B\right)\cdot OPT$$

$\Theta(1/B)$ approximation ratio

Additive noise:

PONSS $$f(X) \geq \left(1-\left(1-\frac{\gamma}{B}\right)^B\right)\cdot OPT - 2\epsilon$$

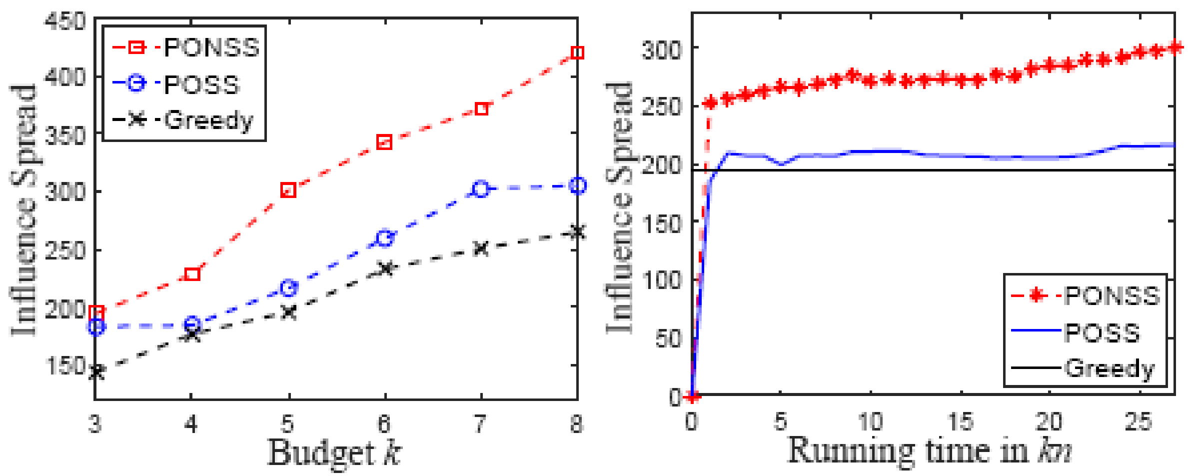
$\geq$ significantly better

POSS & Greedy $$f(X) \geq \left(1-\left(1-\frac{\gamma}{B}\right)^B\right)\cdot OPT - \left(\frac{2B}{\gamma}-\frac{2B}{\gamma}e^{-\gamma}\right)\epsilon$$

Experimental results - influence maximization

PONSS (red line) vs POSS (blue line) vs Greedy (black line):

- Noisy evaluation: the average of 10 independent Monte Carlo simulations
- The output solution: the average of 10,000 independent Monte Carlo simulations

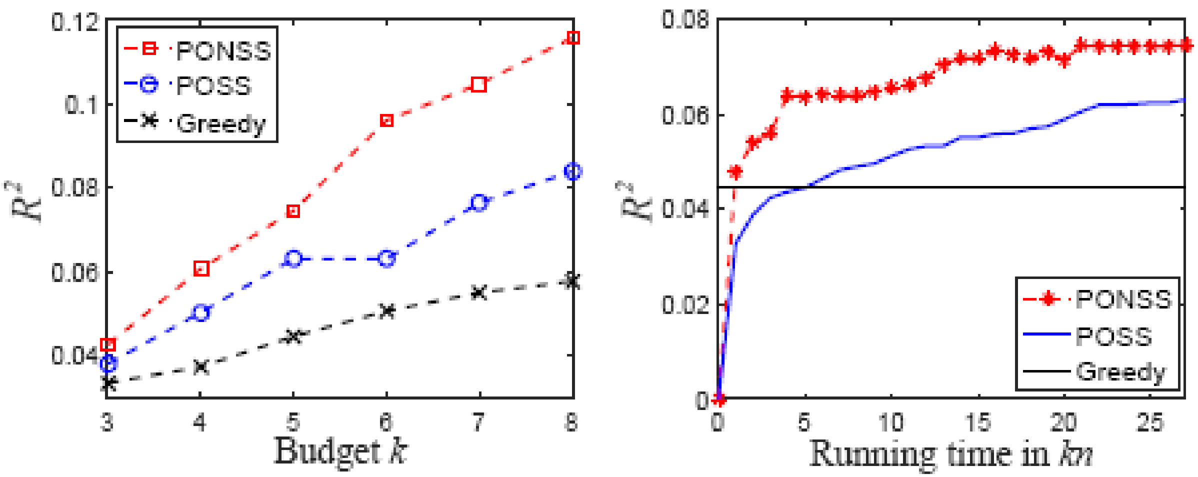


(b) *Weibo* (10,000 #nodes, 162,371 #edges)

Experimental results - sparse regression

PONSS (red line) vs POSS (blue line) vs Greedy (black line):

- Noisy evaluation: a random sample of 1,000 instances
- The output solution: the whole data set



(a) *protein* (24,387 #inst, 357 #feat)

Conclusion

- Pareto optimization for subset selection
- Pareto optimization for large-scale subset selection
- Pareto optimization for noisy subset selection

Future work

- Problem issues
  - Non-monotone objective functions
  - Continuous submodular objective functions
  - Multiple objective functions
- Algorithm issues
  - More complicated MOEAs
- Theory issues
  - Beat the best known approximation guarantee
- Application issues
  - Attempts on more large-scale real-world applications

Collaborators:

Nanjing University:

Jing-Cheng Shi

Yang Yu

Zhi-Hua Zhou

USTC:

Chao Feng

Guiying Li

SUSTech:

Ke Tang

Xin Yao

For details

- C. Qian, Y. Yu, and Z.-H. Zhou. Subset selection by Pareto optimization. In: *Advances in Neural Information Processing Systems 28 (NIPS'15)*, Montreal, Canada, 2015.

- C. Qian, J.-C. Shi, Y. Yu, K. Tang, and Z.-H. Zhou. Parallel Pareto optimization for subset selection. In: *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI'16)*, New York, NY, 2016.

- C. Qian, J.-C. Shi, Y. Yu, and K. Tang. On subset selection with general cost constraints. In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*, Melbourne, Australia, 2017.

- C. Qian, J.-C. Shi, Y. Yu, K. Tang, and Z.-H. Zhou. Optimizing ratio of monotone set functions. In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*, Melbourne, Australia, 2017.

- C. Qian, J.-C. Shi, K. Tang, and Z.-H. Zhou. Constrained monotone *k*-submodular function maximization using multi-objective evolutionary algorithms with theoretical guarantee. *IEEE Transactions on Evolutionary Computation*, in press.

For details

- C. Qian, J.-C. Shi, Y. Yu, K. Tang, and Z.-H. Zhou. Subset selection under noise. In: *Advances in Neural Information Processing Systems 30 (NIPS'17)*, Long Beach, CA, 2017.
- C. Qian, Y. Zhang, K. Tang, and X. Yao. On multiset selection with size constraints. In: *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI'18)*, New Orleans, LA, 2018.
- C. Qian, G. Li, C. Feng, and K. Tang. Distributed Pareto optimization for subset selection. In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*, Stockholm, Sweden, 2018.
- C. Qian, C. Feng, and K. Tang. Sequence selection by Pareto optimization. In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*, Stockholm, Sweden, 2018.
- C. Qian, Y. Yu, and Z.-H. Zhou. Analyzing evolutionary optimization in noisy environments. *Evolutionary Computation*, 2018, 26(1): 1-41.

Codes available at http://staff.ustc.edu.cn/~chaoqian/

THANK YOU !