Novel algorithms for maximum DS decomposition *

Shengminjie Chen[1], Wenguo Yang[1]✉, Suixiang Gao[1], and Rong Jin[2]

[1] School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China
{chenshengminjie19}@mails.ucas.edu.cn,{yangwg,sxgao}@ucas.edu.cn
[2] Department of Computer Science, University of Texas at Dallas, Richardson, TX, 75080, USA {Rong.Jin}@utdallas.edu

Abstract. DS decomposition is an important set function optimization problem. Because DS decomposition is true for any set function, how to solve DS decomposition efficiently and effectively is a heated problem to be solved. In this paper, we focus maximum DS decomposition problem and propose Deterministic Conditioned Greedy algorithm and Random Conditioned algorithm by using the difference with parameter decomposition function and combining non-negative condition. Besides, we get some novel approximation under different parameters. Also, we choose two special case to show our deterministic algorithm get $f\left(S_k\right)-\left(e^{-1}-c_g\right) g\left(S_k\right) \geq\left(1-e^{-1}\right)[f(OPT)-g(OPT)]$ and $f\left(S_k\right)-\left(1-c_g\right) g\left(S_k\right) \geq\left(1-e^{-1}\right) f(OPT)-g(OPT)$ respectively for cardinality constrained problem and our random algorithm get $E[f\left(S_k\right)-\left(e^{-1}-c_g\right) g\left(S_k\right)] \geq\left(1-e^{-1}\right)[f(OPT)-g(OPT)]$ and $E[f\left(S_k\right)-\left(1-c_g\right) g\left(S_k\right)] \geq\left(1-e^{-1}\right) f(OPT)-g(OPT)$ respectively for unconstrained problem, where c_g is the curvature of monotone submodular set function. Because the Conditioned Algorithm is the general framework, different users can choose the parameters that fit their problem to get a better approximation.

Keywords: Non-submodularity · DS decomposition · Conditioned Greedy

1 Introduction

Submodular Set Function Optimization, which is a hot issue in discrete optimization, attracts many researchers. In the past few decades, there have been many important results about maximum submodular set function. Nemhauser et al. (1978)[20] proposed a greedy algorithm, which adds a maximal marginal gains element to current solution in each iteration, can get $1-e^{-1}$ approximation for optimal solution under cardinality constrained. Meanwhile, Fisher et al. (1978) [6] proved that the greedy strategy can get $(1+p)^{-1}$ approximation

* This work was supported by the National Natural Science Foundation of China under Grant 11991022.

for optimal solution under intersection of p matroid. Furthermore, Conforti et al. (1984) [19] constructed a metric called curvature. Using curvature, authors proved that the greedy algorithm can get more tight approximation ($\frac{1-e^{-c}}{c}$ and $\frac{1}{1+c}$) for above two constrained. That is why greedy strategy is excellent for monotone submodular maximization. Unfortunately, they assume the set function is monotone submodular function and $f(\emptyset) = 0$. But these groundbreaking works have inspired a great deal of submodular optimization.

Relaxing the restriction of monotony, Feldman et al. (2011) [5] proposed a novel greedy which has a e^{-1} approximation for non-monotone submodular maximization under matroids constrained. Furthermore, Buchbinder et al. (2014) [3] proved that non-monotone submodular maximization under cardinality constrained also has a e^{-1} approximation polynomial algorithm using double greedy. What's more, Buchbinder et.al (2012) [2] also constructed an algorithm which can get 1/2 approximation for non-monotone submodular maximization without constrains.

Based on these efficient algorithms, submodular optimization has played an important role in data mining, machine learning, economics and operation research such as influence maximization (Kempe et al.,2003 [13]), active learning (Golovin et al.,2011 [7]), document summarization (Lin et al.,2011 [16]), image segmentation (Jegelka et al.,2011 [12]). Unfortunately, more and more objective functions are not submodular in practical problems. Thus, how to optimize a general set function is the most important problem and has puzzled many scholars. Some researchers proposed lots of definition about approximation of submodularity. Krause et al. (2008) [14] constructed ϵ-Diminishing returns to evaluate what is the difference of violation marginal gains decreasing. And authors proved the standard greedy algorithm can get a $f(X) \geq \left(1 - e^{-1}\right)(OPT - k\epsilon)$ approximation for size constrained problem. Das and Kempe (2011) [4] used to measure violation about sub-modularity by submodular ration. Besides, they proved the standard greedy strategy can get a $f(X) \geq (1 - e^{-\gamma})\,OPT$ approximation under cardinality constrained problem. Horel and Singer (2016) [10] proposed ϵ-approximation sub-modularity to calculate the difference of submodularity. According this metric, authors proved that the standard greedy algorithm can return a $f(X) \geq \frac{1}{1+\frac{4k\epsilon}{(1-\epsilon)^2}} \left(1 - e^{-1}\left(\frac{1-\epsilon}{1+\epsilon}\right)^{2k}\right) \cdot \mathrm{OPT}$ approximation under size constrained. It is no surprise that computing these metrics is also a hard problem. It makes these results only theoretical meaning, but lack of practical application.

From the application perspective to solve set function optimization, Lu et al. (2015) [17] utilized a submodular upper bound and a submodular lower bound to constrain a set function and solved these three problems respectively. Their algorithm chose the best one of them to return called Sandwich Approach. Because this approach can get a parametric approximation, some researchers also applied Sandwich Approach to solve their problems (Yang et al. (2020) [22], Zhu et al. (2019) [24], Wang et al. (2017) [21]). In addition, Iyer and Bilmes (2012) [11] proved that any set function can decompose the difference of two submodular set functions called DS decomposition. Specially, the two submodular set function are monotone and non-decreasing.

Lemma 1. *(Iyer and Bilmes 2012) Any set function $h : 2^{\Omega} \to R$ can decompose the difference of two monotone non-decreasing submodular set functions f and g, i.e. $h = f - g$.*

According to this lemma, Iyer and Bilmes proposed Sub-Sup, Sup-Sub, Mod-Mod algorithms to solve minimum this problem. This excellent result has been applied in many ways (Han et al. (2018) [8], Maehara et al. (2015) [18] and Yu et al. (2016) [23]). Although DS decomposition has a bright application prospect, about a general set function, there are problems worth studying how to find the decomposition quickly and how to solve DS decomposition efficiently and effectively. These problems needed to be solved also inspire us to think how to solve maximum DS decomposition.

In this paper, we focus the maximum DS decomposition about set functions. Since DS decomposition is the difference between two submodular functions, we proposed Deterministic and Random Conditioned Greedy algorithm by using the difference with parameter decomposition function and combining non-negative condition. There are general frameworks and users can choose rational parameters according the property of problem. The major contribution of our work are as follows:

· Deterministic Conditioned Greedy is a deterministic framework which introduces some parameters about iteration rounds and non-negative for cardinality constrained problem. Each iteration, the algorithm chooses the element of maximal marginal parametric gains from ground set. Under some rational assumption, the Deterministic Conditioned Greedy can get a polynomial approximation for maximum DS decomposition.

·Two special cases. We choose special parameters in Deterministic Conditioned Greedy. And the algorithm can get two novel approximations $f(S_k) - (e^{-1} - c_g)g(S_k) \geq \left(1 - e^{-1}\right)[f(OPT) - g(OPT)]$ and $f(S_k) - (1 - c_g)g(S_k) \geq (1 - e^{-1})f(OPT) - g(OPT)$ respectively for cardinality constrained problem.

·Random Conditioned Greedy is a random framework which introduces some parameters about iteration rounds and non-negative for unconstrained problem. Each iteration, the algorithm chooses an element from ground set uniformly. Under some rational assumption, the Random Conditioned Greedy can get a polynomial approximation for maximum DS decomposition.

·Two special cases. In Random Conditioned Greedy, we choose the same parameters as Deterministic Conditioned Greedy. And the algorithm can get two novel approximations $E\left[f(S_k) - \left(e^{-1} - c_g\right)g(S_k)\right] \geq \left(1 - e^{-1}\right)[f(OPT) - g(OPT)]$ and $E\left[f(S_k) - (1 - c_g)g(S_k)\right] \geq \left(1 - e^{-1}\right)f(OPT) - g(OPT)$ respectively for unconstrained problem.

The rest of this paper is organized as follow. Some related works about greedy strategy, we put them in Section 2. In Section 3, we propose Deterministic Conditioned Greedy Algorithm and prove approximation. We get some special results for Deterministic Conditioned Greedy for two cases in Section 4. Random Conditioned Greedy Algorithm, we introduce in Section 5. Also, Section 6 is special cases for Random Conditioned Greedy. Conclusion and future works are in Section 7.

2 Related Works

In this section, we introduce some related works about set function decomposition and algorithm.

Iyer and Bilmes (2012) [11] first tried to optimize the set function from the perspective of decomposition. They proved that DS decomposition exists for any set functions. What's more, they proposed three greedy strategy to solve the DS decomposition called Sup-Sub, Sub-Sup, Mod-Mod respectively. But in their work, these three greedy strategies are used to solve the minimization problem. As for maximum DS decomposition, this is an urgent problem to be solved. From the decomposition perspective, Li et al (2020) [15] proved a variation of DS decomposition, any set function can be expressed as the difference of two monotone nondecreasing super-modular functions. Similarly, they also proposed greedy strategies like Iyer and Bilmes called Mod-Mod, Sup-Mod.

Bai and Bilmes (2018) [1] found that some set function can be expressed as the sum of a submodular and super-modular function (BP decomposition), both of which are non-negative monotone non-decreasing. But they didn't show what circumstances exists a BP decomposition. Interestingly, they proved that greedy strategy can get a $\frac{1}{k_f}\left[1-e^{-(1-k_g)k_f}\right]$ and $\frac{1-k_g}{(1-k_g)k_f+p}$ approximation under cardinality and matroid constrained, where k_f, k_g are curvature about submodular and super-modular functions.

Harshaw et al. (2019) [9] focused on the difference of a monotone non-negative submodular set function and a monotone non-negative modular set function. Besides, they proposed Distorted Greedy to solve this problem and get a $f(S)-g(S) \geq \left(1-e^{-1}\right)f(OPT)-g(OPT)$ approximation. This greedy strategy inspires our idea to solve maximization of DS decomposition.

3 Deterministic Conditioned Greedy Algorithm

Greedy strategy is the useful algorithm to solve discrete optimization problem, because it is simple and efficient. In the submodular optimization problem, since greedy algorithm can get an excellent constant approximation, many researchers use greedy strategy. Although, some practical problems are not submodular, fortunately, they have DS decomposition (Lemma 1). Therefore, we proposed a deterministic conditioned greedy algorithm to solve the following problem, where the two submodular set function f,g are monotone and non-decreasing.

$$max_{X\subseteq \Omega,|X|\leq k} h(X) = f(X) - g(X)$$

In Algorithm 1, the $A(i)$ and $B(i)$ are chosen by algorithm designers. From practical perspectives, we assume $f(\emptyset) = g(\emptyset) = 0$, $A(i) \geq 0$ and $B(i) \geq 0$. From the proof, we assume $\left(1-\frac{1}{k}\right)A(i+1) - A(i) \geq 0$ and $B(i+1) -$

$B(i) \geq 0$ specially. When designers choose well-defined parameters that are related to iteration round, it can get a wonderful approximation about maximum DS decomposition. Firstly, we introduce two auxiliary functions and curvature which are useful in process of approximation proof.

Algorithm 1: Deterministic Conditioned Greedy
Input: cardinality k, parameters $A(i)$,$B(i)$
1.Initialize $S_0 \leftarrow \emptyset$
2.For $i = 0$ to $k-1$
3. $e_i \leftarrow \arg\max_{e \in \Omega} \{A(i+1) f(e \mid S_i) - B(i+1) g(e \mid \Omega \setminus e)\}$
4. If $A(i+1) f(e_i \mid S_i) - B(i+1) g(e_i \mid \Omega \setminus e_i) > 0$ then
5. $S_{i+1} \leftarrow S_i \cup \{e_i\}$
6. Else
7. $S_{i+1} \leftarrow S_i$
8.End for
9.Return S_k

Definition 1. *Define two auxiliary functions*

$$\phi_i(T) = A(i) f(T) - B(i) \sum_{e \in T} g(e|\Omega \setminus e)$$

$$\psi_i(T, e) = \max\{0, A(i+1) f(e \mid T) - B(i+1) g(e \mid \Omega \setminus e)\}$$

Definition 2. *(Conforti(1984)) Given a monotone submodular set function $f : 2^{\Omega} \rightarrow R$, the curvature of f is*

$$c_f = 1 - min_{e \in \Omega} \frac{f(e|\Omega \setminus e)}{f(e)}$$

Look at the definition of two functions, $\psi_i(T, e)$ is the condition of the Algorithm 1. And $\phi_i(T)$ is the surrogate objective function. Next, we prove an important property about surrogate objective function.

Property 1. In each iteration

$$\begin{aligned} &\phi_{i+1}(S_{i+1}) - \phi_i(S_i) \\ &= \psi_i(S_i, e_i) + [A(i+1) - A(i)] f(S_i) - [B(i+1) - B(i)] \sum_{e \in S_i} g(e|\Omega \setminus e) \end{aligned} \quad (1)$$

Proof.

$$
\begin{aligned}
&\phi_{i+1}\left(S_{i+1}\right)-\phi_i\left(S_i\right)\\
&=A(i+1) f\left(S_{i+1}\right)-B(i+1) \sum_{e \in S_{i+1}} g(e \mid \Omega \setminus e)-A(i) f\left(S_i\right)+B(i) \sum_{e \in S_i} g(e \mid \Omega \setminus e)\\
&=A(i+1)\left[f\left(S_{i+1}-f\left(S_i\right)\right)\right]-B(i+1) g\left(e_i \mid \Omega \setminus e_i\right)\\
&+[A(i+1)-A(i)] f\left(S_i\right)-[B(i+1)-B(i)] \sum_{e \in S_i} g(e \mid \Omega \setminus e)\\
&=\psi_i\left(S_i, e_i\right)+[A(i+1)-A(i)] f\left(S_i\right)-[B(i+1)-B(i)] \sum_{e \in S_i} g(e \mid \Omega \setminus e)
\end{aligned}
\tag{2}
$$

□

Using the Property 1, we construct a relationship from Deterministic Conditioned Greedy to surrogate objective function. Interestingly, the condition of Algorithm 1 has a lower bound. Therefore, in each iteration of Algorithm 1, the marginal gain of surrogate objective function has some guarantees.

Theorem 1. $\psi_i\left(S_i, e_i\right) \geq \frac{1}{k} A(i+1)\left[f(OPT)-f\left(S_i\right)\right]-\frac{1}{k} B(i+1) g(OPT)$

Proof.

$$
\begin{aligned}
k \cdot \psi_i\left(S_i, e_i\right) &= k \cdot \max_{e \in \Omega}\left\{0, A(i+1) f\left(e \mid S_i\right)-B(i+1) g(e \mid \Omega \setminus e)\right\}\\
&\geq |OPT| \cdot \max_{e \in \Omega}\left\{0, A(i+1) f\left(e \mid S_i\right)-B(i+1) g(e \mid \Omega \setminus e)\right\}\\
&\geq |OPT| \cdot \max_{e \in OPT}\left\{A(i+1) f\left(e \mid S_i\right)-B(i+1) g(e \mid \Omega \setminus e)\right\}\\
&\geq \sum_{e \in OPT}\left[A(i+1) f\left(e \mid S_i\right)-B(i+1) g(e \mid \Omega \setminus e)\right]\\
&= A(i+1) \sum_{e \in OPT} f\left(e \mid S_i\right)-B(i+1) \sum_{e \in OPT} g(e \mid \Omega \setminus e)\\
&\geq A(i+1)\left[f\left(OPT \cup S_i\right)-f\left(S_i\right)\right]-B(i+1) g(OPT)\\
&\geq A(i+1)\left[f(OPT)-f\left(S_i\right)\right]-B(i+1) g(OPT)
\end{aligned}
\tag{3}
$$

The first inequality is $k \geq |OPT|$. The second inequality is $OPT \subseteq \Omega$. The third inequality is the maximum. The fourth inequality is sub-modularity, i.e. $f(OPT \cup S_i) - f(S_i) \leq \sum_{e \in OPT} f(e|S_i)$ and $\sum_{e \in OPT} g(e|\Omega \setminus e) \leq g(OPT)$. The last inequality is monotony. □

Combined Property 1 and Theorem 1, the following corollary is obvious.

Corollary 1.

$$
\begin{aligned}
&\phi_{i+1}\left(S_{i+1}\right)-\phi_i\left(S_i\right) \geq \frac{1}{k} A(i+1) f(OPT)-\frac{1}{k} B(i+1) g(OPT)+\\
&\left[\left(1-\frac{1}{k}\right) A(i+1)-A(i)\right] f\left(S_i\right)-[B(i+1)-B(i)] \sum_{e \in S_i} g(e \mid \Omega \setminus e)
\end{aligned}
\tag{4}
$$

According the above assumption and Lemma 1, we have $\sum_{e\in S_i} g(e|\Omega\setminus e)\geq 0$ and $f\left(S_i\right)\ \geq 0$. That is to say that the proper selection of $A\left(i\right)$ and $B\left(i\right)$ is the key to the final performance of the algorithm. In our work, we assume that they should satisfy $\left(1-\frac{1}{k}\right)A\left(i+1\right)-A\left(i\right)\geq 0$ and $B\left(i+1\right)-B\left(i\right)\geq 0$. If $\left[\left(1-\frac{1}{k}\right)A\left(i+1\right)-A\left(i\right)\right]f\left(S_i\right)-\left[B\left(i+1\right)-B\left(i\right)\right]\sum_{e\in S_i} g\left(e|\Omega\setminus e\right)\ \geq 0$, the result is trivial cause we can ignore them. Therefore, we can get an approximation guarantee about maximum DS decomposition under cardinality constrained using Deterministic Conditioned Greedy.

Theorem 2. *S_k is the solution of Algorithm 1 after k iteration*

$$A\left(k\right)f\left(S_k\right)-\left(B\left(0\right)-c_g\right)g\left(S_k\right)\geq\frac{1}{k}\sum_{i=0}^{k-1}\left[A\left(i+1\right)f\left(OPT\right)-B\left(i+1\right)g\left(OPT\right)\right]$$

where c_g is the curvature of function g.

Proof. $\phi_0\left(S_0\right)=A\left(0\right)f\left(\emptyset\right)-B\left(0\right)g\left(\emptyset\right)=0$ According the curvature $c_g=1-min\frac{g(e|\Omega\setminus e)}{g(e)}$, we have $\frac{g(e|\Omega\setminus e)}{g(e)}\geq 1-c_g$,

$$\begin{aligned}&\phi_k\left(S_k\right)=A\left(k\right)f\left(S_k\right)-B\left(k\right)\sum_{e\in S_k} g\left(e\mid\Omega\setminus e\right)\leq A\left(k\right)f\left(S_k\right)-B\left(k\right)\left(1-c_g\right)\sum_{e\in S_k} g\left(e\right)\\&\leq A\left(k\right)f\left(S_k\right)-B\left(k\right)\left(1-c_g\right)g\left(S_k\right)\end{aligned}$$

Thus, we can rewrite an accumulation statement

$$\begin{aligned}&A\left(k\right)f\left(S_k\right)-B\left(k\right)\left(1-c_g\right)g\left(S_k\right)\geq\phi_k\left(S_k\right)-\phi_0\left(S_0\right)=\sum_{i=0}^{k-1}\phi_{i+1}\left(S_{i+1}\right)-\phi_i\left(S_i\right)\\&\geq\sum_{i=0}^{k-1}\left[\frac{1}{k}A\left(i+1\right)f\left(OPT\right)-\frac{1}{k}B\left(i+1\right)g\left(OPT\right)+\left[\left(1-\frac{1}{k}\right)A\left(i+1\right)-A\left(i\right)\right]f\left(S_i\right)\right]\\&-\sum_{i=0}^{k-1}\left[\left[B\left(i+1\right)-B\left(i\right)\right]\sum_{e\in S_i} g\left(e|\Omega\setminus e\right)\right]\\&\geq\frac{1}{k}\sum_{i=0}^{k-1}\left[A\left(i+1\right)f\left(OPT\right)-B\left(i+1\right)g\left(OPT\right)\right]-\left[B\left(k\right)-B\left(0\right)\right]\sum_{e\in S_k} g\left(e|\Omega\setminus e\right)\\&\geq\frac{1}{k}\sum_{i=0}^{k-1}\left[A\left(i+1\right)f\left(OPT\right)-B\left(i+1\right)g\left(OPT\right)\right]-\left[B\left(k\right)-B\left(0\right)\right]g\left(S_k\right)\end{aligned}\tag{5}$$

Therefore, we can conclude

$$A\left(k\right)f\left(S_k\right)-\left(B\left(0\right)-c_g\right)g\left(S_k\right)\ \geq\frac{1}{k}\sum_{i=0}^{k-1}\left[A\left(i+1\right)f\left(OPT\right)-B\left(i+1\right)g\left(OPT\right)\right]$$

□

4 Two Special Cases for Deterministic Conditioned Greedy

In this section, we choose two special case to show Deterministic Conditioned Greedy strategy can get $f(S_k) - \left(e^{-1} - c_g\right) g(S_k) \geq \left(1 - e^{-1}\right)[f(OPT) - g(OPT)]$ approximation and $f(S_k) - (1 - c_g) g(S_k) \geq \left(1 - e^{-1}\right) f(OPT) - g(OPT)$ approximation for cadinality constrained problem.

4.1 Case 1

We set $A(i) = B(i) = \left(1 - \frac{1}{k}\right)^{(k-i)}$. Therefore, the definition 1 and Algorithm 1 become the following. Obviously, these settings satisfy all conditions and assumptions in Section 3. Hence, the following results are clearly. Because the proofs are similar to Section 3, we omit them here.

$$\phi_i(T) = \left(1 - \frac{1}{k}\right)^{k-i} \left[f(T) - \sum_{e \in T} g(e \mid \Omega \setminus e)\right]$$

$$\psi_i(T, e) = \max\left\{0, \left(1 - \frac{1}{k}\right)^{k-(i+1)} [f(e \mid S_i) - g(e \mid \Omega \setminus e)]\right\}$$

Algorithm 2: Deterministic Conditioned Greedy I

Input: cardinality k
1.Initialize $S_0 \leftarrow \emptyset$
2.For $i = 0$ to $k - 1$
3. $e_i \leftarrow \arg\max_{e \in \Omega} \left\{\left(1 - \frac{1}{k}\right)^{k-(i+1)} [f(e \mid S_i) - g(e \mid \Omega \setminus e)]\right\}$
4. If $\left(1 - \frac{1}{k}\right)^{k-(i+1)} [f(e_i \mid S_i) - g(e_i \mid \Omega \setminus e_i)] > 0$ then
5. $S_{i+1} \leftarrow S_i \cup \{e_i\}$
6. Else
7. $S_{i+1} \leftarrow S_i$
8.End for
9.Return S_k

Property 2. In each iteration

$$\phi_{i+1}(S_{i+1}) - \phi_i(S_i) = \psi_i(S_i, e_i) + \frac{1}{k}(1 - \frac{1}{k})^{-1} \phi_i(S_i)$$

Theorem 3. $\psi_i(S_i, e_i) \geq \frac{1}{k}\left(1 - \frac{1}{k}\right)^{k-(i+1)} [f(OPT) - f(S_i) - g(OPT)]$

Theorem 4. $f(S_k) - \left(e^{-1} - c_g\right) g(S_k) \geq \left(1 - e^{-1}\right)[f(OPT) - g(OPT)]$, *where* c_g *is the curvature of function* g.

From Theorem 4, we find an interesting result and get the following corollary, when $c_g = 0$, i.e. the submodular function g is a modular function.

Corollary 2. *If $c_g = 0$, i.e. g is modular. Then we have*

$$f(S_k) - e^{-1} g(S_k) \geq \left(1 - e^{-1}\right)\left[f(OPT) - g(OPT)\right]$$

4.2 Case 2

We set $A(i) = \left(1 - \frac{1}{k}\right)^{(k-i)}$, $B(i) = 1$. Therefore, the definition 1 and Algorithm 1 become the following. Obviously, these settings satisfy all condition in Section 3. And the proofs are similar with Section 3 and Section 4.1. In this subsection, we have omitted all proofs.

$$\phi_i(T) = \left(1 - \frac{1}{k}\right)^{k-i} f(T) - \sum_{e \in T} g(e \mid \Omega \setminus e)$$

$$\psi_i(T, e) = \max\left\{0, \left(1 - \frac{1}{k}\right)^{k-(i+1)} f(e \mid S_i) - g(e \mid \Omega \setminus e)\right\}$$

Algorithm 3: Deterministic Conditioned Greedy II

Input: cardinality k
1.Initialize $S_0 \leftarrow \emptyset$
2.For $i = 0$ to $k - 1$
3. $e_i \leftarrow \arg\max_{e \in \Omega} \left\{\left(1 - \frac{1}{k}\right)^{k-(i+1)} f(e \mid S_i) - g(e \mid \Omega \setminus e)\right\}$
4. If $\left(1 - \frac{1}{k}\right)^{k-(i+1)} f(e_i \mid S_i) - g(e_i \mid \Omega \setminus e_i) > 0$ then
5. $S_{i+1} \leftarrow S_i \cup \{e_i\}$
6. Else
7. $S_{i+1} \leftarrow S_i$
8.End for
9.Return S_k

Property 3. In each iteration

$$\phi_{i+1}(S_{i+1}) - \phi_i(S_i) = \psi_i(S_i, e_i) + \frac{1}{k}\left(1 - \frac{1}{k}\right)^{k-(i+1)} f(S_i)$$

Theorem 5. $\psi_i(S_i, e_i) \geq \frac{1}{k}\left(1 - \frac{1}{k}\right)^{k-(i+1)}\left[f(OPT) - f(S_i)\right] - \frac{1}{k} g(OPT)$

Theorem 6. $f(S_k) - (1 - c_g)\, g(S_k) \geq \left(1 - e^{-1}\right) f(OPT) - g(OPT)$*, where c_g is the curvature of function g.*

Corollary 3. *If $c_g = 0$, i.e. g is modular. Then we have*

$$f(S_k) - g(S_k) \geq \left(1 - e^{-1}\right) f(OPT) - g(OPT)$$

Remark: Clearly, if g is modular, then $h = f - g$ is submodular. The above approximations are different with submodular maximization under cardinality constrained problem $(1 - e^{-1})$. And also, the different parameters can also cause different approximations. We think first gap is caused by non-monotony, because h is not always monotonous. The second gap give us a clue that we can choose the appropriate parameters $A(i)$ and $B(i)$ according to the characteristics of the problem to get a better approximate ratio.

5 Random Conditioned Greedy Algorithm

Random algorithms are important algorithms for discrete optimization problem. Since randomness can bring a lot of uncertain factors and information, in some cases, random algorithms can get better approximation than deterministic algorithms. Therefore, we propose a random conditioned greedy in this section. Interestingly, our random conditioned greedy can also be used in unconstrained problems and cardinality constrained problem. The following statement is the unconstrained problems for DS decomposition.

$$max_{X \subseteq \Omega} h(X) = f(X) - g(X)$$

According Definition 1, the property 1 is also true for random conditioned greedy. We just have to modify the assumptions a little bit $\left(1 - \frac{1}{n}\right) A(i+1) - A(i) \ \geq 0$ to get the following theorems and corollary directly.

Algorithm 4: Random Conditioned Greedy
Input: ground set Ω, parameters $A(i)$,$B(i)$
1.Initialize $S_0 \leftarrow \emptyset$
2.For $i = 0$ to $n - 1$
3. $e_i \leftarrow$be chosen uniformly from Ω
4. If $A(i+1) f(e_i \mid S_i) - B(i+1) g(e_i \mid \Omega \setminus e_i) > 0$ then
5. $S_{i+1} \leftarrow S_i \cup \{e_i\}$
6. Else
7. $S_{i+1} \leftarrow S_i$
8.End for
9.Return S_n

Theorem 7. $E[\psi_i(S_i, e)] \geq \frac{1}{n} A(i+1)[f(OPT) - f(S_i)] - \frac{1}{n} B(i+1) g(OPT)$

Proof.

$$\begin{aligned}
E\left[\psi_i\left(S_i, e_i\right)\right] &= \frac{1}{n} \cdot \sum_{e_i \in \Omega} \psi_i\left(S_i, e_i\right) \\
&\geq \frac{1}{n} \cdot \sum_{e_i \in OPT} \left[A\left(i+1\right) f\left(e_i \mid S_i\right) - B\left(i+1\right) g\left(e_i \mid \Omega \setminus e_i\right)\right] \\
&= \frac{1}{n} A\left(i+1\right) \sum_{e \in OPT} f\left(e_i | S_i\right) - \frac{1}{n} B\left(i+1\right) \sum_{e \in OPT} g\left(e_i | \Omega \setminus e_i\right) \qquad (6)\\
&\geq \frac{1}{n} A\left(i+1\right)\left[f\left(OPT \cup S_i\right) - f\left(S_i\right)\right] - \frac{1}{n} B\left(i+1\right) g\left(OPT\right) \\
&\geq \frac{1}{n} A\left(i+1\right)\left[f\left(OPT\right) - f\left(S_i\right)\right] - \frac{1}{n} B\left(i+1\right) g\left(OPT\right)
\end{aligned}$$

□

Corollary 4. $E[\phi_{i+1}\left(S_{i+1}\right) - \phi_i\left(S_i\right)] \geq \frac{1}{n} A(i+1) f(OPT) - \frac{1}{n} B(i+1) g(OPT) + [(1 - \frac{1}{n}) A(i+1) - A(i)] f(S_i) - [B(i+1) - B(i)] \sum_{e \in S_i} g(e | \Omega \setminus e)$

With Theorem 7 and Corollary 4, using the same method as Section 3, we can prove the Random Conditioned Greedy can get the following approximation.

Theorem 8. *S_n is the solution of Algorithm 4 after n iteration*

$$E\left[A\left(n\right) f\left(S_n\right) - \left(B\left(0\right) - c_g\right) g\left(S_n\right)\right] \geq \frac{1}{n} \sum_{i=0}^{n-1} \left[A\left(i+1\right) f\left(OPT\right) - B\left(i+1\right) g\left(OPT\right)\right]$$

where c_g is the curvature of function g.

Proof.

$$\begin{aligned}
& E\left[A\left(n\right) f\left(S_n\right) - B\left(n\right)\left(1 - c_g\right) g\left(S_n\right)\right] \\
& \geq E\left[\phi_n\left(S_n\right)\right] - E\left[\phi_0\left(S_0\right)\right] = \sum_{i=0}^{n-1} E[\phi_{i+1}(S_{i+1})] - E[\phi_i(S_i)] \\
& \geq \sum_{i=0}^{n-1} \left[\frac{1}{n} A\left(i+1\right) f\left(OPT\right) - \frac{1}{n} B\left(i+1\right) g\left(OPT\right) + \left[\left(1 - \frac{1}{n}\right) A\left(i+1\right) - A\left(i\right)\right] f\left(S_i\right)\right] \\
& - \sum_{i=0}^{n-1} \left[B\left(i+1\right) - B\left(i\right)\right] \sum_{e \in S_i} g\left(e | \Omega \setminus e\right) \\
& \geq \frac{1}{n} \sum_{i=0}^{n-1} \left[A\left(i+1\right) f\left(OPT\right) - B\left(i+1\right) g\left(OPT\right)\right] - \left[B\left(n\right) - B\left(0\right)\right] \sum_{e \in S_n} g\left(e | \Omega \setminus e\right) \\
& \geq \frac{1}{n} \sum_{i=0}^{n-1} \left[A\left(i+1\right) f\left(OPT\right) - B\left(i+1\right) g\left(OPT\right)\right] - \left[B\left(n\right) - B\left(0\right)\right] g\left(S_n\right)
\end{aligned} \qquad (7)$$

Therefore, we can conclude

$$E\left[A\left(n\right)f\left(S_n\right)-\left(B\left(0\right)-c_g\right)g\left(S_n\right)\right]\geq\frac{1}{n}\sum_{i=0}^{n-1}\left[A\left(i+1\right)f\left(OPT\right)-B\left(i+1\right)g\left(OPT\right)\right]$$

□

From the Theorem 7 and Theorem 8, we find an interesting phenomenon. If we decrease the number of iterations to k , this Random Conditioned Greedy can also be used in problems with cardinality constrained. Since this proof is similar with Theorem 8, we just give the statement without proof.

Theorem 9. *S_k is the solution of Algorithm 4 after k iteration*

$$E\left[A\left(k\right)f\left(S_k\right)-\left(B\left(0\right)-c_g\right)g\left(S_k\right)\right]\geq\frac{1}{n}\sum_{i=0}^{k-1}\left[A\left(i+1\right)f\left(OPT\right)-B\left(i+1\right)g\left(OPT\right)\right]$$

where c_g is the curvature of function g.

6 Two Special Cases for Random Conditioned Greedy

In this section, we choose the same parameters as Deterministic Conditioned Greedy to show Random Conditioned Greedy strategy can get $E[f(S_n)-(e^{-1}-c_g)g(S_n)]\geq(1-e^{-1})[f\left(OPT\right)-g\left(OPT\right)]$ approximation and $E[f(S_n)-(1-c_g)g(S_n)]\geq(1-e^{-1})f(OPT)-g(OPT)$ approximation for unconstrained problem. Since the proof is similar with Section 5, we only give the statement without proofs.

6.1 Case 1

We set $A\left(i\right)=B\left(i\right)=\left(1-\frac{1}{n}\right)^{(n-i)}$. Therefore, the definition 1 and Algorithm 4 become the following. Obviously, these settings satisfy all assumptions in Section 5.

$$\phi_i\left(T\right)=\left(1-\frac{1}{n}\right)^{n-i}\left[f\left(T\right)-\sum_{e\ \in T}g\left(e\mid\Omega\setminus e\right)\right]$$

$$\psi_i\left(T,e\right)=\max\left\{0,\left(1-\frac{1}{n}\right)^{n-(i+1)}\left[f\left(e\mid S_i\right)-g\left(e\mid\Omega\setminus e\right)\right]\right\}$$

Theorem 10. $E[\psi_i(S_i,e_i)]\geq\frac{1}{n}(1-\frac{1}{n})^{n-(i+1)}[f(OPT)-f(S_i)-g(OPT)]$

Theorem 11. $E\left[f\left(S_n\right)-\left(e^{-1}-c_g\right)g\left(S_n\right)\right]\geq\left(1-e^{-1}\right)\left[f\left(OPT\right)-g\left(OPT\right)\right]$, *where c_g is the curvature of function g*

From Theorem 11, we can draw the following corollary when g is modular function.

Corollary 5. *If $c_g=0$, i.e. g is modular. Then we have*

$$E\left[f\left(S_n\right)-e^{-1}g\left(S_n\right)\right]\geq\left(1-e^{-1}\right)\left[f\left(OPT\right)-g\left(OPT\right)\right]$$

Algorithm 5: Random Conditioned Greedy I
Input: ground set Ω
1.Initialize $S_0 \leftarrow \emptyset$
2.For $i = 0$ to $n-1$
3. $e_i \leftarrow$be chosen uniformly from Ω
4. If $\left(1-\frac{1}{k}\right)^{k-(i+1)}\left[f\left(e_i \mid S_i\right)-g\left(e_i \mid \Omega \setminus e_i\right)\right]>0$ then
5. $S_{i+1} \leftarrow S_i \cup \{e_i\}$
6. Else
7. $S_{i+1} \leftarrow S_i$
8.End for
9.Return S_n

6.2 Case 2

We set $A\left(i\right)=\left(1-\frac{1}{n}\right)^{(n-i)}$, $B\left(i\right)=1$. Therefore, the definition 1 and Algorithm 4 become the following. Obviously, these settings satisfy all assumptions in Section 5.

$$\phi_i\left(T\right)=\left(1-\frac{1}{n}\right)^{n-i} f\left(T\right)-\sum_{e\ \in T} g\left(e \mid \Omega \setminus e\right)$$

$$\psi_i\left(T,e\right)=\max\left\{0,\left(1-\frac{1}{n}\right)^{n-(i+1)} f\left(e \mid S_i\right)-g\left(e \mid \Omega \setminus e\right)\right\}$$

Algorithm 6: Random Conditioned Greedy II
Input: ground set Ω
1.Initialize $S_0 \leftarrow \emptyset$
2.For $i = 0$ to $n-1$
3. $e_i \leftarrow$be chosen uniformly from Ω
4. If $\left(1-\frac{1}{k}\right)^{k-(i+1)} f\left(e_i \mid S_i\right)-g\left(e_i \mid \Omega \setminus e_i\right)>0$ then
5. $S_{i+1} \leftarrow S_i \cup \{e_i\}$
6. Else
7. $S_{i+1} \leftarrow S_i$
8.End for
9.Return S_n

Theorem 12. $E[\psi_i(S_i,e_i)] \geq \frac{1}{n}(1-\frac{1}{n})^{n-(i+1)}[f(OPT)-f(S_i)]-\frac{1}{n}g(OPT)$

Theorem 13. $E\left[f\left(S_n\right)-\left(1-c_g\right)g\left(S_n\right)\right] \geq \left(1-e^{-1}\right)f\left(OPT\right)-g\left(OPT\right)$, *where* c_g *is the curvature of function* g

From Theorem 13, we can draw the following corollary when g is modular function.

Corollary 6. *If* $c_g = 0$*, i.e.* g *is modular. Then we have*

$$E\left[f\left(S_n\right)-g\left(S_n\right)\right] \geq \left(1-e^{-1}\right)f\left(OPT\right)-g\left(OPT\right)$$

Remark: Obviously, if g is modular, then $h = f - g$ is submodular but non-monotone. The above approximations are different with non-monotone submodular maximization under unconstrained problem 1/2. But we cannot measure which one is better than others. In some numerical cases, our approximations may be better than 1/2. This give us a clue that we can choose the appropriate parameters $A(i)$ and $B(i)$ according to the characteristics of the problem to get a better approximate ratio.

7 Conclusions

In this paper, we propose Conditioned Greedy strategy with deterministic and random which are general frameworks for maximum DS decomposition under cardinality constrained and unconstrained respectively. Users can choose some rational parameters to fit special practical problems and get a wonderful approximation about problem. Also, we choose two special cases show our strategy can get some novel approximation. In some situations, these novel approximations are better than the best approximation at the state of art.

In the future works, how to remove the curvature parameter in approximation ratio is important, because it can make the approximation much tight. What's more, how to select $A(i)$ and $B(i)$ so that the algorithm can achieve the optimal approximation ratio is also urgent problem to be solved.

References

1. Bai, W., Bilmes, J.A.: Greed is still good: Maximizing monotone submodular+supermodular functions (2018)
2. Buchbinder, N., Feldman, M., Naor, J.S., Schwartz, R.: A tight linear time (1/2)-approximation for unconstrained submodular maximization. IEEE Computer Society, USA (2012). https://doi.org/10.1109/FOCS.2012.73, https://doi.org/10.1109/FOCS.2012.73
3. Buchbinder, N., Feldman, M., Naor, J.S., Schwartz, R.: Submodular maximization with cardinality constraints. Society for Industrial and Applied Mathematics, USA (2014)
4. Das, A., Kempe, D.: Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. Computer ence (2011)
5. Feldman, M., Naor, J.S., Schwartz, R.: A unified continuous greedy algorithm for submodular maximization. IEEE Computer Society, USA (2011). https://doi.org/10.1109/FOCS.2011.46, https://doi.org/10.1109/FOCS.2011.46
6. Fisher, M.L., Nemhauser, G.L., Wolsey, L.A.: An analysis of approximations for maximizing submodular set functions—ii. Mathematical Programming **8**(1), 73–87 (1978)
7. Golovin, D., Krause, A.: Adaptive submodularity: Theory and applications in active learning and stochastic optimization. Journal of Artificial Intelligence Research **42**(1), 427–486 (2012)
8. Han, K., Xu, C., Gui, F., Tang, S., Huang, H., Luo, J.: Discount allocation for revenue maximization in online social networks. pp. 121–130 (06 2018). https://doi.org/10.1145/3209582.3209595

9. Harshaw, C., Feldman, M., Ward, J., Karbasi, A.: Submodular maximization beyond non-negativity: Guarantees, fast algorithms, and applications. vol. 2019-June, pp. 4684 – 4705. Long Beach, CA, United states (2019)
10. Horel, T., Singer, Y.: Maximization of approximately submodular functions. In: Advances in neural information processing systems (2016)
11. Iyer, R., Bilmes, J.: Algorithms for approximate minimization of the difference between submodular functions, with applications. AUAI Press, Arlington, Virginia, USA (2012)
12. Jegelka, S., Bilmes, J.: Submodularity beyond submodular energies: Coupling edges in graph cuts. IEEE Computer Society (2011)
13. Kempe, D.: Maximizing the spread of influence through a social network. Proc.of Acm Sigkdd Intl Conf.on Knowledge Discovery and Data Mining (2003)
14. Krause, A., Singh, A., Guestrin, C.: Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. Journal of Machine Learning Research **9**(3), 235–284 (2008)
15. Li, X., Du, H.G., Pardalos, P.M.: A variation of ds decomposition in set function optimization. Journal of Combinatorial Optimization (2020)
16. Lin, H., Bilmes, J.: A class of submodular functions for document summarization. Association for Computational Linguistics, USA (2011)
17. Lu, W., Chen, W., Lakshmanan, L.V.S.: From competition to complementarity: Comparative influence diffusion and maximization. proceedings of the vldb endowment **9**(2), 60–71 (2015)
18. Maehara, Takanori, Murota, Kazuo: A framework of discrete dc programming by discrete convex analysis. Mathematical Programming (2015)
19. Michele, Conforti, , , Gérard, Cornuéjols: Submodular set functions, matroids and the greedy algorithm: Tight worst-case bounds and some generalizations of the rado-edmonds theorem. Discrete Applied Mathematics (1984)
20. Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An analysis of approximations for maximizing submodular set functions—i. Mathematical Programming **14**(1), 265–294 (1978)
21. Wang, Z., Yang, Y., Pei, J., Chu, L., Chen, E.: Activity maximization by effective information diffusion in social networks. IEEE Transactions on Knowledge and Data Engineering **29**(11), 2374–2387 (2017)
22. Yang, W., Chen, S., Gao, S., Yan, R.: From competition to complementarity: Comparative influence diffusion and maximization. Journal of Combinatorial Optimization (2020)
23. Yu, J., Blaschko, M.: A convex surrogate operator for general non-modular loss functions (04 2016)
24. Zhu, J., Ghosh, S., Zhu, J., Wu, W.: Near-optimal convergent approach for composed influence maximization problem in social networks. IEEE Access **PP**(99), 1–1 (2019)