

# Robust Sequence Networked Submodular Maximization

Qihao Shi<sup>1,2</sup>, Bingyang Fu<sup>1</sup>, Can Wang<sup>1\*</sup>, Jiawei Chen<sup>1</sup>, Sheng Zhou<sup>1</sup>, Yan Feng<sup>1</sup>, Chun Chen<sup>1</sup>

<sup>1</sup>School of Computing and Computer Science, Zhejiang University, Hangzhou, China

<sup>2</sup>School of Computing and Computer Science, Zhejiang University City College, Hangzhou, China  
{shiqihao321, todobe, wcan, sleepyhunt, zhousheng\_zju, fengyan, chenc}@zju.edu.cn

## Abstract

In this paper, we study the Robust optimization for sequence Networked submodular maximization (RoseNets) problem. We interweave the robust optimization with the sequence networked submodular maximization. The elements are connected by a directed acyclic graph and the objective function is not submodular on the elements but on the edges in the graph. Under such networked submodular scenario, the impact of removing an element from a sequence depends both on its position in the sequence and in the network. This makes the existing robust algorithms inapplicable. In this paper, we take the first step to study the RoseNets problem. We design a robust greedy algorithm, which is robust against the removal of an arbitrary subset of the selected elements. The approximation ratio of the algorithm depends both on the number of the removed elements and the network topology. We further conduct experiments on real applications of recommendation and link prediction. The experimental results demonstrate the effectiveness of the proposed algorithm.

## Introduction

Submodularity is an important property that models a diminishing return phenomenon, i.e., the marginal value of adding an element to a set decreases as the set expands. It has been extensively studied in the literature, mainly accompanied with maximization or minimization problems of set functions (Nemhauser and Wolsey 1978; Khuller, Moss, and Naor 1999). This is called submodularity. Mathematically, a set function  $f : 2^V \rightarrow \mathbb{R}$  is submodular if for any two sets  $A \subseteq B \subseteq V$  and an element  $v \in V \setminus B$ , we have  $f(A \cup v) - f(A) \geq f(B \cup v) - f(B)$ . Such property finds a wide range of applications in machine learning, combinatorial optimization, economics, and so on. (Krause 2005; Lin and Bilmes 2011; Li et al. 2022; Shi et al. 2021; Kirchhoff and Bilmes 2014; Gabillon et al. 2013; Kempe, Kleinberg, and Tardos 2003; Wang et al. 2021).

Further equipped with monotonicity, i.e.,  $f(A) \leq f(B)$  for any  $A \subseteq B \subseteq V$ , a submodular set function can be maximized by the cardinality constrained classic greedy algorithm, achieving an approximation ratio up to  $1 - 1/e$  (Nemhauser and Wolsey 1978) (almost the best). Since

then, the study of submodular functions has been extended by a variety of different scenarios, such as non-monotone scenario, adaptive scenario, and continuous scenario, etc (Feige, Mirrokni, and Vondrak 2007; Golovin and Krause 2011; Das and Kempe 2011; Bach 2019; Shi et al. 2019).

The above works focus on the set functions. In real applications, the order of adding elements plays an important role and affects the function value significantly. Recently, the submodularity has been generalized to sequence functions (Zhang et al. 2015; Tschitschek, Singla, and Krause 2017; Streeter and Golovin 2008; Zhang et al. 2013). Considering sequences instead of sets causes an exponential increase in the size of the search space, while allowing for much more expressive models.

In this paper, we consider that the elements are networked by a directed graph. The edges encode the additional value when the connected elements are selected in a particular order. Such setting is not given a specific name before. To distinguish from the classic submodularity, we in this paper name it as networked submodularity (Net-submodularity for short). More specifically, the Net-submodular function  $f(\sigma)$  is a sequence function, which is not submodular on the induced element set by  $\sigma$  but is submodular on the induced edge set by  $\sigma$ . The Net-submodularity is first considered in (Tschitschek, Singla, and Krause 2017), which mainly focuses on the case where the underlying graph is a directed acyclic graph. General graphs and hypergraphs are considered in (Mitrovic et al. 2018).

Recently, robust versions of the submodular maximization problem have arisen (Orlin, Schulz, and Udmani 2018; Mitrovic et al. 2017; Bogunovic et al. 2017; Sallam et al. 2020) to meet the increasing demand in the stability of the system. The robustness of the model mainly concerns with its ability in handling the malfunctions or adversarial attacks, i.e., the removal of a subset of elements in the selected set or sequence. Sample cases of elements removal in real world scenarios include items sold out or stop production in recommendation (Mitrovic et al. 2018), web failure of user logout in link prediction (Mitrovic et al. 2019) and equipment malfunction in sensor allocation or activation (Zhang et al. 2015). In this paper, we take one step further and study a new problem of robust sequence networked submodular maximization (RoseNets). We show an example in Figure 1 to illustrate the importance of RoseNets problem.

\*Corresponding Author.

Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

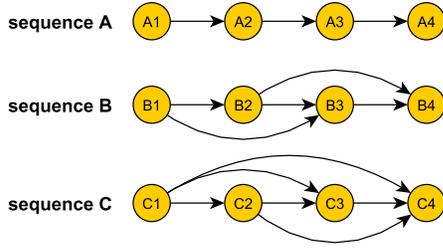


Figure 1: Example of RoseNets

See in Figure 1. Suppose all edge weights in sequence A are 0.9, in sequence B are 0.5, in sequence C are 0.4. Let the net-submodular utility function  $f$  of the sequence be the summation of all the weights of the induced edge set by the sequence. Such utility function is obviously monotone but not submodular.<sup>1</sup> However, it is submodular on the edge set. Now we can see, the utility of sequence A, B and C are 2.7 (largest), 2.5 and 2.4 respectively. We can easily check that if one node would be removed in each sequence, the worst utility after removal of sequence A, B and C is 0.9, 1.0 (largest), and 0.8. If we remove two nodes in each sequence, the utility of A, B and C becomes 0, 0, and 0.4 (largest). With different number of nodes removed, the three sequences show different robustness. Existing non-robust algorithm may select sequence A since it has the largest utility. However, sequence B and C are more robust against node removal.

Given a net-submodular function and the corresponding network, the RoseNets problem aims to select a sequence of elements with cardinality constraints, such that the value of the sequence function is maximized when a certain number of the selected elements may be removed. As far as sequence functions and net-submodularity are concerned, the design and analysis of robust algorithms are faced with novel technical difficulties. The impact of removing an element from a sequence depends both on its position in the sequence and in the network. This makes the existing robust algorithms inapplicable here. It is unclear what conditions are sufficient for designing efficient robust algorithm with provable approximation ratios for RoseNets problem. We aim to take a step for answering this question in this paper. Our contributions are summarized as follows.

1. To the best of our knowledge, this is the first work that considers the RoseNets problem. Combining robust optimization and sequence net-submodular maximization requires subtle yet critical theoretical efforts.
2. We design a robust greedy algorithm that is robust against the removal of an arbitrary subset of the selected sequence. The theoretical approximation ratio depends both on the number of the removed elements and the network topology.
3. We conduct experiments on real applications of recommendation and link prediction. The experimental results demonstrate the effectiveness and robustness of the proposed algorithm, against existing sequence submodular

<sup>1</sup>Easy to see that in sequence B, the utility of  $\{B4\}$  is 0, but  $\{B3, B4\}$  is 0.5, which violates the submodularity.

baselines. We hope that this work serves as an important first step towards the design and analysis of efficient algorithms for robust submodular optimization.

## Related Works

Submodular maximization has been extensively studied in the literature. Efficient approximation algorithms have been developed for maximizing a submodular set function in various settings (Nemhauser and Wolsey 1978; Khuller, Moss, and Naor 1999; Calinescu et al. 2011; Chekuri, Vondrák, and Zenklusen 2014). By considering the robustness requirement, recently, robust versions of submodular maximization have been extensively studied. These works aim at selecting a set of elements that is robust against the removal of a subset of elements. The first algorithm for the cardinality constrained robust submodular maximization problem is studied in (Orlin, Schulz, and Udvani 2018). A constant factored approximation ratio is achieved. The selected  $k$ -sized set is robust against the removal of any  $\tau$  elements of the selected set. The constant approximation ratio is valid as long as  $\tau = O(\sqrt{k})$ . An improvement is made in (Bogunovic et al. 2017), which provide an algorithm that guarantees the same constant approximation ratio but allows the removal of a larger number of elements (i.e.,  $\tau = O(k)$ ). With a mild assumption, the algorithm proposed in (Mitrovic et al. 2017) allows the removal of an arbitrary number of elements. The restriction on  $\tau$  is relaxed in (Tzoumas et al. 2017), while the derived approximation ratio is parameterized  $\tau$ . This work is extended to a multi-stage setting in (Tzoumas, Jadbabaie, and Pappas 2018) and (Tzoumas, Jadbabaie, and Pappas 2020). The decision at each stage would take into account the failures that happened in the previous stages. Other constraints that are combined with the robust optimization include fairness, privacy issues and so on (Mirzasoleiman, Karbasi, and Krause 2017; Kazemi, Zadimoghaddam, and Karbasi 2018).

The concept of sequence (or string) submodularity for sequence functions is a generalization of submodularity, which has been introduced recently in several studies (Zhang et al. 2015; Streeter and Golovin 2008; Zhang et al. 2013). The above works all consider element-based robust submodular maximization. Networked submodularity is considered in (Tschitschek, Singla, and Krause 2017; Mitrovic et al. 2018), where the sequential relationship among elements is encoded by a directed acyclic graph. Following the networked submodularity setting, the work in (Mitrovic et al. 2019) introduces the idea of adaptive sequence submodular maximization, which aims to utilize the feedback obtained in previous iterations to improve the current decision. In this paper, we follow the networked submodularity setting, and study the RoseNets problem. It is unclear whether all of the above algorithms can be properly extended to our problem, as converting a set function to a sequence function and submodularity to networked submodularity could result in an arbitrarily bad performance. Establishing the approximation guarantees for RoseNets problem would require a more sophisticated analysis, which calls for more in-depth theoretical efforts.

## System Model and Problem Definition

In this paper, we follow the networked submodular sequence setting (Tschitschek, Singla, and Krause 2017; Mitrovic et al. 2018). Let  $V = \{v_1, v_2, \dots, v_n\}$  be the set of  $n$  elements. A set of edges  $E$  represents that there is additional utility in picking certain elements in a certain order. More specifically, an edge  $e_{ij} = (v_i, v_j)$  represents that there is additional utility in selecting  $v_j$  after  $v_i$  has already been chosen. Self-loops (i.e., edges that begin and end at the same element) represents that there is individual utility in selecting an element.

Given a directed graph  $G = (V, E)$ , a non-negative monotone submodular set function  $h : 2^E \rightarrow \mathbb{R}_{\geq 0}$ , and a parameter  $k$ , the objective is to select a non-repeating sequence  $\sigma$  of  $k$  unique elements that maximizes the objective function:

$$f(\sigma) = h(E(\sigma)),$$

where  $E(\sigma)$  contains all the edges  $(v_i, v_j) \in E$  that  $v_i$  is select before  $v_j$  in  $\sigma$ .

We say  $E(\sigma)$  is the set of edges induced by the sequence  $\sigma$ . It is important to note that the function  $h$  is a submodular set function over the edges, not over the elements. Furthermore, the objective function  $f$  is neither a set function, nor is it necessarily submodular on the elements. We call such a function  $f(\sigma)$  as a *networked submodular* function.

We define  $f(\sigma - Z)$  to represent the residual value of the objective function after the removal of elements in set  $Z$ . In this paper, the robust sequence networked submodular maximization (RoseNets) problem is formally defined below.

**Definition 1.** Given a directed graph  $G = (V, E)$ , a networked submodular function  $f(\cdot)$  and robustness parameter  $\tau$ , the RoseNets problem aims at finding a sequence  $\sigma$  such that it is robust against the worst possible removal of  $\tau$  nodes:

$$\max_{\sigma: |\sigma| \leq k} \min_{Z \in \sigma, |Z| \leq \tau} f(\sigma - Z).$$

The robustness parameter  $\tau$  represents the size of the subset  $Z$  that is removed. After the removal, the objective value should remain as large as possible. For  $\tau = 0$ , the problem reduces to the classic sequence submodular maximization problem (Mitrovic et al. 2018).

## Robust Algorithm and Theoretical Results

Direct applying the Sequence Greedy algorithm (Mitrovic et al. 2018) to solve the RoseNets problem would return an arbitrary bad solution. We can construct a very simple example for an illustration. See in Figure 2. Let the edge weights of  $(A, B)$ ,  $(B, C)$ ,  $(B, E)$ ,  $(B, F)$  be 0.9 and  $(C, D)$ ,  $(C, G)$ ,  $(D, G)$  be 0.5. Let the net-submodular utility function  $f$  of the selected sequence be the summation of all weights of the induced edge set by the sequence. Suppose we are to select a sequence with 5 elements. Using the Sequence Greedy algorithm, sequence  $\langle A, B, C, E, F \rangle$  will be selected<sup>2</sup> for maximizing the utility, i.e.,  $(0.9 \cdot 4 = 3.6)$ .

<sup>2</sup>Suppose elements are selected in the alphabetic order if the edge weights are equal. Similar examples can be easily constructed when elements are selected at random.

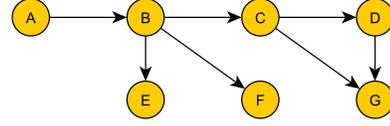


Figure 2: Example of Sequence Greedy

However, if  $\tau = 2$ , i.e., two elements would be removed, removing  $B$  and any other one element (worst case) makes the utility become 0.

## RoseNets Algorithm

We wish to design an algorithm that is robust against the removal of an arbitrary subset of  $\tau$  selected elements. In this paper, we propose the RoseNets Algorithm, which can approximately solves the RoseNets problem and is shown in Algorithm 1. Note we consider the case that  $k \geq 3$ .

The limitation of the Sequence Greedy algorithm is that the selected sequence is vulnerable. The overall utility might be concentrated in the first few elements. Algorithm 1 is motivated by this key observation and works in two steps. In Step 1 (the first *while* loop), we select a sequence  $\sigma_1$  of  $\tau$  elements from  $V$  in a greedy manner as in Sequence Greedy. In Step 2 (the second *while* loop), we select another sequence  $\sigma_2$  of  $k - \tau$  elements from  $V \setminus \sigma_1$ , again in a greedy manner as in Sequence Greedy. Note that when we select sequence  $\sigma_2$ , we perform the greedy selection as if sequence  $\sigma_1$  does not exist at all. This ensures that the value of the final returned sequence  $\sigma = \sigma_1 \oplus \sigma_2$  is not concentrated in either  $\sigma_1$  or  $\sigma_2$ . The complexity of Algorithm 1 is  $O(k|E|)$ , which is in terms of the number of function evaluations used in the algorithm.

To show the differences and benefits of the RoseNets algorithm, we go back to see the example in Figure 2. When  $k = 5$  and  $\tau = 2$ , the RoseNets algorithm will select the sequence  $\sigma_1 = \langle A, B \rangle$  and sequence  $\sigma_2 = \langle C, D, G \rangle$ . When selecting  $\sigma_2$ , the RoseNets algorithm would not consider element  $B$  in  $\sigma_1$ . Thus element  $E$  and  $F$  are regarded as making not contribution to the utility function. The RoseNets algorithm will return the sequence  $\sigma = \langle A, B, C, D, G \rangle$ . The worst case of removing  $\tau = 2$  elements, is to remove  $B$  and any one element in  $\{C, D, G\}$ . The residual utility is 0.5. Remember the case for Sequence Greedy algorithm, the residual utility of the worst case is 0. This example shows the benefits of the RoseNets algorithm.

Both the examples in Figure 1 and Figure 2 imply that a robust sequence should have complex network structure. The utility should not concentrate in a center element, but aggregated from all the edges among elements in the sequence. Such a robust sequence can only be selected by multiple trials of greedy selection, with the trials neglecting each other. Otherwise, the central element (if exists) with its high edge weight neighbors are probability selected, as node  $B$  in Figure 2. If such a node is removed, the utility would slump. In this paper, we implement such intuitive strategy by using two trials of greedy selection. Intuitively, invoking more times of greedy selection trial may improve the approximation ratio. However, as we are to take a first step in the

---

**Algorithm 1: RoseNets Algorithm**


---

```

1  $\sigma = \emptyset, \sigma_1 = \emptyset, \sigma_2 = \emptyset;$ 
2 while  $|\sigma_1| < \tau$  do
3   if  $|\sigma_1| = \tau - 1$  then
4      $E' = \{e_{ij} | v_j \notin \sigma_1\} \wedge (v_i = v_j \vee v_i \in \sigma_1);$ 
5      $e_{ij} = \arg \max_{e_{ij} \in E'} h(e|E(\sigma_1));$ 
6      $\sigma_1 = \sigma_1 \oplus v_j;$ 
7   else
8      $e_{ij} = \arg \max_{\{e_{ij} | v_j \notin \sigma_1\}} h(e|E(\sigma_1));$ 
9     if  $v_j = v_i$  or  $v_i \in \sigma_1$  then
10       $\sigma_1 = \sigma_1 \oplus v_j;$ 
11     else
12       $\sigma_1 = \sigma_1 \oplus v_i \oplus v_j;$ 
13 while  $|\sigma_2| < k - \tau$  do
14   if  $|\sigma_2| = k - \tau - 1$  then
15      $E' = \{e_{ij} | v_j \notin \sigma_1 \cup \sigma_2 \wedge (v_i = v_j \vee v_i \in \sigma_2)\};$ 
16      $e_{ij} = \arg \max_{e_{ij} \in E'} h(e|E(\sigma_2));$ 
17      $\sigma_2 = \sigma_2 \oplus v_j;$ 
18   else
19      $e_{ij} = \arg \max_{\{e_{ij} | v_i \notin \sigma_1, v_j \notin \sigma_1 \cup \sigma_2\}} h(e|E(\sigma_2));$ 
20     if  $v_j = v_i$  or  $v_i \in \sigma_2$  then
21       $\sigma_2 = \sigma_2 \oplus v_j;$ 
22     else
23       $\sigma_2 = \sigma_2 \oplus v_i \oplus v_j;$ 
24  $\sigma = \sigma_1 \oplus \sigma_2;$ 
25 return  $\sigma$ 

```

---

RoseNets problem while our theoretical analysis is already non-trivial, we leave the design of multi-part selection algorithm and the approximation ratio analysis for future works.

### Theoretical Results

Let  $\alpha = 2d_{in} + 1, \beta = 1 + d_{in} + d_{out}, \gamma = e^{\frac{k-3}{k-2}}, \eta = e^{\frac{k-2\tau-1}{k-\tau-1}}$ . Note  $d_{in}$  and  $d_{out}$  are the maximum in and out degree of the network respectively. For convenience, we denote  $f(v|\sigma)$  and  $f(\sigma'|\sigma)$  as the marginal gain of attending  $v$  and  $\sigma'$  to sequence  $\sigma$  respectively. We denote  $\sigma^*(V, k, \tau)$  as the optimal solution of the RoseNets problem with element set  $V$ , cardinality  $k$  and robust parameter  $\tau$ , and  $g_\tau(\sigma)$  be the minimum value of  $f(\sigma)$  after  $\tau$  elements are removed from  $\sigma$ .

**Theorem 1.** Consider  $\tau = 1$ , Algorithm 1 achieves an approximation ratio of

$$\max\left\{\frac{1 - e^{-(1-1/k)}}{\alpha\beta}, \frac{\gamma^{\frac{1}{d_{in}}} - 1}{\beta\gamma^{\frac{1}{d_{in}}} - 1}\right\}.$$

**Theorem 2.** Consider  $1 \leq \tau \leq k$ , Algorithm 1 achieves an approximation ratio of

$$\max\left\{\frac{1 - e^{-(1-1/k)}}{\alpha\beta}, \frac{\tau\alpha\beta(\eta^{\frac{1}{d_{in}}} - 1)}{\tau\alpha\eta^{\frac{1}{d_{in}}} - \beta(1 - e^{-(1-1/k)})}\right\}.$$

In Theorem 1, it is hard to compare the two approximation ratios directly due to their complex mathematical expression. Thus we consider specific network setting to show the different advantages of the two terms.

First, it is easy to verify that both the two terms are monotonically increasing function of  $k$ . When  $k = 3$ , we have  $\frac{1 - e^{-(1-1/k)}}{\alpha\beta} - \frac{\gamma^{\frac{1}{d_{in}}} - 1}{\beta\gamma^{\frac{1}{d_{in}}} - 1} = \frac{1 - e^{-2/3}}{\alpha\beta} > 0$ . Thus we know that when  $k$  is small, the first term is larger. When  $k \rightarrow \infty$ , the first term has a limit value of  $(1 - 1/e)/\alpha\beta$ , and the second term has a limit value of  $\frac{e^{\frac{1}{d_{in}}} - 1}{\beta e^{\frac{1}{d_{in}}} - 1}$ . When  $d_{in} = 1$ , then  $\alpha = 3$  and  $(1 - 1/e)/\alpha\beta - \frac{e^{\frac{1}{d_{in}}} - 1}{\beta e^{\frac{1}{d_{in}}} - 1} = (1 - 1/e)/3\beta - \frac{e-1}{\beta e-1} = \frac{-1-2\beta e + \frac{1}{e} + 2\beta}{3\beta(\beta e-1)} < 0$ . In this case, the second term is larger than the first term in Theorem 1. Thus we can conclude that under specific network structure, the second term would be larger with large  $k$ .

Similarly, for Theorem 2, when  $k = 3$  and  $\tau = 1$ ,  $\frac{1 - e^{-(1-1/k)}}{\alpha\beta} - \frac{\tau\alpha\beta(\eta^{\frac{1}{d_{in}}} - 1)}{\tau\alpha\eta^{\frac{1}{d_{in}}} - \beta(1 - e^{-(1-1/k)})} = \frac{1 - e^{-2/3}}{\alpha\beta} > 0$ . Thus we know when  $k$  and  $\tau$  is small, the first term is larger. When  $k \rightarrow \infty$  while  $\tau$  remains a constant, the first term has a limit value of  $(1 - 1/e)/\alpha\beta$ , the second term has a limit value of  $\frac{\tau\alpha\beta(e^{\frac{1}{d_{in}}} - 1)}{\tau\alpha e^{\frac{1}{d_{in}}} - \beta(1 - e^{-1})}$ . When  $d_{in} = 1$  and  $d_{out} < \frac{3\tau e^2}{e-1} - 2$ , then  $\alpha = 3, \beta < \frac{3\tau e^2}{e-1}$  and  $(1 - 1/e)/\alpha\beta - \frac{\tau\alpha\beta(e^{\frac{1}{d_{in}}} - 1)}{\tau\alpha e^{\frac{1}{d_{in}}} - \beta(1 - e^{-1})} = \frac{(1 - \frac{1}{e})(3\tau e - \beta(1 - \frac{1}{e})) - 9\beta^2\tau(e-1)}{3\beta(3\tau e - \beta(1 - \frac{1}{e}))} < 0$ . In this case, the second term is larger than the first term in Theorem 2. Thus we can conclude that under specific network structure, the second term would be larger with large  $k$ .

According to the above analysis, we know that the value of  $k$  and  $\tau$ , together with the network topology, significantly affect the approximation ratio. It would be an interesting future direction to explore how the approximation ratio change when the parameters and network topology change.

To prove the above two theorems, we need the following three auxiliary lemmas. Due to space limitations, we here assume Lemma 1, 2 and 3 hold and show the proof of Theorem 1 below. We provide the proofs of Lemma 1, 2, 3 and Theorem 2 in the supplementary material.

**Lemma 1.** There exists an element  $v$  for sequence  $\sigma_1$  and  $\sigma_2$  satisfies that  $f(v|\sigma_1) \geq \frac{1}{d_{in}|\sigma_2|} f(\sigma_2|\sigma_1)$ .

**Lemma 2.** Consider  $c \in (0, 1]$  and  $1 \leq k' \leq k$ . Suppose that the sequence selected is  $\sigma$  with  $|\sigma| = k$  and that there exists a sequence  $\sigma'$  with  $|\sigma'| = k - k'$  such that  $\sigma' \subseteq \sigma$  and  $f(\sigma') \geq cf(\sigma)$ . Then we have  $f(\sigma) \geq \frac{e^{\frac{k'}{d_{in}k}} - 1}{e^{\frac{k'}{d_{in}k}} - c} f(\sigma^*(V, k, 0))$ .

**Lemma 3.** Consider  $1 \leq \tau \leq k$ . The following holds for any  $Z \subseteq V$  with  $|Z| \leq \tau$ :  $g_\tau(\sigma^*(V, k, \tau)) \leq f(\sigma^*(V - Z, k - \tau, 0))$ .

### Proof of Theorem 1

Given  $\tau = 1$ , the selected sequence  $\sigma_1$  has one element. And we have  $\sigma_2 = k - 1$ . Let  $\sigma_1 = \{v_1\}$ . Then the final sequence

is  $\sigma = \{v_1\} \oplus \sigma_2$ . Suppose the the removed vertex from the sequence is  $z$ .

First, we show a lower bound on  $f(\sigma_2)$ :

$$\begin{aligned} f(\sigma_2) &\geq \frac{1 - e^{-(1-1/k)}}{\alpha} f(\sigma^*(V \setminus v_1, k - 1, 0)) \\ &\geq \frac{1 - e^{-(1-1/k)}}{\alpha} g_\tau(\sigma^*(V, k, \tau)) \end{aligned} \quad (1)$$

The first inequality is due to the approximation ratio of Sequence Greedy algorithm for net-submodular maximization (Mitrovic et al. 2018). The second inequality is due to Lemma 3.

Now, we can see the removed element  $z$  can be either  $v_1$  or an element in  $\sigma_2$ . In the following, we will consider these two cases:

**Case 1.** Let  $z = v_1$ . Then we have

$$f(\sigma - z) = f(\sigma_2) \geq \frac{1 - e^{-(1-1/k)}}{\alpha} g_\tau(\sigma^*(V, k, \tau)) \quad (2)$$

**Case 2.** Let  $z \in \sigma_2$ . We then further consider two cases:

**Case 2.1.** Let  $f(\sigma_2) \leq f(\sigma_2 - z)$ .

In this case, the removal does not reduce the overall value of the remaining sequence  $\sigma_2 - \{z\}$ . Then we have

$$\begin{aligned} f(\sigma - v) &= f(v_1 \oplus (\sigma_2 - z)) \geq f(\sigma_2 - z) \\ &\geq f(\sigma_2) \geq \frac{1 - e^{-(1-1/k)}}{\alpha} g_\tau(\sigma^*(V, k, \tau)) \end{aligned} \quad (3)$$

**Case 2.2.** Let  $f(\sigma_2) > f(\sigma_2 - z)$ .

We define  $q = \frac{f(\sigma_2) - f(\sigma_2 - z)}{(d_{in} + d_{out})f(\sigma_2)}$ , to represent the ratio of the loss of removing element  $z$  from sequence  $\sigma_2$  to the value of the sequence  $\sigma_2$ . Obviously, we have  $q \in (0, \frac{1}{d_{in} + d_{out}}]$  since  $f(\sigma_2) > f(\sigma_2 - z)$ .

First, we have

$$\begin{aligned} (d_{in} + d_{out})qf(\sigma_2) &= f(\sigma_2) - f(\sigma_2 - z) \\ &= f(\sigma_2^1 \oplus z \oplus \sigma_2^2) - f(\sigma_2^1 \oplus \sigma_2^2) \\ &= f(\sigma_2^1) + f(z|\sigma_2^1) + f(\sigma_2^2|\sigma_2^1 \oplus z) \\ &\quad - f(\sigma_2^1) - f(\sigma_2^2|\sigma_2^1) \\ &= f(z|\sigma_2^1) + f(\sigma_2^2|\sigma_2^1 \oplus z) - f(\sigma_2^2|\sigma_2^1) \\ &\leq d_{in}h(e_z^{in}) + d_{out}h(e_z^{out}) \\ &\leq (d_{in} + d_{out}) \max\{h(e_z^{in}), h(e_z^{out})\} \end{aligned} \quad (4)$$

where  $h(e_z^{in})/h(e_z^{out})$  are the edge that has maximum utility over all the incoming/outgoing edges of  $z$ . The first inequality is due to the fact that the marginal gain of a vertex  $z$  to the prefix and subsequent sequence is at most  $d_{in}h(e_z^{in})$  and  $d_{out}h(e_z^{out})$ . Then the second inequality follows intuitively.

Given Equation (4), we need to prove four inequalities for finally proving the theorem.

First, suppose the first vertex of  $\sigma_2$  is  $v_2$ . By the monotonicity of function  $f(\cdot)$  and Equation (4), we have

$$\begin{aligned} f(\sigma - \{z\}) &\geq f(v_1 \oplus v_2) \\ &\geq \max\{h(e_z^{in}), h(e_z^{out})\} \geq qf(\sigma_2), \text{ and} \\ f(\sigma - \{z\}) &= f(v_1 \oplus (\sigma_2 - z)) \\ &\geq f(\sigma_2 - z) \geq (1 - (d_{in} + d_{out})q)f(\sigma_2) \end{aligned} \quad (5)$$

Given Equation (5), we have Inequality 1 as below.

**Inequality 1:**

$$\begin{aligned} f(\sigma - z) &\geq \max\{q \cdot f(\sigma_2), (1 - (d_{in} + d_{out})q) \cdot f(\sigma_2)\}. \\ \text{We know } \max\{x, 1 - bx\} &\geq \frac{1}{1+b} \text{ for } x \in (0, \frac{1}{b}] \text{ and } \\ b > 0.^3 \text{ Thus we have Inequality 2 as below.} \end{aligned}$$

**Inequality 2:**  $\max\{q, 1 - (d_{in} + d_{out})q\} \geq 1/\beta$ .

Note that the first two elements  $v_2, v_3$  in  $\sigma_2$  satisfy that

$$f(v_2 \oplus v_3) \geq \max\{h(e_z^{in}), h(e_z^{out})\} \geq qf(\sigma_2)$$

Thus by replacing the parameters in Lemma 2 and Lemma 3, we have the following result, which implies Inequality 3.

$$\begin{aligned} f(\sigma_2) &\geq \frac{\gamma^{\frac{1}{d_{in}}} - 1}{\gamma^{\frac{1}{d_{in}}} - q} f(\sigma^*(V \setminus v_1, k - \tau, 0)) \\ \implies \text{Inequality 3: } f(\sigma_2) &\geq \frac{\gamma^{\frac{1}{d_{in}}} - 1}{\gamma^{\frac{1}{d_{in}}} - q} g_\tau(\sigma^*(V, k, \tau)) \end{aligned}$$

Now define  $\ell_1(q) = q \frac{\gamma^{\frac{1}{d_{in}} - 1}}{\gamma^{\frac{1}{d_{in}}} - q}$  and  $\ell_2(q) = (1 - (d_{in} + d_{out})q) \frac{\gamma^{\frac{1}{d_{in}} - 1}}{\gamma^{\frac{1}{d_{in}}} - q}$ . It is easy to verify that for  $k \geq 3$  and  $q \in (0, \frac{1}{d_{in} + d_{out}}]$ ,  $\ell_1(q)/\ell_2(q)$  is monotonically increasing/decreasing. Note when  $q = \frac{1}{\beta}$ ,  $\ell_1(q) = \ell_2(q) = \frac{\gamma^{\frac{1}{d_{in}} - 1}}{\beta \gamma^{\frac{1}{d_{in}}} - 1}$ . We consider two cases for  $q$ : (1) when  $q \in (0, \frac{1}{\beta}]$ , we have  $\max\{\ell_1(q), \ell_2(q)\} \geq \ell_2(\frac{1}{\beta})$  as  $\ell_2(q)$  is monotonically decreasing; (2) when  $q \in (\frac{1}{\beta}, \frac{1}{d_{in} + d_{out}}]$ , we have  $\max\{\ell_1(q), \ell_2(q)\} \geq \ell_1(\frac{1}{\beta})$  as  $\ell_1(q)$  is monotonically increasing. Thus we have the Inequality 4 as below.

**Inequality 4:**

$$\max\left\{q \frac{\gamma^{\frac{1}{d_{in}}} - 1}{\gamma^{\frac{1}{d_{in}}} - q}, (1 - (d_{in} + d_{out})q) \frac{\gamma^{\frac{1}{d_{in}}} - 1}{\gamma^{\frac{1}{d_{in}}} - q}\right\} \geq \frac{\gamma^{\frac{1}{d_{in}}} - 1}{\beta \gamma^{\frac{1}{d_{in}}} - 1}$$

Combining Inequality 1, Inequality 2 and Equation (1), we can have the first lower bound in Theorem 1:

$$\begin{aligned} f(\sigma - z) &\geq \max\{q \cdot f(\sigma_2), (1 - (d_{in} + d_{out})q)f(\sigma_2)\} \\ &\geq \max\{q, 1 - (d_{in} + d_{out})q\} \frac{1 - e^{-(1-1/k)}}{\alpha} g_\tau(\sigma^*(V, k, \tau)) \\ &\geq \frac{1 - e^{-(1-1/k)}}{\alpha \beta} g_\tau(\sigma^*(V, k, \tau)) \end{aligned}$$

Combining Inequality 1, Inequality 3 and Inequality 4, we can have the second lower bound in Theorem 1:

$$\begin{aligned} f(\sigma - z) &\geq \max\{q \cdot f(\sigma_2), (1 - (d_{in} + d_{out})q) \cdot f(\sigma_2)\} \\ &\geq \max\left\{q \cdot \frac{\gamma^{\frac{1}{d_{in}}} - 1}{\gamma^{\frac{1}{d_{in}}} - q}, \right. \\ &\quad \left. (1 - (d_{in} + d_{out})q) \frac{\gamma^{\frac{1}{d_{in}}} - 1}{\gamma^{\frac{1}{d_{in}}} - q}\right\} g_\tau(\sigma^*(V, k, \tau)) \end{aligned}$$

$$\geq \frac{\gamma^{\frac{1}{d_{in}}} - 1}{\beta \gamma^{\frac{1}{d_{in}}} - 1} g_\tau(\sigma^*(V, k, \tau))$$

<sup>3</sup>As  $x$  is monotone increasing and  $1 - bx$  is monotone decreasing, the function achieves the maximum value when  $x = 1 - bx$

Now we are done.

## Experiments

We compare the performance of our algorithms RoseNets to the non-robust version Sequence Greedy (**Sequence** for short) (Mitrovic et al. 2018), the existing submodular sequence baseline (**OMegA**) (Tschitschek, Singla, and Krause 2017), and a naive baseline (**Frequency**) which outputs the most popular items the user has not yet reviewed.

To evaluate the performance of the algorithms, we use three evaluation metrics in this paper. The first one is **Accuracy Score**, which simply counts the number of accurately recommended items. While this is a sensible measure, it does not explicitly consider the order of the sequence. Therefore, we also consider the **Sequence Score**, which is a measure based on the Kendall-Tau distance (Kendall 1938). This metric counts the number of ordered pairs that appear in both the predicted sequence and the true sequence. These two metrics are also used in (Mitrovic et al. 2019). The third metric is the **Utility Function Value** of the selected sequence.

We use a probabilistic coverage utility function as our Net-submodular function  $f$ . Mathematically,

$$f(\sigma) = h(E_1) = \sum_{j \in V} [1 - \prod_{(i,j) \in E_1} (1 - w_{ij})],$$

where  $E_1 \in E$  is the edges induced by sequence  $\sigma$ . And to simulate the worst case removal, we remove the first  $\tau$  elements to evaluate the robustness of the algorithms.

### Amazon Product Recommendation

Using the Amazon Video Games review dataset (Ni, Li, and McAuley 2019), we conduct experiments for the task of recommending products to users. In particular, given a specific user with the first 4 products she has purchased, we want to predict the next  $k$  products she will buy. We first build a graph  $G = (V, E)$ , where  $V$  is the set of all products and  $E$  is the set of edges between these products. The weight of each edge,  $w_{ij}$ , is defined to be the conditional probability of purchasing product  $j$  given that the user has previously purchased product  $i$ . We compute  $w_{ij}$  by taking the fraction of users that purchased  $j$  after having purchased  $i$  among all the users that purchased  $i$ . There are also self-loops with weight  $w_{ii}$  that represent the fraction of users that purchased product  $i$  among all the users. We focused on the products that have been purchased at least 50 times each, leaving us with a total of 9383 unique products. Also we select the users that have purchased at least 29 products, leaving us 909 users. We conduct recommendation task on these 909 users and take the average value of each evaluation metric.

Figure 3 shows the performance of the comparison algorithms using the accuracy score, sequence score and utility function value respectively. In Figure 3(a), 3(b), 3(c) and 3(d), we find that after the removal of  $\tau$  elements, the RoseNets outperforms all the comparisons. Such results demonstrate that the RoseNets algorithm is effective and robust in real applications since accuracy score and sequence

score are common evaluation metrics in practical. In Figure 3(e) and 3(f), the only difference is that the OMeGA algorithm is outperforming when  $\tau$  is small or  $k$  is large. The OMeGA algorithm aims to find a global optimal solution. It topologically resorts all the candidates after each element selection. It can return a solution with better utility function value when  $k$  is large and  $\tau$  is small, but runs much slower than RoseNets or Sequence. Also, it shows poor performance in accuracy and sequence score. Thus the RoseNets algorithm is more effective and robust in real applications.

In addition, we also show the case of RoseNets and Sequence with  $\tau = 0$ . We can see that in all the experimental results for the three metrics, the RoseNets outperforms Sequence, which is consistent to our expectation. On utility function value, the Sequence( $\tau = 0$ ) is better than the RoseNets( $\tau = 0$ ). However, on accuracy score and sequence score, the RoseNets( $\tau = 0$ ) is very close to the Sequence( $\tau = 0$ ), sometimes shows better performance. The former result is due to the effectiveness of greedy framework. The RoseNets algorithm indeed invokes Sequence for two trials independently, which intuitively cannot achieve comparable performance with one trial Sequence execution, due to the Net-submodularity. But the latter result shows that directly implementing greedy selection is not always outperforming. This is due to the intrinsic property of the greedy algorithm. Though  $1 - 1/e$  is almost the best approximation ratio, some heuristic algorithms may achieve better performance in specific cases. However, if we invoke more trials of Sequence algorithm, better robust experimental results and approximation ratio might be achieved but the utility value would become lower. This is because more trials of independent greedy selection would give high probability of triggering the diminishing return phenomenon. In real applications, this is a trade-off for designing robust algorithms that requires a balance between high efficiency on robustness and maximization of utility value.

### Wikipedia Link Prediction

Using the Wikispeedia dataset (West, Pineau, and Precup 2009), we consider users who are surfing through Wikipedia towards some target article. Given a sequence of articles the user has previously visited, we want to guide her to the page she is trying to reach. Since different pages have different valid links, the order of pages we visit is critical to this task. Formally, given the first 4 pages each user visited, we want to predict which page she is trying to reach by making a series of suggestions for which link to follow. In this case, we build the graph  $G = (V, E)$ , where  $V$  is the set of all pages and  $E$  is the set of existing links between pages. Similarly to the recommendation case, the weight  $w_{ij}$  of an edge  $(i, j) \in E$  is the probability of moving to page  $j$  given that the user is currently at page  $i$ , i.e., the fraction of moves from  $i$  to  $j$  among all the visit of  $i$ . In this case, we build no self-loops as we assume we can only move using links. Thus we cannot jump to random pages. We condense the dataset to include only articles and edges that appeared in a path, leaving us 4170 unique pages and 55147 edges. We run the algorithm on paths with length at least 29, which leaves

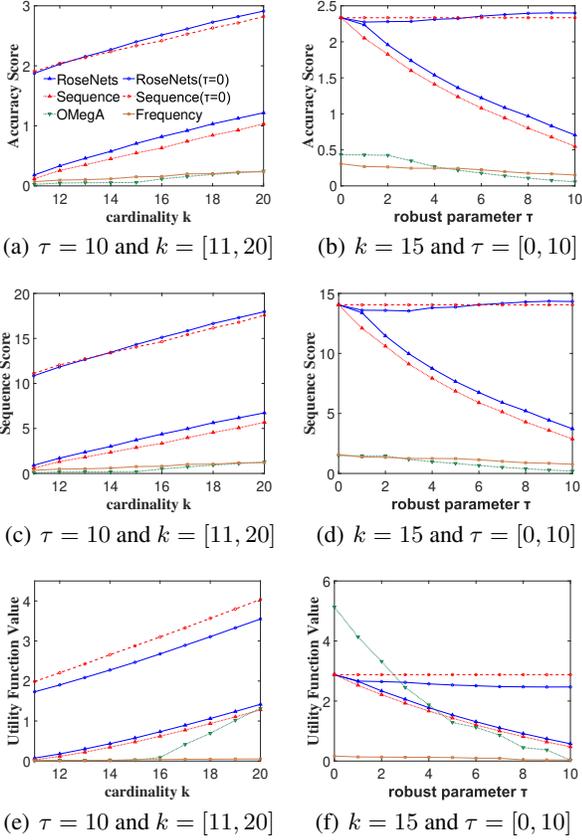


Figure 3: Recommendation Application

us 271 paths. We conduct link prediction task on these 271 paths and take the average value of each evaluation metric.

Figure 4 shows the performance of the comparison algorithms using the accuracy score, sequence score and utility function value respectively. In Figure 4, we find that after the removal of  $\tau$  elements, the RoseNets outperforms all the comparisons in all the cases. These results demonstrate that the RoseNets algorithm is effective and robust in real applications. The OMeGA algorithm does not show comparable performance to RoseNets algorithm any more. This is because that (1) the path need to be predicted is very long, and (2) the intersection part of different paths is not as large as the case in Amazon recommendation experiments. Thus the global algorithm OMeGA cannot exploit its advantages. We still can find in Figure 4(a), 4(c) and 4(e) that when  $k$  becomes larger, the performance of OMeGA algorithm increases faster. This in turn demonstrates that the RoseNets algorithm is more general, effective and robust, which does not assume specific real application scenario.

Another difference in the link prediction application is that, the RoseNets( $\tau = 0$ ) outperforms Sequence( $\tau = 0$ ) almost in all the cases on accuracy and sequence score. This again verifies that directly implementing greedy selection may sometimes far away from the optimal solution. As discussed in the end of the recommendation case, an interesting future direction is to explore the trade-off between high efficiency on robustness and the maximization of utility value

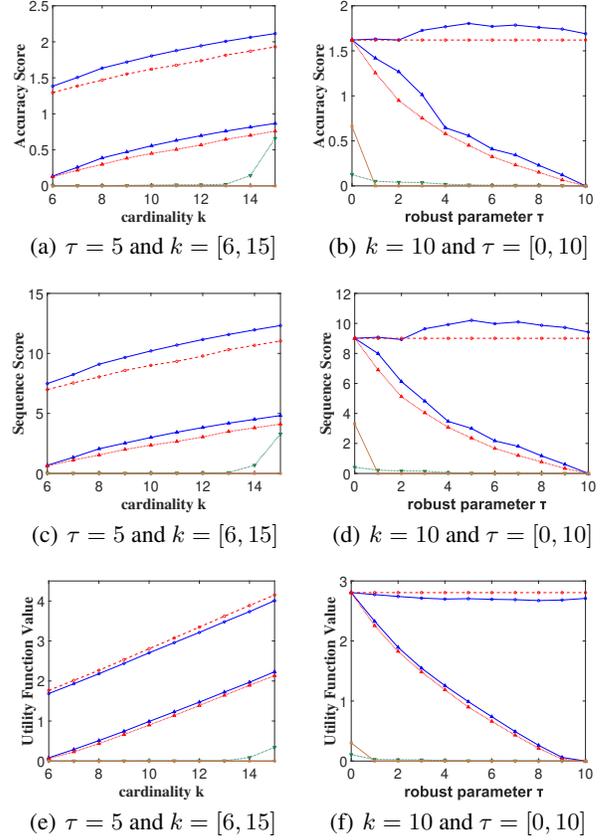


Figure 4: Link Prediction Application

by invoking more independent greedy selection trials.

## Conclusion

In this paper, we are the first to study the RoseNets problem, which combines the robust optimization and sequence networked submodular maximization. We design a robust algorithm with an approximation ratio that is bounded by the number of the removed elements and the network topology. Experiments on real applications of recommendation and link prediction demonstrate the effectiveness of the proposed algorithm. For future works, one direction is to develop robust algorithms that can achieve higher approximation ratio. An intuitive improvement is to invoke multiple trials of independent greedy selection. Another direction is to consider the robustness against the removal of edges. This is non-trivial since different removal operation would change the network topology and affect the approximation ratio.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No: 62102357, U1866602, 62106221, 62102382), and the Starry Night Science Fund of Zhejiang University Shanghai Institute for Advanced Study (Grant No: SN-ZJU-SIAS-001). It is also partially supported by the Zhejiang Provincial Key Research and Development Program of China (2021C01164).

## References

- Bach, F. 2019. Submodular functions: from discrete to continuous domains. *Mathematical Programming*, 175(1): 419–459.
- Bogunovic, I.; Mitrović, S.; Scarlett, J.; and Cevher, V. 2017. Robust submodular maximization: A non-uniform partitioning approach. In *International Conference on Machine Learning*, 508–516. PMLR.
- Calinescu, G.; Chekuri, C.; Pal, M.; and Vondrák, J. 2011. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6): 1740–1766.
- Chekuri, C.; Vondrák, J.; and Zenklus, R. 2014. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *SIAM Journal on Computing*, 43(6): 1831–1879.
- Das, A.; and Kempe, D. 2011. Submodular meets spectral: greedy algorithms for subset selection, sparse approximation and dictionary selection. In *Proceedings of the 28th International Conference on Machine Learning*, 1057–1064.
- Feige, U.; Mirrokni, V. S.; and Vondrak, J. 2007. Maximizing Non-Monotone Submodular Functions. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, 461–471.
- Gabillon, V.; Kveton, B.; Wen, Z.; Eriksson, B.; and Muthukrishnan, S. 2013. Adaptive submodular maximization in bandit setting. *Advances in Neural Information Processing Systems*, 26.
- Golovin, D.; and Krause, A. 2011. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42: 427–486.
- Kazemi, E.; Zadimoghaddam, M.; and Karbasi, A. 2018. Scalable deletion-robust submodular maximization: Data summarization with privacy and fairness constraints. In *International conference on machine learning*, 2544–2553. PMLR.
- Kempe, D.; Kleinberg, J.; and Tardos, É. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 137–146. ACM.
- Kendall, M. G. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2): 81–93.
- Khuller, S.; Moss, A.; and Naor, J. S. 1999. The budgeted maximum coverage problem. *Information processing letters*, 70(1): 39–45.
- Kirchhoff, K.; and Bilmes, J. 2014. Submodularity for data selection in machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 131–141.
- Krause, A. 2005. Near-optimal nonmyopic value of information in graphical models. In *Proc. of the 21st Annual Conf. on Uncertainty in Artificial Intelligence (UAI 2005)*, 324–331.
- Li, M.; Wang, T.; Zhang, H.; Zhang, S.; Zhao, Z.; Zhang, W.; Miao, J.; Pu, S.; and Wu, F. 2022. HERO: Hierarchical spatio-temporal reasoning with Contrastive Action Correspondence for End-to-End Video Object Grounding. In *ACM MM*.
- Lin, H.; and Bilmes, J. 2011. A class of submodular functions for document summarization. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, 510–520.
- Mirzasoleiman, B.; Karbasi, A.; and Krause, A. 2017. Deletion-robust submodular maximization: Data summarization with “the right to be forgotten”. In *International Conference on Machine Learning*, 2449–2458. PMLR.
- Mitrovic, M.; Feldman, M.; Krause, A.; and Karbasi, A. 2018. Submodularity on hypergraphs: From sets to sequences. In *International Conference on Artificial Intelligence and Statistics*, 1177–1184. PMLR.
- Mitrovic, M.; Kazemi, E.; Feldman, M.; Krause, A.; and Karbasi, A. 2019. Adaptive sequence submodularity. *Advances in Neural Information Processing Systems*, 32.
- Mitrovic, S.; Bogunovic, I.; Norouzi-Fard, A.; Tarnawski, J. M.; and Cevher, V. 2017. Streaming robust submodular maximization: A partitioned thresholding approach. *Advances in Neural Information Processing Systems*, 30.
- Nemhauser, G. L.; and Wolsey, L. A. 1978. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of operations research*, 3(3): 177–188.
- Ni, J.; Li, J.; and McAuley, J. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, 188–197.
- Orlin, J. B.; Schulz, A. S.; and Udewani, R. 2018. Robust monotone submodular function maximization. *Mathematical Programming*, 172(1): 505–537.
- Sallam, G.; Zheng, Z.; Wu, J.; and Ji, B. 2020. Robust sequence submodular maximization. *Advances in Neural Information Processing Systems*, 33: 15417–15426.
- Shi, Q.; Wang, C.; Ye, D.; Chen, J.; Feng, Y.; and Chen, C. 2019. Adaptive influence blocking: Minimizing the negative spread by observation-based policies. In *2019 IEEE 35th international conference on data engineering (ICDE)*, 1502–1513. IEEE.
- Shi, Q.; Wang, C.; Ye, D.; Chen, J.; Zhou, S.; Feng, Y.; Chen, C.; and Huang, Y. 2021. Profit maximization for competitive social advertising. *Theoretical Computer Science*, 868: 12–29.
- Streeter, M.; and Golovin, D. 2008. An online algorithm for maximizing submodular functions. *Advances in Neural Information Processing Systems*, 21.
- Tschiatschek, S.; Singla, A.; and Krause, A. 2017. Selecting sequences of items via submodular maximization. In *Thirty-First AAAI Conference on Artificial Intelligence*.

- Tzoumas, V.; Gatsis, K.; Jadbabaie, A.; and Pappas, G. J. 2017. Resilient monotone submodular function maximization. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 1362–1367. IEEE.
- Tzoumas, V.; Jadbabaie, A.; and Pappas, G. J. 2018. Resilient monotone sequential maximization. In *2018 IEEE Conference on Decision and Control (CDC)*, 7261–7268. IEEE.
- Tzoumas, V.; Jadbabaie, A.; and Pappas, G. J. 2020. Robust and adaptive sequential submodular optimization. *IEEE Transactions on Automatic Control*.
- Wang, C.; Shi, Q.; Xian, W.; Feng, Y.; and Chen, C. 2021. Efficient diversified influence maximization with adaptive policies. *Knowledge-Based Systems*, 213: 106692.
- West, R.; Pineau, J.; and Precup, D. 2009. Wikispeedia: An online game for inferring semantic distances between concepts. In *Twenty-First International Joint Conference on Artificial Intelligence*.
- Zhang, Z.; Chong, E. K.; Pezeshki, A.; and Moran, W. 2015. String submodular functions with curvature constraints. *IEEE Transactions on Automatic Control*, 61(3): 601–616.
- Zhang, Z.; Wang, Z.; Chong, E. K.; Pezeshki, A.; and Moran, W. 2013. Near optimality of greedy strategies for string submodular functions with forward and backward curvature constraints. In *52nd IEEE Conference on Decision and Control*, 5156–5161. IEEE.