Partition of Random Items: Tradeoff between Binning Utility and Meta Information Leakage

Farhang Bayat, Shuangqing Wei
Department of Electrical and Computer Engineering, Louisiana State University, USA

***Abstract*—In this paper, we propose a novel formulation to understand the tradeoff between binning utility and meta information leakage when we face the problems of partitioning random items. As an example, such problems could emerge when online users attempt to protect their browsing behavior patterns to certain extent by resorting to multiple proxy websites. Under the framework, we formulate a constrained optimization problem where the goal is to maximize binning utility while restraining a certain level of information leakage by properly dividing a set of M random items into N bins. By doing so, we formulate a new multi-agent multi-variate optimization problem which is NP-complicated. We then utilize the submodular nature of the problem to find sufficient conditions to (1) secure the existence of a solution to our problem; (2) lower the complexity of the problem at the cost of accuracy. To do so, we exploit the dual nature of set functions in multi-agent multi-variate problems, a novel addition to the field. After proving sufficient conditions to secure the existence of a solution in more general cases, we offer algorithms and complexity orders to solve a simplified version of the problem where $N = 2$ which helps signify the use of submodular properties.**

***Index Terms*—partition, information leakage, privacy, mutual information, submodularity**

I. Introduction

Today internet has become so intertwined with our everyday activities that it is impossible to imagine living without it. However, this dependency could be subject to exploitation by the eavesdroppers. Assuming every browse as a query, it could be argued that the search history of a user contains a series of queries specific to him which could point to his specific likes and dislikes. By following every user's history of browses vital information specific to each user could be developed. It then becomes important that each user attempt to add some level of protection to their browses to hide enough information about themselves.

One method is the use of proxy websites. Such websites offer the user a URL box where he can input any website he wishes to visit. The only difference is that in such websites, the address input is encoded into a series of characters which appear at the end of the URL of the original proxy website. These characters change through time by seconds meaning if the service provider records the URL opened through the proxy website and decides to open it to access specific content by inputting the URL, he will not go to the encoded web page. Also, such websites slow the connection. However, through this method, the user has the option of choosing multiple proxy websites and thus presumably cutting down on the utility loss. Thus if a utility function based upon connection speed -bandwidth- for the user is calculable, a privacy constrained problem between the user and an eavesdropper (for example a service provider) could be defined. The solution to such a problem could offer insights in regards to the tradeoff between proxy allocation utility and meta information leakage when we face the problems of partitioning a set of random items (i.e. websites that a user has chosen to visit following his own distributions) into a given number of bins (i.e. a given set of proxy servers each of which has its own utility function, as will be further detailed in Section II).

In our proposed framework as detailed in Section II, meta data information refers to the patterns about a sequence of items(for example the user's favorable websites) infer-able based upon a sequence of bins (e.g. proxy sites) observed by an eavesdropper. This assumption is an expansion of [1] and [2]. However, it should be noted that due to usage of proxy sites, an eavesdropper cannot directly observe the original input items, but rather bin indexes. Under our proposed novel framework, we introduce multi-submodularity and submodularity as two means of reducing the complexity level of such problems, namely, dividing M random items into N bins, under an upper-bound on leaked meta information.

This paper represents an expansion of our previous works in [3], [4] where in [3] a utility function was introduced in the form of the average stopping time for detection of an active subgraph using certain queries while in [4], we developed a concept of information leakage through a vast set of possible queries. In this paper, we shift our attention from detection of an active subgraph to seeking tradeoff between optimizing utility of partitioning a set of random items and restraining information leakage.

The concept of privacy has already been explored in many works such as [5], [6] where a general but non-mathematical explanation was offered. However; in our work, we go into further details as to what privacy represents in our framework and how it could be formulated into many settings. Later in this paper, we find it necessary to utilize the concept of multi-submodular set function problems and their solutions. This concept was widely discussed in [7] where they introduced a series of sufficient conditions on multi-submodular set functions by which the multi-submodular problem could be transformed into a submodular set function problem. Then, further discussions about the existence of a solution to the new problem were made. By doing so -and if a solution was proven to exist- , the complexity of the problem could be shown to be reduced from NP to polynomial. However, [7] did not offer any algorithmic solutions in such cases.

The novel contributions of this paper are as follows: (1)

978-1-5386-2321-3/18/$31.00 ©2018 IEEE

we introduce a new multi-agent multi-variant optimization problem with a privacy leakage constraint offering specific accommodations for online browsing which turns out to be NP-complicated; (2) we introduce the novel idea concerning dual nature of the multi-agent optimization problems and their corresponding implications in such problems; (3) we utilize multi-submodularity property to prove the existence of a transformation to submodular set problems given a series of sufficient conditions and then conclude the existence of a polynomial solution; by doing so, we simplify the complexity of the problem from NP to polynomial (4) we offer a new less restrictive set of sufficient conditions as well as a submodular set function optimization solution algorithm with its corresponding calculation complexities for the specific case of $N = 2$ (5) we offer a toy example by which the logic behind the use of submodualr set function is demonstrated.

The rest of the paper is organized as follows. In Section II we formulate the problem in terms of privacy and utility functions. We dissect what the goal and the constraints are. In Section III, we first introduce a revised version of the utility function and then find the sufficient conditions under which this function is equipped with multi-submodular property. Due to general limited knowledge about multi-submodular solutions and how they are developed, we then assume a specific case $N = 2$. In Section IV an algorithm for this specific case is developed which has a polynomial cost and an accuracy of 0.432. In Section V, a toy example is presented to show the application of submodular set function solutions and their suboptimality. Finally, in Section VI, we conclude the paper by reiterating what we have accomplished in this paper.

It is important to note that due to page limitations, we have left the proof of any theorems and lemmas in this paper to a more comprehensive technical report [8] available online. Through the rest of this paper, anytime a proof is required, we cite the technical report and offer insight as to how the proof is carried out.

II. System Model

First, we propose an abstract framework to formalize the goal of seeking partition of M items into N bins. More specifically, we aim to allocate each one of $1 \leq i \leq M$ possible items (queries) to one of N output bins. There could be at most N^M such partitions. It follows that any set allocation $A_l, 1 \leq l \leq N^M$ results in N sets $S_j^{(l)} \subseteq \{1, 2, ..., M\}, j = 1, 2, ..., N$. Each such set is defined as

$$S_j^{(l)} = \{i | \theta_{i,j} = 1\}$$
$$where \quad \theta_{ij}^{(l)} = \begin{cases} 1 & i \in S_j^{(l)} \\ 0 & i \notin S_j^{(l)} \end{cases} \tag{1}$$

We further assume $S_j^{(l)} \cap S_k^{(l)} = \emptyset, j \neq k$. Furthermore we have $\bigcup_{j=1}^{N} S_j^{(l)} = \{1, 2, ..., M\}$. Finally the size of each such set $S_j^{(l)}$ is defined as $L_j^{(l)}$.

A. Probabilistic Model

We assume at any time slot one and only one of the inputs is chosen with a certain probability. Thus if we use variable $X \in \{1, 2, ..., M\}$ as a representation of set of items, we could have $P(X = i) = P(\gamma_i = 1) = \pi_i, 1 \leq i \leq M$ as a representation of the probability of choosing item i from the set X where $\gamma_i \in \{0, 1\}$. It further follows that $\sum_{i=1}^{M} \gamma_i = 1$, stipulating that one and only one of M items is selected. These M items could represent a set of M web pages to be visited by a user at a particular time instant. The prior probability distribution of M items reflects the user's favoritism toward these web pages. .

Next, we introduce an observable random variable $Y \in \{1, 2, \cdots, N\}$, denoting the index of the bin (the proxy site) employed to carry one of the $M > N$ items. It follows that the probability of each bin's appearance given a set allocation scheme such as A_l will be equal to $P(Y = j|A_l) = \sum_{i=1}^{M} P(Y = j|A_l, X = i)P(X = i|A_l) = \sum_{i=1}^{M} P(Y = j|A_l, X = i)P(X = i)$ where we have dropped the second conditional probability due to the independence between X and A_l. Furthermore $P(Y = j|A_l, X = i) = \theta_{ij}^{(l)} \in \{0, , 1\}$. It thus follows that

$$P(Y = j|A_l) = \sum_{i=1}^{M} \theta_{ij}^{(l)} P(X = i) \rightarrow$$
$$P(Y = j|A_l) = \sum_{i \in S_j^{(l)}} \pi_i = \alpha_j^{(l)} \tag{2}$$

B. Revealed Information

By choosing to allocate M items to N bins where $N \leq M$, we have injected ambiguity and uncertainty into the output binning index sequence about the input item sequence over a successive n visits or channel uses. In other words, if we originally chose to transmit n of such items, our total set of possible sequences would be of form $\overrightarrow{\mathbf{X^n}} = [X_1 X_2 ... X_n]$ out of M^n possible outcomes. From an observer's perspective which can only have access to which one of N bins is deployed in each time slot, sequences in the form of $\overrightarrow{\mathbf{Y^n}} = [Y_1 Y_2 ... Y_n]$ has cardinality of at most $N^n < M^n$. Despite the amount of uncertainty added due to the many-to-one mapping between items and bins, the output sequence sill reveals certain amount of information regarding the patterns of sequences of M random items.

This observation could be further studied by indicating how our allocation system resembles a coding framework where we have an equivalent channel whose input variable is X and output Y, as show in Figure 1.

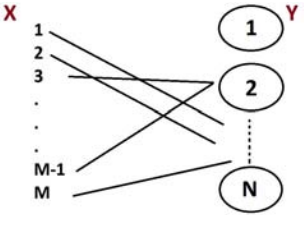

Figure 1. Coding Channel Representation of the Problem

Under such a framework, the equivalent channel output sequence $\overline{\mathbf{Y}}^n$ can help an eavesdropper classify the input sequence $\overline{\mathbf{X}}^n$ into a number of differential classes. As a result, information about the specific input item patterns is leaked to certain degree and can be measured using conditional mutual information $I(X; Y|A_l)$ between X

and Y, given a particular channel mapping (i.e. partition A_l relationship as illustrated in Figure 1.

Such conditional mutual information thus measures the maximum number of bits of meta information about item sequence per channel use. Therefore, we can have at most $2^{nI(X;Y|A_l)}$ sequences $\overline{\mathbf{X}}^n$ distinguishable by inferring based on $\overline{\mathbf{Y}}^n$. We thus adopt $I(X;Y|A_l)$ as the privacy metric conditioned on a particular partition mapping A_l.

It follows that due to the combinatorial nature of a set allocation problem there are a total of N^M possible methods to allocate these M items to the N sets. We can formulate the mutual information over a set allocation $A_l, 1 \leq l \leq N^M$ as:

$$\begin{gathered} I(X;Y|A_l) = H(X|A_l) - H(X|Y, A_l) = \\ H(Y|A_l) - H(Y|X, A_l) = H(Y|A_l) = \\ H(\alpha_1^{(l)}, \alpha_2^{(l)}, ..., \alpha_N^{(l)}) \end{gathered} \quad (3)$$

where we have used the notion of $H(a_1, a_2, ..., a_m) = -\sum_{v=1}^{m} a_v \log a_v$ and the fact that $H(Y|X, A_l) = 0$ because if both the input X and the channel scheme A_l are known, then output Y could simply be calculated.

C. Utility Function

Note that the main reason why we chose bin allocation was to reach a higher utility function. In this section we define a utility function to apply to the problem. If an allocation scheme A_l has resulted in $S_1^{(l)}, S_2^{(l)}, ..., S_N^{(l)}$, we assume each of the N bins offer a utility function of their own based upon the set they have been bestowed. Every bin j thus offers a utility represented by $f_j(S_j^{(l)})$. It is important to note that f_j represents a set function meaning it would change as different subsets of the universal set are chosen. An example for such a function would be if $f_j(S_j^{(l)}) = f_j(|S_j^{(l)}|)$ meaning the function changes as the number of members within the set $S_j^{(l)}$ changes.

It then follows that the average utility function will be in the form of $U_l = \sum_{j=1}^{N} f_j(S_j^{(l)}) P(Y = j|A_l) = \sum_{j=1}^{N} \alpha_j^{(l)} f_j(S_j^{(l)})$.

D. Problem Definition

Based on the previous observations we set a goal to find the set allocation A_l over which (a) U_l is maximized and (b) $I(X;Y|A_l) \leq I_{th}$ where I_{th} represents the maximal allowed revealed information.

We aim to gather both the utility and the constraint imposed in the form of one function we hope to maximize. Thus, a new maximization problem could be developed:

$$\max_{1 \leq l \leq N^M} \quad U_l + \lambda(I_{th} - H(\alpha_1^{(l)}, \alpha_2^{(l)}, ..., \alpha_N^{(l)})) \quad (4)$$

where $\lambda \geq 0$ represents a variable connecting the utility and the constraint to each other so as to allow comparison between them. We can further express this equation by opening it as:

$$\max_{1 \leq l \leq N^M} \quad \sum_{j=1}^{N} [\alpha_j^{(l)} f_j(S_j^{(l)}) + \lambda \alpha_j^{(l)} \log(\alpha_j^{(l)})] \quad (5)$$

We can now see that if we define $F_j^{(l)} = \alpha_j^{(l)} f_j(S_j^{(l)}) + \lambda \alpha_j^{(l)} \log(\alpha_j^{(l)})$, Equation (4) is simply a sum of functions defined over a series of sets. We refer to these as multivariate set functions seeing as how their values are based upon specific sets and variables introduced in each of these sets.

III. Multi-Submodular Set Functions as a Means of Solution

As mentioned previously, the problem formulated in Eq. (4) is NP-complicated (it is solved when a search over N^M possible set allocations is carried out and the best allocation is chosen). Still, we could opt to utilize the definition of multi-submodular set functions so as to reduce the complexity to that of polynomial at the cost of accuracy. In order to do so, we raise concerns about possible solutions. Next, we offer insight as to how we could deal with each case.

A. Imposing Multi-submodularity

In [7], it was shown that if we can prove multi-submodularity for functions such as those formulated in Eq. (4), then they could be modeled as simpler problems (submodular set functions). We thus, aim to find the sufficient conditions for such occurrence. To do so, we first offer a review of multi-submodularity.

As mentioned in [7], if we define $\mathbb{M} = \{1, 2, ..., M\}$, then a multivariate function $F : (2^{\mathbb{M}})^N \to \mathbb{R}^+$ is multi-submodular if for all pairs of tuples $(S_1, ..., S_N)$ and $(T_1, ...T_N) \in (2^{\mathbb{M}})^N$ we will have:

$$\begin{aligned} F(S_1, ..., S_N) + F(T_1, ..., T_N) \geq F(S_1 \cup T_1, ..., S_N \cup T_N) \\ + F(S_1 \cap T_1, ..., S_N \cap T_N) \end{aligned} \quad (6)$$

Since in our formulation functions are separately defined on different sets, the condition in Eq. (6) is simplified to the sufficient condition of submodularity of $F_j^{(l)}$ for all sets S_j for Equation (4). We now need to find the sufficient condition for submodularity of $F_j^{(l)}$ when defined over a set S_j.

B. Imposing Separate Submodularities

For an easier mathematical representation of the following derivation we denote $F(S_j) = F_j^{(l)}$. Furthermore, we denote $f_j(S_j^{(l)}) = f(S_j^{(l)})$. Both these denotations allude to the fact that once a set allocation A_l is chosen, its index could be dropped.

In the next step, we opt to use diminishing return property as the means of making certain each of these functions are submodular. Following is a definition of diminishing returns for submodular functions, after which we derive the sufficient conditions for the case discussed in Eq. (4).

Diminishing Property Return dictates that if we define $\mathbb{S}$ as the universal set, a set function $F : 2^{\mathbb{S}} \to \mathbb{R}^+$ is submodular if, for all $A, B \subseteq \mathbb{S}$ with $A \subseteq B$ and for each $x \in \mathbb{S} - B$ we have [9]:

$$F(A \cup \{x\}) - F(A) \geq F(B \cup \{x\}) - F(B) \quad (7)$$

Now we attempt to expand Eq. (7) for each $F(S_j)$. However, to properly do so, we first need to account for the behavior of this function.

We have defined $F(S_j) = \alpha_j f(S_j) + \lambda \alpha_j \log(\alpha_j)$ where it seems that the function has a singular relationship with set S_j. However; there is a secondary relationship the function shares with the set $S_j^{\mathbf{C}} = S - S_j$ where $S = \mathbb{S}$ represents the universal set. This relationship could be modeled as $F(S_j^{\mathbf{C}}) = (1-\beta_j)f(S - S_j^{\mathbf{C}}) + \lambda(1-\beta_j)\log(1-\beta_j)$ where we have used the fact that $\beta_j = 1 - \alpha_j$ seeing as how we define

$$\beta_j^{(l)} = \sum_{i \in \{S - S_j^{(l)}\}} \pi_i \qquad (8)$$

Thus, for any set S_j, we must find the sufficient conditions for the existence of diminishing property for both functions $F_1(S_j) = \alpha_j f(S_j) + \lambda\alpha_j \log(\alpha_j)$ and $F_2(S_j) = (1-\alpha_j)f(S - S_j) + \lambda(1-\alpha_j)\log(1-\alpha_j)$. To do so, we will evaluate their necessary conditions and then find their intersection as the final conditions (assuming they do not negate one another).

Note: For any further references, we first need to address a series of variable and function definitions which are going to play a vital role in the rest of this paper:

Definitions

1. Any variable represented with a capital Letter represents a set.
2. Any variable represented with a small letter represents an element.
3. $A - B$ represents a set containing all elements of set A which do not appear in set B.
4. α_x represents the probability of item x and α_A represents the sum of probabilities of items mapped into a set A.
5. α_{BA} represents the difference in the sum of probabilities of items mapped into the sets B and A which could be further shown as $\alpha_{BA} = \alpha_B - \alpha_A$.
6. $g(C, D)$ represents the 1^{st} order difference of a set function $f(C)$ from $f(C - D)$ where $D \subseteq C$ which could be formulated as $g(C) = f(C) - f(C - D)$.
7. $q(C, C_1, D, D_1)$ represents the 2^{nd} order difference of a set function $f(C)$ where $C_1 \subseteq C$ and $D_1 \subseteq D$ which could be formulated as $q(C, C_1, D, D_1) = g(C, D) - g(C_1, D_1)$.
8. We assume the probability of items is sorted in a decreasing manner such as $\pi_1 \geq \pi_2 \geq ... \geq \pi_M$.

Theorem III.1. *The set functions $F_1(S_j)$ and $F_2(S_j)$ and as a result $F(S_j)$ are submodular if*

(1) $g(S_j, S_w) \leq 0$
(2) $q(S_j, S_u, S_w, S_y) \leq 0$
(3) $|g(S_j, S_w)| \geq \lambda \log(\frac{1}{\pi_M})$

for all possible sets $S_w \subseteq S_j \subseteq S$ and $S_u \subseteq S_j$ and $S_y \subseteq S_w$ where S is the universal set.

The proof for this theorem is presented in the Appendix section under Theorem III-1 of [8]. In the proof, a series of sufficient conditions for either $F_1(S_j)$ and $F_2(S_j)$ are evaluated separately. This is done because although their conditions turn out to be the same, their derivations are vastly different as require separate discussions. It then follows that since both functions require the same set of sufficient conditions, the function $F(S_j)$ which represents either of them being chosen, also follows the same set of sufficient conditions. In the proof, we rewrite inequality (7) for set function $F_1(S_j)$, start factorizing α_A, α_{BA} separately and α_x and 1 together and impose sufficient conditions so that each of their coefficients is always positive.

Unfortunately, [7] does not provide us with an algorithm to remodel our multi-submodular problem in a submodular problem, they simply prove that this could be done. Thus, in order to expand upon the idea of polynomial complexity of solution algorithms we opt to assume $N = 2$ and offer the reader the algorithm to deal with such a specific case. We then calculate the complexity imposed by the algorithm to further stress the benefits of using such an idea in spite of accepting error.

C. Specific Case of $N = 2$

As mentioned previously, in order to show the applicability of submodular functions we choose to reiterate the utility function dictated in Eq. (4) for when $N = 2$:

$$\max_{1 \leq l \leq 2^M} \quad \alpha_1^{(l)} f(S_1^{(l)}) + (1 - \alpha_1^{(l)}) f(S - S_1^{(l)})$$
$$+\lambda(I_{th} + \alpha_1^{(l)} \log(\alpha_1^{(l)}) + (1 - \alpha_1^{(l)}) \log(1 - \alpha_1^{(l)})) = T(S_1) \qquad (9)$$

As can be seen, the problem is still exponentially complex seeing as how we need to search over 2^M possible solutions to find the optimal. Thus, once again we aim to impose multi-submodularity (in this case simplified to submodularity) on the new utility function. For the utility function above the same results derived for a general N could be used as a set of sufficient conditions. However, taking into account the joint relationship between the 2 sets and writing the same Inequality (7) for Equality (9) we are able to find a less restrictive set of sufficient conditions for the submodularity of this utility function as indicated below:

Lemma III.2. *When $N = 2$, the function in Eq. (9) is submodular if*

(1) $g(S_1^{(l)}, S_w^{(l)}) \leq 0$
(2) $2|g(S_1^{(l)}, S_w^{(l)})| \geq \lambda \log(K), K < (\frac{1}{\pi_M})^2$
(3) $q(S_j^{(l)}, S_u^{(l)}, S_w^{(l)}, S_y^{(l)}) \leq 0$

for all possible sets $S_w^{(l)} \subseteq S_j^{(l)} \subseteq S$ and $S_u^{(l)} \subseteq S_j^{(l)}$ and $S_y^{(l)} \subseteq S_w^{(l)}$ where S is the universal set.

The proof for this lemma is presented in Appendix section under Lemma III-2 of [8]. In the proof, we rewrite inequality (7) for set function described in Eq. (9), start factorizing α_A, α_{BA} separately and α_x and 1 together and impose sufficient conditions so that each of their coefficients is always positive. In the following section, we will offer a method of solving a problem as introduced in Eq. (9).

IV. Submodular Solution

Starting by [10] there has been monumental work done over greedy algorithms with constraints (as long as they introduce down-monotone solvable polytopes) with a solution proximity of $\frac{1}{e}$. Later [11] introduced a solution proximity of 0.372. Finally [12] proved that this proximity could be increased to 0.432 in maximization problems which is quite close to the no-constraint solution of a symmetric problem.

It is important to note that while all these papers dealt with the issue of constraints, they assumed much more complex constraints than we are dealing with in this paper. Our only constraint is that $\alpha_1^{(l)} \leq \alpha_s$ where we assume $h(\alpha_s) = I_{th}, \alpha_s \leq 0.5$ which is obviously a down-monotone constraint. Thus, we can simply use the results from their work to create our own algorithm to find the submodular function solution to our problem. We present:

```
Algorithm 1: Submodular Function Solution to the problem as described in Eq.(4)
    1. Let $S_1 = argmax_{e \in X=\{1,...,M\}} T[S_1 = \{e\}]$ while $|0.5 - \alpha_1^{(l,1)}| \geq |0.5 - \alpha_s|$.
    2. If there is an element $e \in X \setminus S_1$ such that $T[S_1 + \{e\}] \geq T[S_1]$ and $|0.5 - \alpha_1^{(l)} - \alpha_e| \geq |0.5 - \alpha_s|$, let $S_1 = S_1 + \{e\}$.
    3. If there is an element $e \in S_1$ such that $T[S_1 \setminus \{e\}] \geq T[S_1]$ and $|0.5 - \alpha_1^{(l)} + \alpha_e| \geq |0.5 - \alpha_s|$, let $S_1 = S_1 - \{e\}$. Go to Step 2.
    4. Return maximum of $T[S_1]$ and $T[X \setminus S_1]$.
```

where we know that at the very last step $T[S_1] = T[X \backslash S_1]$. Now we opt to calculate the complexities of this method. Steps 2 and 3 could repeat $(M-1)+(M-2)+...+1 = \frac{M(M+1)}{2}$ times each while every item could be removed and thus replaced a total of $2M$ times. Thus the total complexity of steps 2 and 3 is equal to $M^2(M+1) = O(M^3)$. The complexity of step 1 is also equal to M. Thus the total complexity of the solution is equal to $O(M^3)$.

This polynomial solution simply makes certain the maximal function obtained is at least 0.432 times the optimal objective function. This range of error occurs because in this method, we are removing and adding members from and to the set S_1 one by one. Thus, at each decision point we are making one locally optimal decision. However, it is widely known that a locally greedy method is not necessarily globally optimal [13].

V. Numerical Comparison

In this section we offer two problems where we are hoping to use the results gathered in Lemma III.2. to (1) find a proper utility function given the specifics of each case (2) follow Algorithm 1 and compare its results with that of an exhaustive search to compare the two methods in terms of complexity and exactness of the solution. In both examples, we assume $M = 4, N = 2, \lambda = 0.25, I_{th} = 0.4$ and that $f(S) = f(|S|)$ where S and $|S|$ represent any set and its cardinality respectively. The only difference between the two examples would then lay in their item probability distributions. We assume that $\pi_1 = \pi_2 = \pi_3 = \pi_4 = 0.25$ in the first scenario and that $\pi_1 = 0.5, \pi_2 = 0.25, \pi_3 = \pi_4 = 0.125$ in the second scenario. The set function $f_1(|S|) = -|S|^2 - 2|S| + 25$ is then an example of the set functions that satisfy the submodularity conditions for both scenarios. We then need to run Algorithm 1 for the same set function over the two sets of probability distribution.

The results are gathered in Table I. Each cell represents the maximum overall utility achieved in either case by each method where it is obvious that the probability distribution plays a major role on the exactness of the solution compared to the utility function -which is the same in both cases. Also of interest is the negligible loss of utility at $1 - \frac{16.8614}{16.8972} = 0.0021$ for a desirable cost reduction of NP down to polynomial.

Table I
Solution Exactness Comparison

scenario	exhaustive search	submodularsolution
1	16.85	16.85
2	16.8972	16.8614

VI. Conclusions

In this paper, we introduced and formulated a problem widely regarded in online browses. To do so, we revisited the concept of privacy leakage discussed in both other and our own previous publications. We further introduced a utility function based upon the user's utilization of the network. We showcased how the problem formulation results in a multi-agent multi-variate problem which is NP-complicated. We then introduced the concept of submodularity and multi-submodularity which help reduce the complexity of such problems to that of a polynomial at the cost of some accuracy. We derived a series of sufficient conditions which would guarantee the existence of a solution. To do so, we introduced a novel observation of the behavior of such multi-agent multi-variate set functions which we called duality. Once the existence of such solutions was guaranteed, we introduced algorithms that could help us when $N = 2$ (but the original complexity is still combinatorial) and showcased how they help reduce the complexities.

References

[1] Y. Chen, C. Suh, and A. J. Goldsmith, "Information recovery from pairwise measurements," *IEEE Transactions on Information Theory*, vol. 62, no. 10, pp. 5881–5905, Oct 2016.

[2] E. Zheleva and L. Getoor, *Preserving the Privacy of Sensitive Relationships in Graph Data*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 153–171.

[3] F. Bayat and S. Wei, "Sequential detection of disjoint subgraphs over boolean mac channels: A probabilistic approach," in *2016 IEEE Globecom Workshops (GC Wkshps)*, Dec 2016, pp. 1–6.

[4] ——, "Non-adaptive sequential detection of active edge-wise disjoint subgraphs under privacy constraints," in *IEEE Transactions on Information Forensics Security*, To appear April 2018.

[5] F. Laforet, E. Buchmann, and K. B?hm, "Individual privacy constraints on time-series data," *Information Systems*, vol. 54, pp. 74 – 91, 2015.

[6] J. H. Ziegeldorf, O. G. Morchon, and K. Wehrle, "Privacy in the internet of things: threats and challenges," *Security and Communication Networks*, vol. 7, no. 12, pp. 2728–2742, 2014.

[7] R. Santiago and F. B. Shepherd, "Multi-agent and multivariate submodular optimization," *CoRR*, vol. abs/1612.05222, 2016.

[8] F. Bayat and S. Wei, "Partition of random items: Tradeoff between binning utility and meta information leakage." [Online]. Available: https://archive.org/details/ICTTechnicalReport

[9] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver, *Combinatorial Optimization*. New York, NY, USA: John Wiley & Sons, Inc., 1998.

[10] M. Feldman, J. Naor, and R. Schwartz, "A unified continuous greedy algorithm for submodular maximization." in *FOCS*, R. Ostrovsky, Ed. IEEE Computer Society, 2011, pp. 570–579.

[11] A. Ene and H. L. Nguyen, "Constrained submodular maximization: Beyond 1/e," in *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, vol. 00, Oct. 2016, pp. 248–257.

[12] M. Feldman, "Maximizing symmetric submodular functions," *CoRR*, vol. abs/1409.5900, 2014.

[13] M. Resende and C. Ribeiro, "Greedy randomized adaptive search procedures," in *Handbook of Metaheuristics*, F. Glover and G. Kochenberger, Eds. Kluwer Academic Publishers, 2003, pp. 219–249.