Information Privacy Tradeoff

Cole Becker, advised by Amir Ali Ahmadi

ORF 375 Report

1 Introduction

An important public policy issue that has arisen in recent years is the extent to which online advertisers should be allowed to mine and sell consumers' personal information to improve message targeting and sales. Whether it be social network platforms like Instagram and Facebook collecting personal information, or websites tracking "cookies," surveillance advertising has intrusively commodified our personal information for the purpose of private profit. The question of how to balance personal privacy protection and information gain is not well regulated, studied, or formulated. What is a "reasonable amount" of information a company should be able to extract about a consumer, and what kinds of personal data should consumers have a justifiable expectation of keeping private? Though often debated, clear metrics and standards don't yet exist. We introduce one possibility for defining and regulating privacy in a data-purchasing transaction, in the form of a decision problem we call Information Privacy Tradeoff Problem (IPT). Simply put, IPT quantifies privacy (or lackthereof) of a data purchase in terms of the degree to which revealed user data can be used to uniquely identify the user from a dataset with other users. IPT then seeks to balance the amount of information that is gained by a data purchaser with the amount of privacy the user experiences. In this work we introduce IPT, analyze some of its properties including computational complexity and solvability, and provide various problem bounds.

2 Related Works

2.1 Differential Privacy

Differential Privacy [4][6][12][7] is a relatively young field of privacy-preserving data analysis which broadly seeks to learn statistical information about a dataset without compromising the privacy of individuals. In particular, Differential Privacy is interested in quantifying (and reducing) how much the addition or removal of a user in a database will alter distributional information for the database, the assumption being is that distributional information can be shared. While related, our problem IPT takes a slightly different approach of seeking to quantify privacy of information sharing transaction for a single user in the database. For more detailed information on differential privacy, see any of the previously referenced survey papers.

2.2 The Knapsack Problem

The Knapsack Problem (KNAP) [8] [11] is a famous decision problem in combinatorial optimization, which seeks to determine the most valuable items to include in a knapsack from a given subset, constrained by a fixed capacity of the knapsack. While a different problem to IPT, various complexity and algorithmic results from KNAP can be leveraged to find similar results in IPT.

2.3 Submodular Maximization

Consider a function $f : 2^E \to \mathbb{R}$ mapping from the power set of some ground set S to $\mathbb{R}$. We say that f is submodular if $f(S + e) - f(S) \geq f(T + e) - f(T)$ for any fixed subsets $S \subseteq T \subseteq E$. Submodular maximization [9][10] problems occur when seek a maximizing subset $A \subseteq E$ for a submodular function f. While not yet well understood, there may be some relevant results from the submodular maximization literature to improve or inform IPT algorithms.

2.4 L0 Norm formulations

The formulation we discover for IPT contains the L0 pseudonorm, which counts the number of non-zero entries in a vector. While the L0 norm is not a true norm, and thus creates a non-convex inequality constraint, various works [2] [5], exist which produce closely related convex, or integer constraint reformulation of the L0 norm. Investigating these results could prove useful for future work with IPT.

3 The Problem

We consider the following decision problem, we denote IPT

Input: Consider a dataset $\mathcal{D} = \{d_i\}_{i \leq m} \subseteq \mathbb{R}^n$ where $d_i \in \{0,1\}^n$, and denote $D \in \mathbb{R}^{m \times n}$ as the matrix containing this data. We can imagine for example that m is the number of people in the dataset, and $d_i \in \mathbb{R}^n$ represents a binary feature vector of information on n features for the ith person. In addition, you are given a non-negative weight vector $w \in \mathbb{R}^n$ representing the relative importance of each the n features, some number $r \in [0,1]$ and an index $k \in \{1, ..., m\}$. Finally we also take in a non-negative number $l \in \mathbb{R}$.

Question: Is there a way to select a subset of features of d_k, indexed by $x \in \{0,1\}^n$ such that $w^T x \geq l$ and the subset features of d_k are shared by at least rm people in the dataset (d_k included). **Note**: Throughout the entirety of this paper, we will assume $k = 0$ unless otherwise specified, and we will refer to d_k as Person of Interest (poi).

3.1 Motivation

Suppose you are a company holding such a dataset $\mathcal{D}$ containing various boolean features of a large quantity of individuals. Some advertising company Ad.co comes to you, and wants to purchase the data of some individual k to run some targeted advertising of their own. In order to convey which features they are most interested in knowing, or which features are most useful to their advertising campaign, they supply you with a weight vector w which represents how much Ad.co values knowing each feature of individual k in your dataset. However, as your company must comply with privacy regulations, you are not allowed to share too much information about individual k which would distinguish them too much. Specifically, the features you reveal to Ad.co about individual k must also be shared by $100 * r\%$ of the members of your dataset. Your task is to maximize the useful information you can share with Ad.co while satisfying the imposed privacy constraints.

3.2 Formulation of IPT

Taking all the same inputs from the above problem, let us first define a helper matrix $K \in \mathbb{R}^{m \times n}$ as

$$K_{ij} = \begin{cases} 0 & \text{if } D_{ij} = D_{kj} = (d_k)_j \\ 1 & \text{otherwise} \end{cases}$$

We note that IPT can be reduced to the following non-convex optimization problem

$$\text{IPT} = \begin{array}{ll} \underset{x}{\text{maximize}} & w^T x \\ \text{subject to} & ||Kx||_0 \leq C \\ & x_i \in \{0, 1\} \end{array} \tag{IPT}$$

where here for ease of notation we define $C := \lceil m(1-r) \rceil$ and the answer to the decision question is given by whether IPT $\geq l$. To see why the constraint in this problem is equivalent to the requirement that the subset x of d_k's features is shared by at least rm people, note that

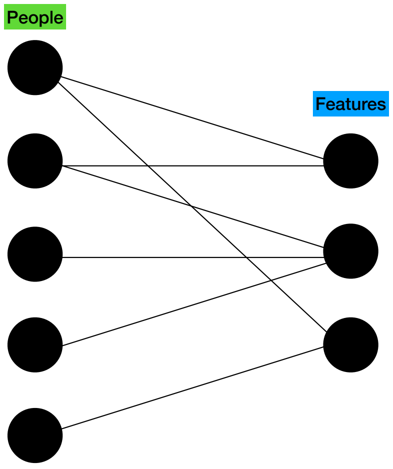
$$(Kx)_i = \begin{cases} 0 & \text{if } d_i \text{ shares the same } x \text{ features with } d_k \\ > 0 & \text{otherwise} \end{cases}$$

so the number of 0-elements in Kx must be greater than mr or equivalently, the number of non-zero elements must be less than $m(1-r)$. For ease of notation, we will use $C = \lceil m(1-r) \rceil$ for the rest of the paper.

3.3 Graph Interpretation

Another way to interpret this problem is via a graph optimization problem. Given a K matrix, we can define a respective bipartite graph $G(P, F, E)$ with vertex subsets P, F representing the people (excluding poi) in our dataset, and the features we know of. Two vertices p and f share an edge if person p is unique from poi in feature f. An example K matrix and bipartie graph is pictured below

$$K = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$



Given this Interpretation, we can define the following equivalent optimization problem:

$$\begin{aligned} \text{IPT} = \quad & \underset{X \subseteq V(F)}{\text{maximize}} \quad w^T X \\ & \text{subject to} \quad |N(X)| \leq C \end{aligned} \tag{IPT}$$

where $N(X)$ represents the vertex subset of all the neighbors of X. Here we note that for each $x \in X$, $N(x)$ represents all the individuals with feature x distinct, so we have that $|N(X)|$ really is equivalent of $||KX||_0$ from the original formulation.

3.4 Related Problem

A closely related problem is the following problem is Generalized Information Privacy Tradeoff Problem (GIPT)

$$\text{GIPT} = \text{argmin}_k \ \text{IPT}_k$$

where IPT_k represents the information privacy tradeoff of the kth individual in the dataset. We can think of the minimum over all such individauls as the most distinguishable/unique, or least anonymous individual. This is the individual for whom you can reveal the least information before they begin to be distinguishable from $r\%$ of the dataset. In a closely related problem, we have

$$\text{argmax}_k \ \text{IPT}_k$$

which can be thought of as the most anonymous and least unique individual in the dataset.

20 Questions A simple way to think of this problem is as a game. Given a dataset of possible things to guess in a game of twenty questions (or people to learn information from), you would like to choose the least distinguishable individual to maximize the amount of guesses. **However**, it is important to note that $\text{argmax}_k \ \text{IPT}_k$ isn't exactly identical to 20 questions. We might ask ourself what is the goal of twenty questions? The goal as a guesser is to reveal the features that on average leave the fewest members remaining. Here we are interested in privacy, so it would be more of a word producing game. We would want to produce a word such that on average, it is very anonymous. In other words, no matter what features you reveal, it doesn't narrow down your search very much. argmax_k IPT_k, means that you can reveal lots of features without narrowing the search down much. However think of the worst case scenario, in which there is one feature that such an individual has that she shares with nobody. This would make them distinguishable in a way, even if they were very anonymous with respect to their other features. So perhaps it isn't entirely obvious that argmax_k IPT_k is the best metric for a 20 questions guessing game.

4 Complexity of IPT

Theorem 4.1. *IPT is NP-complete*

Proof. It is sufficient to show that IPT $\in$ NP and that KNAP $\longrightarrow$ IPT

1. IPT $\in$ NP: Given a certificate solution x^* to IPT, it is easy to check that the constraints are satisfied, and that $w^T x^* \geq l$

2. KNAP $\longrightarrow$ IPT: Consider the classic KNAP:
Input: *$w \in \mathbb{R}^n$ a weight vector of n items, $p \in \mathbb{R}^n$ a price vector for the items, $W \in \mathbb{R}$ a weight capacity, and $P \in \mathbb{R}$*
Question: *Is there a set of items of combined weight less than W but with combined price greater than P?*

   Using these inputs we will construct an instance of IPT (Note we will assume for convenience that $W \in [0, 1^T w]$. If not we can define a new W that is respective endpoint). To begin, set $m = 1 + 1^T w$ and construct the following data matrix $D \in \mathbb{R}^{m \times n}$ by the following procedure:

   (a) Set the first row of D to a row of all 0s

   (b) For each weight w_i, $i \in \{1, ..., n\}$ add w_i copies of the one hot encoded row vector e_i. As an example:

$$w = [3, 1, 2, 1] \implies D = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

   This is the data matrix D we will consider. Note that the auxiliary matrix K used in IPT is identical to D via the way we constructed it.

In addition, set $r = 1 - W/m$ (whereby assumption on W we have $r \in [0,1]$), $l = P$, $k = 1$ the first row, and $\bar{w} = p$, the information weighting vector. We now have an instance of IPT. In terms of the KNAP variables, the equivalent IPT problem looks like the following:

$$\text{IPT(KNAP)} = \begin{array}{ll} \underset{x}{\text{maximize}} & p^T x \\ \text{subject to} & ||Dx||_0 \leq W \\ & x_i \in \{0,1\} \end{array} \tag{1}$$

Notice the problem construction requires a polynomial amount of operations, as the size of $D = n * (1 + 1^T w)$, and everything else requires an affine amount of operations. We would now like to show the following equivalence between the problems.

$$\text{KNAP} \geq P \iff \text{IPT(KNAP)} \geq P$$

($\Longrightarrow$): Consider a solution $\hat{x} \in \{0,1\}^n$ to KNAP satisfying the weight constraints and with $p^T x \geq P$. This same solution $\hat{x}$ will be feasible to IPT(KNAP). To see why, just note that $||Dx||_0 = w^T x$ is equivalent to taking the sum of the weights, because the construction of D and x mean that $||Dx||_0 = 1^T Dx$ and $1^T D = w$ by construction.
($\Longleftarrow$): The backwards argument is similar, but we need to build back the weight vector w from the matrix D. To do so we again note that $w = 1^T D$, and then the problem is identical to knapsack.

■

5 Bounds and Relaxations for IPT

5.1 Mixed Integer Program (MIP) Bounds

As written in its L-0 form, IPT is not even a mixed-integer program. Here we investigate affine relaxations of the L-0 norm and possible interpretations.

5.1.1 MIP Upper Bound

We write an IP relaxation of IPT as the following problem

$$\text{IPT}^{\text{IP}} = \begin{array}{ll} \underset{x \in \mathbb{R}^n}{\text{maximize}} & w^T x \\ \text{subject to} & \mathbf{1}^T K x \leq m(1-r)\mathbf{1}^T x \\ & x_i \in \{0,1\} \end{array} \tag{IPT$^{\text{IP}}$}$$

Where we have $\text{IPT} \leq \text{IPT}^{\text{IP}}$. To see why we would like to show

$$x \text{ feasible to IPT} \implies x \text{ feasible to IPT}^{\text{IP}}.$$

Consider x^* feasible to IPT. If $x^* = 0$ then x^* is clearly feasible to IPT^{IP} as both constraints are trivially satisfied. Now suppose $x^* \neq 0$, we want to show that

$$\frac{\mathbf{1}^T K x^*}{\mathbf{1}^T x^*} \leq C,$$

or equivalently, that $\frac{\mathbf{1}^T K x^*}{\mathbf{1}^T x^*} \leq ||Kx^*||_0$. The key is to notice is that $(Kx^*)_i \leq \mathbf{1}^T x^*$ for all $i \leq m$ due to the binary nature of the entries of K and x^*, which means that

$$\left(\frac{\mathbf{1}^T K x^*}{\mathbf{1}^T x^*}\right)_i = \begin{cases} 0 & \text{if } (Kx^*)_i = 0 \\ \leq 1 & \text{otherwise} \end{cases}$$

from the above statement we have

$$\frac{\mathbf{1}^T K x^*}{\mathbf{1}^T x^*} \leq ||Kx^*||_0$$

so x^* is feasible to IPT^{IP}, and $\text{IPT} \leq \text{IPT}^{\text{IP}}$.

5.1.2 MIP lower bound

Next consider the following linear integer program

$$\begin{aligned} \text{IPT}_{\text{IP}} \quad = \underset{x}{\text{maximize}} \quad & w^T x \\ \text{subject to} \quad & \mathbf{1}^T K x \leq C \\ & x_i \in \{0,1\} \end{aligned} \qquad (\text{IPT}_{\text{IP}})$$

To see why $\text{IPT}_{\text{IP}} \leq \text{IPT}$, we would like to show that

$$x \text{ feasible to } \text{IPT}_{\text{IP}} \implies x \text{ feasible to IPT}.$$

Fix x^* feasible to IPT_{IP}. Due to binary nature of K and x^*, we know Kx^* is a non-negative integer vector. This means $\mathbf{1}^T K x^* = ||Kx^*||_1$ and so we have

$$||Kx^*||_0 \leq ||Kx^*||_1 = \mathbf{1}^T K x^* \leq C$$

so x^* is feasible to IPT, and $\text{IPT}_{\text{IP}} \leq \text{IPT}$

5.1.3 Relation to KNAP

Succinctly, we have shown that $\text{IPT}_{\text{IP}} \leq \text{IPT} \leq \text{IPT}^{\text{IP}}$ due to the fact that for non-zero x, we have

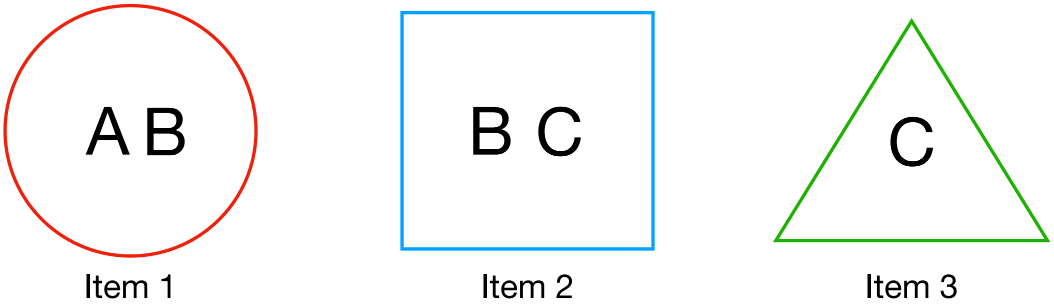
$$\frac{\mathbf{1}^T K x}{\mathbf{1}^T x} \leq ||Kx||_0 \leq \mathbf{1}^T K x$$

However if we define $\alpha := \mathbf{1}^T K$, then we can interpret IPT_{IP} as a knapsack problem, where each item in the knapsack corresponds to a feature in our dataset. In this way, we can interpret the ith item α_i as a collection of people with feature i distinct from our choice user. In other words, the weight of each item α_i is the number of people with feature i distinct. This means that for a chosen subset of items (or features) x to put in our knapsack, we can interpret the total weight of each of these items $\alpha^T x = \mathbf{1}^T K x$, as the total number of people in our knapsack, where we count each person once for each distinct feature they have. This means if person i has 4 distinct features from poi, then they will be counted 4 times and provide a weight of 4 into the knapsack. As $(Kx)_i$ represents the number of unique features individual i has from poi, every person is added $(Kx)_i$ times to the bag.

In other words, $(Kx)_i$ represents how many items individual i is in out of the subset x. In this way, counting the number of "unique" people in our knapsack corresponds to counting the number of people with ≥ 1 distinct feature from poi, which is equivalent to $||Kx||_0$. To illustrate this, consider an example K, and feature vector x

$$K = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \quad x = [1,1,0]^T, \quad Kx = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 1 \end{bmatrix}, \quad \alpha = \begin{bmatrix} 2 & 2 & 1 \end{bmatrix}$$

Where we he have three individuals (call them A,B,C) aside from poi (row 1) and three features. We can view the items in our knapsack $\alpha = \mathbf{1}^T K$ as the following objects (one for each of the three features).



Here we only select the first two items $x = [1,1,0]^T$, so the weight of our knapsack $\mathbf{1}^T Kx = \alpha^T x = 4$. However, there are only three unique individuals in our knapsack, as we double count B. In order to only count B once, we just take $||Kx||_0 = 3$ (i.e. count the number of non-zeros)

5.2 Convex Upper Bounds

5.2.1 SDP Upper bound

Consider the following SDP:

$$\begin{aligned} \text{IPT}^{\text{SDP}} = \quad & \underset{x \in \mathbb{R}^n}{\text{maximize}} \quad w^T x \\ & \text{subject to} \quad \begin{bmatrix} X & Z & x \\ Z^T & Y & y \\ x^T & y^T & 1 \end{bmatrix} \succeq 0 \\ & \qquad\qquad\quad x_i = X_{ii}, \quad \forall i \leq n \\ & \qquad\qquad\quad y_j = Y_{jj}, \quad \forall j \leq m \\ & \qquad\qquad\quad \mathbf{1}^T y \geq rm \\ & \qquad\qquad\quad 0 \leq x, y, X, Y, Z \leq 1 \end{aligned} \tag{IPT$^{\text{SDP}}$}$$

To see why IPT $\leq$ IPT$^{\text{SDP}}$, we would like to show that

$$x^* \text{ feasible to IPT} \implies \exists x, y, X, Z, Y \text{ feasible to IPT}^{\text{SDP}}$$

Take x^* feasible to IPT and let $x = x^*$. Next define a vector $y \in \mathbb{R}^n$ with each entry define as

$$y_i = \begin{cases} 1 & \text{if } (Kx)_i = 0 \\ 0 & \text{otherwise} \end{cases}$$

In other words, $y_i = 1$ if all subfeatures of individaul i are shared with poi. As $\mathbf{1}^T y$ represents the number of individauls indistinguishable from poi, clearly we will satisfy the constraint that $\mathbf{1}^T y \geq rm$ by x being feasible to IPT. As x, y are also binary vectors, we satisfy that equality as well. Next define

$$X = xx^T, \; Z = xy^T, \; Y = yy^T$$

and by Schur's complement, we satisfy the psd constraint, and the equality constraints. So we have feasiblity to IPT^{SDP}. Note that of course we can add on a the linear constraint from the MIP upper bound $\mathbf{1}^T Kx \leq C\mathbf{1}^T x$ to get a tighter SDP relaxation.

6 Algorithmic approximations

6.1 Greedy Lower Bound

Define $\bar{K} = 1 - K$ as the matrix with the entries in K flipped. Now for any column $\bar{K}_i$ of matrix $\bar{K}$, define its value

$$v_i = (1^T \bar{K}_i) w_i$$

and order the columns from largest to smallest value, then perform a greedy selection algorithm by selecting the columns with the largest value up until the privacy constraint is violated. The idea is we both want to select columns which are weighted highly and who share features with the kth column. Note that we can always rewrite $\mathbf{1}^T \bar{K} = m - \mathbf{1}^T K$, so this formulation need not involve a new $\overline{K}$.[1]. Together, the greedy algorithm is the following: of course the output of this algorithm x^* will be feasible to IPT, and therefore $w^T x^*$ will be a lower bound.

Algorithm 1 Greedy Algorithm for IPT

```
$v_i \leftarrow (\mathbf{1}^T \bar{K}_i) w_i, \quad \forall i \leq n$
$x_i \leftarrow 0, \quad \forall i \leq n$
for $j = 1 : n$ do
    $k$ = idx of $j$th largest element of $v$
    $x^{\text{copy}} \leftarrow x$
    $x_k^{\text{copy}} \leftarrow 1$
    if $||Kx^{\text{copy}}||_0 \leq C$ then
        $x \leftarrow x^{\text{copy}}$
return $x$
```

[1] Inspired by question 1 on the 2016 ORF 363 Final

6.2 KNAP Dynamic Programming Lower Bound

Inspired by the original Dynamic Programming Knapsack algorithm from [8] we give a dynamic programming lower bound with a runtime complexity of $\mathbf{O}(nC)$. This works in the following way: first we define a smaller IPT problem

$$\begin{array}{rll} \mathrm{IPT}_j(d) = & \underset{x}{\text{maximize}} & \sum_{i=1}^{j} w_i x_i \\ & \text{subject to} & \|Kx\|_0^j \le d \\ & & x_i \in \{0,1\} \end{array} \tag{$\mathrm{IPT}_j(d)$}$$

where we use the notation $||y||_0^j$ to represent the number of non-zero elements in the first j entries of some vector y. In other words, $\mathrm{IPT}_j(d)$ represents the optimal value for subproblem consisting of the first j features and capacity of d, where $d \in \{0, 1, ..., C\}$. We can represent the optimal subset of revealed features corresponding to the above problem as $X_j(d)$. Next, by letting K_j be the jth column of the matrix K, we will define a quantity $z_j(d)$ as the following:

$$z_j(d) = \begin{cases} z_{j-1}(d) & \text{if } \mathbf{1}^T K_j > d \\ \max\{z_{j-1}(d), z_{j-1}(d - \mathbf{1}^T K_j) + w_j\} & \text{otherwise} \end{cases}$$

for $j \in \{0, ..., n\}$, $d \in \{0, ...C\}$, and where $z_0(d) = 0, \forall d \le C$. The idea is that for each incremental feature j we are considering. We choose to reveal j only if

Theorem 6.1. *$z_j(d) \le IPT_j(d)$, $\forall j, d$*

Proof. We will prove this by induction for any arbitrary d. As a base case when $j = 0$ we have $z_j(d) = \mathrm{IPT}_j(d)$ and so the inequality is trivially satisfied. Now fixing j we may assume by our inductive hypothesis that $z_{j-1}(d) \le \mathrm{IPT}_{j-1}(d)$. Now if $\mathbf{1}^T K_j > d$, then we get that $z_j(d) = z_{j-1}(d)$ and $\mathrm{IPT}_j(d) = \mathrm{IPT}_{j-1}(d)$, so the inequality is satisfied. Now consider the case where $\mathbf{1}^T K_j \le d$. We have

$$\max\{z_{j-1}(d), z_{j-1}(d - \mathbf{1}^T K_j) + w_j\} \le \max\{\mathrm{IPT}_{j-1}(d), \mathrm{IPT}_{j-1}(d - \mathbf{1}^T K_j) + w_j\}.$$

Next recall the optimal subset $X_{j-1}(d)$ for the corresponding $\mathrm{IPT}_j(d)$ subproblem, and define $\bar{X}_j(d) \in \{0,1\}^n$ as the optimal subset of features to reveal out of the first j features, conditional on feature j being revealed. By assumption let us assume $\bar{X}_j(d)$ is only defined in the case that $\mathbf{1}^T K_j \le d$. In this case, denote

$$\alpha_j := ||K\bar{X}_j(d)||_0 - ||KX_{j-1}(d)||_0$$

so that α_j denotes the additional number of non-zero entries of (i.e. additional number of distinguishable individuals) revealing a feature j would add to optimal $j - 1$ subset of revealed features. By noting that

$$\mathbf{1}^T K_j \ge \alpha_j$$

we get that

$$d - \mathbf{1}^T K_j \le d - \alpha_j$$

. Now because for any fixed j, $z_j(d)$ is monotonically increasing in d, i.e. $z_j(a) \leq z_j(b)$ for $a \leq b$, we get that $\text{IPT}_{j-1}(d - \mathbf{1}^T K_j) \leq \text{IPT}_{j-1}(d - \alpha_j)$ and so we have

$$\begin{aligned} \max\{\text{IPT}_{j-1}(d), \text{IPT}_{j-1}(d - \mathbf{1}^T K_j) + w_j\} &\leq \max\{\text{IPT}_{j-1}(d), \text{IPT}_{j-1}(d - \alpha_j) + w_j\} \\ &= \text{IPT}_j(d) \end{aligned}$$

so we have by induction that $z_j(d) \leq \text{IPT}_j(d), \ \forall j, d$. Solving for $z_n(C)$ will thus give us a lower bound on $\text{IPT}_n(C)$. ■

The above lower bound on $\text{IPT}_j(d)$ can be achieved via the following dynamic programming algorithm: Of course this method doesn't on its own recover a feature subset $X_j(d)$

Algorithm 2 Dynamic Programming Algorithm for IPT

```
$z_0(d) \leftarrow 0 \quad \forall d \leq C$
for $d = 1 : c$ do
    for $j = 1 : n$ do
        if $\mathbf{1}^T K_j > d$ then
            $z_j(d) \leftarrow z_{j-1}(d)$
        else
            $z_j(d) \leftarrow \max\{z_{j-1}(d), z_{j-1}(d - \mathbf{1}^T K_j) + w_j\}$
return $x$
```

for all j, d, but it is easy to keep track of the optimal solution for each j, d combination. For more memory efficient ways of doing so, see [8]

7 Methods to Solve IPT

Currently, the best method we have for an exact solve of IPT is the exponential time algorithm of evaluating all 2^n subsets of features to reveal. The optimal subset $x^* \in 0, 1^n$ then, is of course the subset which has the largest values of $w^T x^*$ for all x which satisfy the L0-norm constraint.

Future Work

Moving forward, we would like to investigate more efficient algorithms for computing the optimzal values and solution to IPT. In particular, we would like to investigate the submodular maximization and differential privacy literature more deeply, to see if any results could be leveraged for IPT. We would also live to investigate SOS methods such as in [1] for lending convex bounds on IPT via semidefinite optimization. Finally, we are interested in investigating possible convex or integer formulations of the L0 norm, such as in [2] [5]. Finally, there may well be itesting connections between IPT, chordal graphs, and perfect elimination ordering [3]

References

[1] Ahmadi, A. A., Dibek, C., and Hall, G. Sums of separable and quadratic polynomials, 2021.

[2] Atamturk, A., Gomez, A., and Han, S. Sparse and smooth signal estimation: Convexification of l0 formulations, 2018.

[3] Blair, J. R., and Peyton, B. An introduction to chordal graphs and clique trees. In *Graph theory and sparse matrix computation*. Springer, 1993, pp. 1–29.

[4] Dwork, C. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation* (2008), Springer, pp. 1–19.

[5] Feng, M., Mitchell, J. E., Pang, J.-S., Shen, X., and Wächter, A. Complementarity formulations of l0-norm optimization problems. *Industrial Engineering and Management Sciences. Technical Report. Northwestern University, Evanston, IL, USA 5* (2013).

[6] Hassan, M. U., Rehmani, M. H., and Chen, J. Differential privacy techniques for cyber physical systems: a survey. *IEEE Communications Surveys & Tutorials 22*, 1 (2019), 746–789.

[7] Ji, Z., Lipton, Z. C., and Elkan, C. Differential privacy and machine learning: a survey and review. *arXiv preprint arXiv:1412.7584* (2014).

[8] Kellerer, H., Pferschy, U., and Pisinger, D. *Knapsack Problems*. Springer, Berlin, Germany, 2004.

[9] Krause, A., and Golovin, D. Submodular function maximization. *Tractability 3* (2014), 71–104.

[10] McCormick, S. T. Submodular function minimization. *Handbooks in operations research and management science 12* (2005), 321–391.

[11] Salkin, H. M., and De Kluyver, C. A. The knapsack problem: a survey. *Naval Research Logistics Quarterly 22*, 1 (1975), 127–144.

[12] Zhao, Y., and Chen, J. A survey on differential privacy for unstructured data content. *ACM Computing Surveys (CSUR) 54*, 10s (2022), 1–28.