Information Privacy Tradeoff

Amir Ali Ahmadi and Cole Becker

September 2022

1 The Problem

We consider the following decision problem, we denote Information Privacy Tradeoff Problem (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).

1.1 Motivation

Suppose you are a company holding such a dataset $\mathcal{D}$ about various attributes 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 cares about 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.

1.2 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

$$\begin{aligned} \text{IPT} = \quad & \underset{x}{\text{maximize}} \quad w^T x \\ & \text{subject to} \quad ||Kx||_0 < m(1-r) \\ & \qquad\qquad x_i \in \{0,1\} \end{aligned} \tag{IPT}$$

where 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 $m(1-r)$

Convex Upper Bound Define $\bar{K}$ as the matrix with the entries in K flipped, i.e.

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

we can then write an Linear Program (LP) relaxation of IPT as the following problem

$$\begin{aligned} \text{IPT}^{\text{LP}} = \quad & \underset{x}{\text{maximize}} \quad w^T x \\ & \text{subject to} \quad rm1^T x - 1^T \bar{K} x \leq 0 \\ & \qquad\qquad 0 \leq x \leq 1 \end{aligned} \tag{IPT$^{\text{LP}}$}$$

Where we have IPT $\leq$ IPT$^{\text{LP}}$. To see why take any feasible x to (IPT), and consider an indicator vector $y \in \mathbb{R}^m$ with

$$y_i = \begin{cases} 1 & \text{if } d_i \text{ shares the same } x \text{ features with } d_k \\ 0 & \text{otherwise.} \end{cases}$$

We want to ensure that $1^T y \geq rm$. However we claim that $\frac{1^T \bar{K} x}{1^T x} \geq 1^T y$. To see why, note that $(\bar{K}x)_i$ counts the number of features out of x that d_i and d_k share, so

$$\left(\frac{\bar{K}x}{1^T x}\right)_i = \begin{cases} 1 & \text{if } d_i \text{ shares the same } x \text{ features with } d_k \\ \in [0,1) & \text{otherwise} \end{cases}$$

Of course to be sure that at least rm individuals share the same features as the d_kth individual, we can sum to get $\frac{1^T \bar{K} x}{1^T x} \geq 1^T y \geq rm$, and we get the constraint in IPT$^{\text{LP}}$.

1.3 Complexity of IPT

Theorem 1.1. *IPT is NP-complete*

Proof. It is sufficient to show that IPT $\in$ NP and that Knapsack Problem (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:

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

   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.

■

2 Other questions to consider

1. Given an r,w,D, how would we go about calculating Generalized Information Privacy Tradeoff Problem (GIPT) = $\min_k$ IPT

2. what algorithms to consider in order to give a lower bound on IPT
   Idea: for any column $\bar{K}_i$ of matrix $\bar{K}$, define its value

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

   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. Could make sense because we both want to select columns which are weighted highly and who share features with the kth column

   Cole: In a similar essence to question 1 on the 2016 ORF 363 Final

3. are there tighter upper bounds on IPT than the LP one above

To Do

:

1. SDP (tighter) relaxation: Think about SDP relaxation for entire problem, by considering the SDP relaxation for an integer constraint, and seeing if there is an interpretation in the 0-norm part of the problem

2. Better Algorithm: Can well known "Dynamic Programming" algorithm for knapsack be generalized to IPT

3. Better LP relaxation/Bridge between relaxation & algorithm: Jeff's 2016 ORF 363 Final problem, is there a way to draw a connection between LP relaxation and the greedy algorithm (i.e. recall that the greedy algorithm for knapsack was equivalent to LP relaxation and pushing inequalities to equalities etc.)

4. Solving original problem: Oktay Gunluk has written IP formulations of l_0 norm.