An Expectation-Maximization Relaxed Method for Privacy Funnel

Lingyi Chen[1], Jiachuan Ye[1], Shitong Wu[1], Huihui Wu[2†], Hao Wu[1], and Wenyi Zhang[3]

[1]Department of Mathematical Sciences, Tsinghua University, Beijing 100084, China
[2]Yangtze Delta Region Institute (Huzhou),
University of Electronic Science and Technology of China, Huzhou, Zhejiang, 313000, P.R. China.
[3]Department of Electronic Engineering and Information Science,
University of Science and Technology of China, Hefei, Anhui 230027, China
Email: huihui.wu@ieee.org

***Abstract*—The privacy funnel (PF) gives a framework of privacy-preserving data release, where the goal is to release useful data while also limiting the exposure of associated sensitive information. This framework has garnered significant interest due to its broad applications in characterization of the privacy-utility tradeoff. Hence, there is a strong motivation to develop numerical methods with high precision and theoretical convergence guarantees. In this paper, we propose a novel relaxation variant based on Jensen's inequality of the objective function for the computation of the PF problem. This model is proved to be equivalent to the original in terms of optimal solutions and optimal values. Based on our proposed model, we develop an accurate algorithm which only involves closed-form iterations. The convergence of our algorithm is theoretically guaranteed through descent estimation and Pinsker's inequality. Numerical results demonstrate the effectiveness of our proposed algorithm.**

I. Introduction

An increasing amount of private user data is flowing into the network nowadays, probably collected by certain individuals or companies eventually for customizing personalized services or other purposes. Usually, such data contains private or sensitive information. Considering general content of private information and the task of system, the problem is reduced to learning private representations, *i.e.*, representations that are informative of the data (utility) but not of the private information. Researchers have started to model and study privacy protection mechanisms, in order to develop privacy preserving technologies and characterize the privacy-utility tradeoff. A general framework of statistical inference from an information-theoretic perspective has been proposed in [1]. Specifically, given a public variable $X \in \mathcal{X}$ we want to transmit, and a correlated variable $S \in \mathcal{S}$ we want to keep private, one needs to encode X as a variable $Y \in \mathcal{Y}$, forming a Markov chain

$$S \longleftrightarrow X \longrightarrow Y.$$

The first two authors contributed equally to this work and † marked the corresponding author. This work was partially supported by National Key Research and Development Program of China (2018YFA0701603) and National Natural Science Foundation of China (12271289 and 62231022).

Our goal is to minimize the average cost gain by the adversary after observation, while keeping the distortion of privacy-preserving mapping under certain threshold. When the self-information cost and log-loss metric are introduced, the privacy funnel (PF) [2] is formulated to find a privacy preserving mapping from $\mathcal{X}$ to $\mathcal{Y}$, such that it minimizes the average information leakage $I(S;Y)$ with the disclosure $I(X;Y)$ kept above a certain threshold. More precisely, given the joint distribution $P_{S,X}$, the PF problem pursues the above tradeoff by considering the following optimization problem

$$\min_{P_{Y|X}} I(S;Y), \quad \text{s.t. } I(X;Y) \geq R, \tag{1}$$

where $R \leq H(X)$ to ensure that the problem is feasible [2]. Research related to the PF model has covered a variety of setups in information theory [3]–[5], machine learning [6]–[9] and other fields. Moreover, the PF problem can be viewed as a dual of the well-known information bottleneck (IB) problem [2], [10], suggesting an intriguing connection between them.

However, different from the IB problem benefiting from a variety of algorithms including the BA algorithm [11], [12] and the recently proposed ABP algorithm [13], the PF problem still lacks an adequately effective algorithm. In fact, the PF problem is inherently non-convex, and therefore developing its numerical algorithms is a generally challenging task. Several algorithms have been proposed to solve the PF problem, but it is difficult to ensure effectiveness and convergence guarantee simultaneously. The greedy algorithm [2] and the submodularity-based algorithm [14] motivated by agglomerative clustering [15] have been proposed early, merging the alphabet of sanitized variable to construct the mapping. Although strict descent of the objective is ensured, it only descends to a local minimum, resulting in limited computational accuracy and efficiency due to greedy search. The semi-definite programming (SDP) framework [10] has also been applied to the PF problem, yet solving the SDP proves to be a time-consuming endeavor. Besides, variational approaches have been proposed in [16], [17], where each parameter is learnt through gradient descent. They only pro-

vide approximate results and require intensive computations to obtain gradient for each iteration, severely limiting the computational efficiency.

A recent work [18] has proposed a unified framework to solve the IB and PF problems through the Douglas-Rachford splitting (DRS) method, ensuring locally linear rate of convergence. This approach involves solving complex subproblems via gradient descent, and exceedingly large penalty is required in large-scale scenarios to ensure convergence, leading to gradient explosion and numerical instability. Consequently, the approach has been primarily suitable for limited scale cases. Such difficulty has been tackled with variational inference in a subsequent work [19], but it only deals with a surrogate bound as an approximation.

In order to address the aforementioned difficulty, we propose a novel approach to solve the PF model by computing its upper bound relaxation variant, which is derived under the inspiration of the E-step of the celebrating Expectation-Maximization (EM) algorithm [20]. This algorithm, named as the Alternating Expectation Minimization (AEM) algorithm, is developed to minimize the Lagrangian in an alternative manner, where each primal variable can be computed by a closed-form expression, with dual variables similarly updated or searched via Newton's method in only a few inner iterations. The closed-form solutions ensure the efficiency of our algorithm. Moreover, the descent of objective is theoretically estimated, and the convergence of iterative sequence to a Karush-Kuhn-Tucker (KKT) point is guaranteed by the Pinsker's inequality. Numerical experiments exhibit the effectiveness of our algorithm in a wide range of scenarios including traditional distributions and real-world datasets.

II. Problem Formulation

Consider a discrete public variable $X \in \mathcal{X}$ with a relevant private variable $S \in \mathcal{S}$, and a representation variable $Y \in \mathcal{Y}$, where $\mathcal{S} = \{s_1, \cdots, s_K\}$, $\mathcal{X} = \{x_1, \cdots, x_M\}$, and $\mathcal{Y} = \{y_1, \cdots, y_N\}$. Then the Markov chain $S \leftrightarrow X \rightarrow Y$ yields the following distribution

$$P_{S,Y}(s,y) = \sum_{s,x,y} P_{S,X,Y}(s,x,y) = \sum_{x} P_{S|X}(s|x) P_{X,Y}(x,y).$$

Denote $s_{ki} = P_{S|X}(s_k|x_i)$, $u_{ij} = P_{X,Y}(x_i, y_j)$, $w_{ij} = P_{X|Y}(x_i|y_j)$, $r_j = P_Y(y_j)$, $p_i = P_X(x_i)$ and $\hat{R} = R + \sum_i p_i \log p_i$, then the PF problem (1) can be formulated properly as the following constrained optimization problem

$$\min_{\boldsymbol{u},\boldsymbol{w},\boldsymbol{r}} \quad \sum_{j,k} \Big(\sum_i s_{ki} u_{ij} \Big) \Big(\log \big(\sum_i s_{ki} u_{ij} \big) - \log r_j \Big), \tag{2a}$$

$$\text{s.t.} \quad \sum_j u_{ij} = p_i, \ \forall i; \qquad \sum_i u_{ij} = r_j, \ \forall j; \tag{2b}$$

$$\sum_i w_{ij} = 1, \ \forall j; \qquad u_{ij} = w_{ij} r_j, \ \forall i, j; \tag{2c}$$

$$\sum_j r_j = 1; \qquad \sum_{i,j} u_{ij} \log w_{ij} \geq \hat{R}, \tag{2d}$$

where a feasible solution $\mathcal{Y} \supseteq \mathcal{X}$, $p(x|y) = \mathbf{1}\{x = y\}$ exists if $R \leq H(X)$ [2]. It is worth mentioning that our formulation (2) guarantees the convexity with respect to each variable, which ensures numerical stability during optimization.

III. Upper Bound Relaxation Variant and Its Equivalence with the Original Model

For the PF problem (2) formulated above, it is still difficult to solve the variable $\boldsymbol{u}$, mainly due to the term taking the logarithm of the sum in the objective

$$f(\boldsymbol{u}, \boldsymbol{r}) = \sum_{j,k} \Big(\sum_i s_{ki} u_{ij} \Big) \Big(\log \big(\sum_i s_{ki} u_{ij} \big) - \log r_j \Big).$$

Therefore, we propose the following upper bound to relax the problem (for similar techniques, see [9], [21]):

$$\tilde{f}(\boldsymbol{u}, \boldsymbol{r}, \boldsymbol{q}) = \sum_{i,j,k} s_{ki} u_{ij} \Big(\log(s_{ki} u_{ij}) - \log r_j - \log q_{ijk} \Big), \tag{3}$$

where $\sum_i q_{ijk} = 1$, making it much easier to minimize with respect to $\boldsymbol{u}$ from the first-order conditions. The corresponding novel model is the basis for us to design an alternating algorithm.

Surprisingly, as will be shown, the upper bound relaxation variant is equivalent to the original problem (2), so there is no loss considering such a variant.

The following two subsections will specifically elaborate on these two points, constituting our main contributions.

A. Upper Bound Relaxation Variant

First, we notice that the sum $\sum_i s_{ki} u_{ij}$ corresponds to $P_{S,Y}$, so the idea similar to the E-step of the EM algorithm [20] inspires us to estimate $P_{X|S,Y}$ first, from which an upper bound of objective is established. In this scenario, X corresponds to the latent variable in the EM algorithm. More specifically, the upper bound is given by

$$\begin{aligned} &\sum_{j,k} \Big(\sum_i s_{ki} u_{ij} \Big) \log \Big(\sum_i s_{ki} u_{ij} \Big) \\ \leq &\sum_{i,j,k} s_{ki} u_{ij} \Big(\log(s_{ki} u_{ij}) - \log q_{ijk} \Big), \end{aligned}$$

where $\boldsymbol{q}$ is an auxiliary variable such that $\sum_i q_{ijk} = 1$. The equality holds if and only if $q_{ijk} = s_{ki} u_{ij} \Big/ \Big(\sum_{i'} s_{ki'} u_{i'j} \Big)$.

Next, the following constraints (4) given by the Markov chain and the transition probability conditions can be relaxed from our model, similar to the treatment in our previous work [13]. It is reasonable since they can be restored in our update scheme. Moreover, after relaxation we obtain an optimization problem that is convex with respect to each variable. It can be solved by analyzing the Lagrangian with closed-form iterations.

$$\sum_i u_{ij} = r_j, \quad \forall j; \tag{4a}$$

$$u_{ij} = w_{ij} r_j, \quad \forall i, j; \tag{4b}$$

$$q_{ijk} = s_{ki}u_{ij} \Big/ \Big(\sum_{i'} s_{ki'}u_{i'j} \Big), \quad \forall i,j,k. \tag{4c}$$

Under these crucial observations, the process of directly optimizing $\boldsymbol{u}$ in the original PF problem (2) can be transformed to that of estimating $\boldsymbol{q}$ first, and optimizing $\boldsymbol{u}$ thereafter in our upper bound relaxation variant:

$$\min_{\boldsymbol{u},\boldsymbol{w},\boldsymbol{r},\boldsymbol{q}} \quad \sum_{i,j,k} s_{ki}u_{ij}\Big(\log(s_{ki}u_{ij}) - \log r_j - \log q_{ijk} \Big), \tag{5a}$$

$$\text{s.t.} \quad \sum_j u_{ij} = p_i, \ \forall i; \qquad \sum_i w_{ij} = 1, \ \forall j; \tag{5b}$$

$$\sum_j r_j = 1; \qquad \sum_i q_{ijk} = 1, \ \forall j,k; \tag{5c}$$

$$\sum_{i,j} u_{ij} \log w_{ij} \geq \hat{R}. \tag{5d}$$

B. Equivalence of Optimal Solutions

We establish the equivalence between model (2) and its upper bound relaxation variant (5) as shown in the following theorem.

Theorem 1. *The optimal values as well as the optimal triples* $(\boldsymbol{u}^\star, \boldsymbol{w}^\star, \boldsymbol{r}^\star)$ *of* (2) *and* (5) *are identical.*

Proof. Suppose $(\boldsymbol{u}^\star, \boldsymbol{w}^\star, \boldsymbol{r}^\star)$ is optimal for (2), then $(\boldsymbol{u}^\star, \boldsymbol{w}^\star, \boldsymbol{r}^\star, \boldsymbol{q}^\star)$ is feasible for (5) if we define $q^\star_{ijk} = s_{ki}u^\star_{ij} \Big/ \Big(\sum_{i'} s_{ki'}u^\star_{i'j} \Big)$. The expression of $\boldsymbol{q}^\star$ implies the identity of two objectives. Since (5) is an upper bound of (2), $(\boldsymbol{u}^\star, \boldsymbol{w}^\star, \boldsymbol{r}^\star, \boldsymbol{q}^\star)$ is optimal for (5).

On the other hand, suppose $(\boldsymbol{u}^\star, \boldsymbol{w}^\star, \boldsymbol{r}^\star, \boldsymbol{q}^\star)$ is optimal for (5), then the KKT conditions yield the existence of $\gamma^\star \in \mathbb{R}$, $\boldsymbol{\beta}^\star \in \mathbb{R}^N$, $\boldsymbol{\zeta} \in \mathbb{R}^{N\times K}$ such that

$$\gamma^\star - \sum_{i,k} s_{ki}u^\star_{ij} \Big/ r^\star_j = 0, \ r^\star_j = \Big(\sum_i u^\star_{ij} \Big) \Big/ \gamma^\star; \tag{6a}$$

$$\beta^\star_j - \lambda^\star u^\star_{ij}/w^\star_{ij} = 0, \ w^\star_{ij} = \lambda^\star u^\star_{ij}/\beta^\star_j; \tag{6b}$$

$$\zeta^\star_{jk} - s_{ki}u^\star_{ij}/q^\star_{ijk} = 0, \ q^\star_{ijk} = s_{ki}u^\star_{ij}/\zeta^\star_{jk}. \tag{6c}$$

Substituting (6) into (5) we get $\gamma^\star = 1$, $\beta^\star_j = \lambda^\star \sum_i u^\star_{ij} = \lambda^\star r^\star_j$, $\zeta^\star_{jk} = \sum_i s_{ki}u^\star_{ij}$, and thus (4) is satisfied and $(\boldsymbol{u}^\star, \boldsymbol{w}^\star, \boldsymbol{r}^\star)$ is feasible for (2). Suppose it is not optimal for (2), then there exists $(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{r})$ such that

$$\tilde{f}(\boldsymbol{u}, \boldsymbol{r}, \boldsymbol{q}) = f(\boldsymbol{u}, \boldsymbol{r}) < f(\boldsymbol{u}^\star, \boldsymbol{r}^\star) = \tilde{f}(\boldsymbol{u}^\star, \boldsymbol{r}^\star, \boldsymbol{q}^\star),$$

where $q_{ijk} = s_{ki}u_{ij} \Big/ \Big(\sum_{i'} s_{ki'}u_{i'j} \Big)$, and the second equality follows from the expression of $\boldsymbol{q}^\star$. Then $(\boldsymbol{u}^\star, \boldsymbol{w}^\star, \boldsymbol{r}^\star, \boldsymbol{q}^\star)$ is not optimal for (5), which is a contradiction. It means that $(\boldsymbol{u}^\star, \boldsymbol{w}^\star, \boldsymbol{r}^\star)$ is optimal for (2). □

IV. The Alternating Expectation Minimization Algorithm

In this section, we propose a convergence guaranteed alternating algorithm to solve problem (5). Since the update of $\boldsymbol{q}$ corresponds to the E-step of the EM algorithm, and the update of other variables corresponds to the M-step, we name it the Alternating Expectation Minimization (AEM) algorithm.

A. Algorithm Derivation and Implementation

We introduce multipliers $\boldsymbol{\alpha} \in \mathbb{R}^M$, $\boldsymbol{\beta} \in \mathbb{R}^N$, $\gamma \in \mathbb{R}$, $\boldsymbol{\zeta} \in \mathbb{R}^{N\times K}$, $\lambda \in \mathbb{R}^+$ and obtain the Lagrangian of (5):

$$\begin{aligned}\mathcal{L}(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{r}, \boldsymbol{q}; \boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma, \boldsymbol{\zeta}, \lambda) &= \sum_{i,j,k} s_{ki}u_{ij}\Big(\log(s_{ki}u_{ij}) - \log r_j \\ &- \log q_{ijk} \Big) + \sum_i \alpha_i \Big(\sum_j u_{ij} - p_i \Big) + \sum_j \beta_j \Big(\sum_i w_{ij} - 1 \Big) \\ &+ \gamma \Big(\sum_j r_j - 1 \Big) + \sum_{j,k} \zeta_{jk} \Big(\sum_i q_{ijk} - 1 \Big) \\ &- \lambda \Big(\sum_{i,j} u_{ij} \log w_{ij} - \hat{R} \Big).\end{aligned}$$

Our key ingredient is to alternatively update the primal variables with their corresponding dual variables simultaneously updated. Based on the convexity of the Lagrangian with respect to each variable, we take partial derivatives for each primal variable and obtain their closed-form iterative expressions. This update scheme ensures high computation efficiency and offers an accurate descent estimation of the objective.

1) Updating $\boldsymbol{q}$ and $\boldsymbol{\zeta}$: The first-order condition yields

$$\frac{\partial \mathcal{L}}{\partial q_{ijk}} = -\frac{s_{ki}u_{ij}}{q_{ijk}} + \zeta_{jk} = 0, \ q_{ijk} = \frac{s_{ki}u_{ij}}{\zeta_{jk}}.$$

Substituting them into the constraint of $\boldsymbol{q}$ we have

$$\sum_i (s_{ki}u_{ij}) \Big/ \zeta_{jk} = 1, \ \zeta_{jk} = \sum_i s_{ki}u_{ij}.$$

Then we can update $\boldsymbol{q}$ by

$$q_{ijk} = s_{ki}u_{ij} \Big/ \Big(\sum_{i'} s_{ki'}u_{i'j} \Big),$$

which is exactly the relaxed constraint of $\boldsymbol{q}$ in (4c).

2) Updating $\boldsymbol{u}$ and $\boldsymbol{\alpha}, \lambda$: Define $\phi_{ij} = \sum_k s_{ki}\big(\log q_{ijk} - \log s_{ki} \big)$, then the first-order condition yields

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial u_{ij}} &= \log u_{ij} - \phi_{ij} + 1 + \alpha_i - \log r_j - \lambda \log w_{ij} = 0, \\ u_{ij} &= e^{\lambda \log w_{ij} + \phi_{ij} - \alpha_i - 1} r_j.\end{aligned}$$

Substituting them into the constraint of $\boldsymbol{u}$ we have

$$\begin{aligned}\sum_j e^{\lambda \log w_{ij} + \phi_{ij} - \alpha_i - 1} r_j &= p_i, \\ \alpha_i &= \log \Big(\sum_j e^{\lambda \log w_{ij} + \phi_{ij}} r_j \Big) - \log p_i - 1.\end{aligned}$$

Then we can update $\boldsymbol{u}$ by

$$u_{ij} = \frac{e^{\lambda \log w_{ij} + \phi_{ij}} r_j}{\sum_{j'} e^{\lambda \log w_{ij'} + \phi_{ij'}} r_{j'}} p_i,$$

where λ can be updated by finding the unique root of the following monotonic function[1] via the Newton's method:

$$G(\lambda) = \sum_{i,j} \frac{e^{\lambda \log w_{ij} + \phi_{ij}} r_j}{\sum_{j'} e^{\lambda \log w_{ij'} + \phi_{ij'}} r_{j'}} p_i \log w_{ij} - \hat{R} = 0.$$

[1]We can easily verify that $G(\lambda) \geq 0$ as discussed in [22], [23].

3) Updating $\boldsymbol{r}$ and γ: The first-order condition yields

$$\frac{\partial \mathcal{L}}{\partial r_j} = -\sum_{i,k} \frac{s_{ki}u_{ij}}{r_j} + \gamma = 0,\ r_j = \sum_i \frac{u_{ij}}{\gamma}.$$

Substituting them into the constraint of $\boldsymbol{r}$ we have

$$\sum_{i,j} u_{ij} \Big/ \gamma = 1,\ \gamma = 1,$$

which follows from the fact that $\sum_{i,j} u_{ij} = 1$. Then we can update $\boldsymbol{r}$ by

$$r_j = \sum_i u_{ij},$$

which is exactly the relaxed constraint of $\boldsymbol{r}$ in (4a).

4) Updating $\boldsymbol{w}$ and $\boldsymbol{\beta}$: The first-order condition yields

$$\frac{\partial \mathcal{L}}{\partial w_{ij}} = -\frac{\lambda u_{ij}}{w_{ij}} + \beta_j = 0,\ w_{ij} = \frac{\lambda u_{ij}}{\beta_j}.$$

Substituting them the constraint of $\boldsymbol{w}$ we have

$$\sum_i \lambda u_{ij} \Big/ \beta_j = 1,\ \beta_j = \lambda \sum_i u_{ij}.$$

Then we can update $\boldsymbol{w}$ by

$$w_{ij} = u_{ij} \Big/ \Big(\sum_{i'} u_{i'j} \Big) = u_{ij}/r_j,$$

which is exactly the relaxed constraint of $\boldsymbol{w}$ in (4b).

To summarize, the proposed AEM algorithm is presented in Algorithm 1.

Algorithm 1 Alternating Expectation Minimization (AEM)

Input $p_i = p(x_i)$, $s_{ki} = p(s_k|x_i)$, $\hat{R}$, *max_iter*

Output $\min \sum_{i,j,k} s_{ki}u_{ij}\Big(\log(s_{ki}u_{ij}) - \log r_j - \log q_{ijk} \Big)$

Initialize a feasible solution $u_{ij} = \frac{\mathbf{1}\{i=j\}}{M}, r_j = \sum_i u_{ij}$

for $n = 1 :$ *max_iter* **do**

$q_{ijk} \leftarrow s_{ki}u_{ij} \Big/ \Big(\sum_{i'} s_{ki'}u_{i'j} \Big)$

$\phi_{ij} \leftarrow \sum_k s_{ki} \big(\log q_{ijk} - \log s_{ki} \big)$

Find λ such that $G(\lambda) = 0$

$u_{ij} \leftarrow \dfrac{e^{\lambda \log w_{ij} + \phi_{ij}} r_j}{\sum_{j'} e^{\lambda \log w_{ij'} + \phi_{ij'}} r_{j'}} p_i$

$r_j \leftarrow \sum_i u_{ij}$

$w_{ij} \leftarrow u_{ij}/r_j$

Return $\sum_{i,j,k} s_{ki}u_{ij}\Big(\log(s_{ki}u_{ij}) - \log r_j - \log q_{ijk} \Big)$

B. Convergence Analysis

With the guarantee of Theorem 1, we estimate the descent of the objective (3) for model (5). For short, the closed-form updates of each primal variable provide the descent in the form of Kullback-Leibler (KL) divergence between the corresponding variables in two consecutive iterations.

Lemma 1. *The objective $\tilde{f}(\boldsymbol{w}, \boldsymbol{r}, \boldsymbol{q})$ is non-increasing, i.e.*

$$\begin{aligned} &\tilde{f}(\boldsymbol{u}^{(n+1)}, \boldsymbol{r}^{(n+1)}, \boldsymbol{q}^{(n+1)}) \le \tilde{f}(\boldsymbol{u}^{(n+1)}, \boldsymbol{r}^{(n)}, \boldsymbol{q}^{(n+1)}) \\ \le &\tilde{f}(\boldsymbol{u}^{(n)}, \boldsymbol{r}^{(n)}, \boldsymbol{q}^{(n+1)}) \le \tilde{f}(\boldsymbol{u}^{(n)}, \boldsymbol{r}^{(n)}, \boldsymbol{q}^{(n)}). \end{aligned}$$

Moreover, the descent of objective can be estimated by

$$\begin{aligned} &\tilde{f}(\boldsymbol{u}^{(n)}, \boldsymbol{r}^{(n)}, \boldsymbol{q}^{(n)}) - \tilde{f}(\boldsymbol{u}^{(n+1)}, \boldsymbol{r}^{(n+1)}, \boldsymbol{q}^{(n+1)}) \\ &= \sum_{j,k} \Big(\sum_i s_{ki}u_{ij}^{(n)} \Big) D(\boldsymbol{q}_{jk}^{(n+1)} \| \boldsymbol{q}_{jk}^{(n)}) + D(\boldsymbol{r}^{(n+1)} \| \boldsymbol{r}^{(n)}) \\ &\quad + D(\boldsymbol{u}^{(n)} \| \boldsymbol{u}^{(n+1)}) + \lambda^{(n+1)} \sum_j r_j^{(n)} D(\boldsymbol{w}_j^{(n)} \| \boldsymbol{w}_j^{(n-1)}), \end{aligned} \tag{10}$$

where $\boldsymbol{q}_{jk}$ denotes the row of $\boldsymbol{q}$ where the j-th column slice intersects with the k-th vertical slice, and $\boldsymbol{w}_j$ denotes the j-th column of $\boldsymbol{w}$.

Proof. We have the following estimations proved in Appendix A:

$$\begin{aligned} &\tilde{f}(\boldsymbol{u}^{(n)}, \boldsymbol{r}^{(n)}, \boldsymbol{q}^{(n)}) - \tilde{f}(\boldsymbol{u}^{(n)}, \boldsymbol{r}^{(n)}, \boldsymbol{q}^{(n+1)}) \\ &= \sum_{j,k} \Big(\sum_i s_{ki}u_{ij}^{(n)} \Big) D(\boldsymbol{q}_{jk}^{(n+1)} \| \boldsymbol{q}_{jk}^{(n)}), \\ &\tilde{f}(\boldsymbol{u}^{(n)}, \boldsymbol{r}^{(n)}, \boldsymbol{q}^{(n+1)}) - \tilde{f}(\boldsymbol{u}^{(n+1)}, \boldsymbol{r}^{(n)}, \boldsymbol{q}^{(n+1)}) \\ &= D(\boldsymbol{u}^{(n)} \| \boldsymbol{u}^{(n+1)}) + \lambda^{(n+1)} \sum_j r_j^{(n)} D(\boldsymbol{w}_j^{(n)} \| \boldsymbol{w}_j^{(n-1)}), \\ &\tilde{f}(\boldsymbol{u}^{(n+1)}, \boldsymbol{r}^{(n)}, \boldsymbol{q}^{(n+1)}) - f(\boldsymbol{u}^{(n+1)}, \boldsymbol{r}^{(n+1)}, \boldsymbol{q}^{(n+1)}) \\ &= D(\boldsymbol{r}^{(n+1)} \| \boldsymbol{r}^{(n)}). \end{aligned}$$

The non-increasing property follows from the non-negativity of KL divergence. □

Lemma 2. *The objective $\tilde{f}(\boldsymbol{u}, \boldsymbol{r}, \boldsymbol{q})$ is non-negative.*

Proof.

$$\begin{aligned} \tilde{f}(\boldsymbol{u}, \boldsymbol{r}, \boldsymbol{q}) &= \sum_{i,j,k} s_{ki}u_{ij}\Big(\log(s_{ki}u_{ij}) - \log r_j - \log q_{ijk} \Big) \\ &\ge \sum_{i,j,k} s_{ki}u_{ij} \log(s_{ki}u_{ij}) \\ &\ge \Big(\sum_{i,j,k} s_{ki}u_{ij} \Big) \log \Big(\sum_{i,j,k} s_{ki}u_{ij} \Big) = 0. \end{aligned}$$

The first inequality is due to $0 \le r_j \le 1$, $0 \le q_{ijk} \le 1$, and the second inequality follows from Jensen's inequality. □

The objective converges since it is non-increasing and lower bounded throughout iterations. Furthermore, the convergence of iterative sequence is also guaranteed.

Theorem 2. *The sequence $\{(\boldsymbol{u}^{(n)}, \boldsymbol{w}^{(n)}, \boldsymbol{r}^{(n)})\}$ converges to a stationary point $(\boldsymbol{u}^\star, \boldsymbol{w}^\star, \boldsymbol{r}^\star)$.*

Proof. Applying Pinsker's inequality to (10) we have

$$\tilde{f}(\boldsymbol{u}^{(n)}, \boldsymbol{r}^{(n)}, \boldsymbol{q}^{(n)}) - \tilde{f}(\boldsymbol{u}^{(n+1)}, \boldsymbol{r}^{(n+1)}, \boldsymbol{q}^{(n+1)})$$

$$\geq \frac{1}{2}\Bigg(\sum_{j,k} \Big(\sum_i s_{ki} u_{ij}^{(n)} \Big) \|\boldsymbol{q}_{jk}^{(n+1)} - \boldsymbol{q}_{jk}^{(n)}\|^2 + \|\boldsymbol{r}^{(n+1)} - \boldsymbol{r}^{(n)}\|_1^2$$

$$+ \|\boldsymbol{u}^{(n)} - \boldsymbol{u}^{(n+1)}\|_1^2 + \lambda^{(n+1)} \sum_j r_j^{(n)} \|\boldsymbol{w}_j^{(n)} - \boldsymbol{w}_j^{(n-1)}\|_1^2 \Bigg) \geq 0.$$

Since the objective converges, we have

$$\sum_{n=1}^{\infty} \big(\tilde{f}(\boldsymbol{u}^{(n)}, \boldsymbol{r}^{(n)}, \boldsymbol{q}^{(n)}) - \tilde{f}(\boldsymbol{u}^{(n+1)}, \boldsymbol{r}^{(n+1)}, \boldsymbol{q}^{(n+1)}) \big) < +\infty.$$

This implies $\sum_{n=1}^{\infty} \|\boldsymbol{r}^{(n+1)} - \boldsymbol{r}^{(n)}\|_1^2 < +\infty$, so $\{\boldsymbol{r}^{(n)}\}$ converges. Similar analysis goes for $\{\boldsymbol{u}^{(n)}\}$. The update rule of $\boldsymbol{w}$ ensures the convergence of $\{\boldsymbol{w}^{(n)}\}$. Denote the limit point by $(\boldsymbol{u}^\star, \boldsymbol{w}^\star, \boldsymbol{r}^\star)$, then the iterative scheme guarantees the feasibility of the limit point. Taking limit of our iterative expressions, the KKT conditions are satisfied at $(\boldsymbol{u}^\star, \boldsymbol{w}^\star, \boldsymbol{r}^\star)$. □

V. Numerical Results

This section evaluates our AEM algorithm on a synthetic distribution and two real-world datasets of different sizes. These experiments have been implemented by Matlab R2023b on a laptop with 16G RAM and one Intel(R) Core(TM) i7-12700H CPU @ 2.30GHz.

A. Experiments on A Synthetic Distribution

The synthetic conditional distribution in our experiment is given by [18]

$$P_{S|X} = \begin{pmatrix} 0.9 & 0.08 & 0.4 \\ 0.025 & 0.82 & 0.05 \\ 0.075 & 0.1 & 0.55 \end{pmatrix}.$$

We evaluate the performance with uniform and non-uniform P_X respectively given by

$$P_{X,\text{unif}} = \begin{pmatrix} 1/3 & 1/3 & 1/3 \end{pmatrix}^T, P_{X,\text{nonunif}} = \begin{pmatrix} 0.1 & 0.3 & 0.6 \end{pmatrix}^T.$$

In our experiment, we set $N = 4$ and the maximum number of iterations 500. We compare the AEM algorithm with the DRS method [18] and plot PF curves on the information plane given in Fig. 1. The reported values $I(S;Y)$ are the best ones by performing 30 different trials for each $R \leq H(X)$.

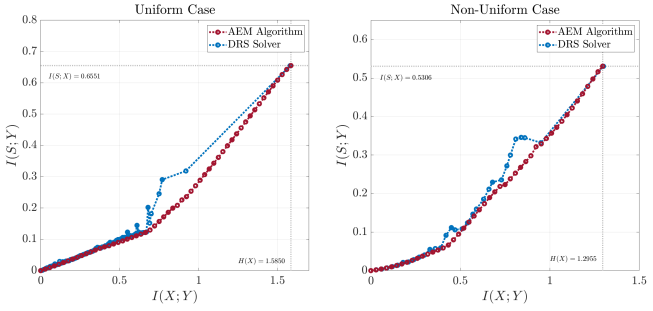

Fig. 1: Comparison of PF curves between the AEM algorithm (red dashed line) and the DRS method (blue dashed line).

Both algorithms reach the theoretical bound $I(X;Y) = H(X)$ and $I(S;Y) = I(S;X)$. Compared to the DRS method, the proposed AEM algorithm provides more uniform points and smoother curves, demonstrating its numerical stability. In contrast, the DRS method performs worse on certain disclosures and shows poorer ability to output complete curves.

B. Experiment on Real-World Datasets

We evaluate the performance on the following two datasets: "Heart failure clinical records" dataset and "Census income" dataset from the UCI Machine Learning Repository [24]. The former dataset has 299 items and 13 attributes. We select $\mathcal{S} = \{$"sex","death"$\}$ and $\mathcal{X} = \{$"anaemia","high blood pressure","diabetes","smoking"$\}$ where all selected attributes are binary, so $|\mathcal{S}| = 4$, $|\mathcal{X}| = 16$. The latter dataset has 32561 items and 14 attributes. We select $\mathcal{S} = \{$"age","income level"$\}$, and $\mathcal{X} = \{$"age","gender","education level"$\}$ where all selected attributes are all integers, so $|\mathcal{S}| = 10$, $|\mathcal{X}| = 160$. The distributions are taken empirically and normalized after adding a perturbation of 10^{-3} to each entry.

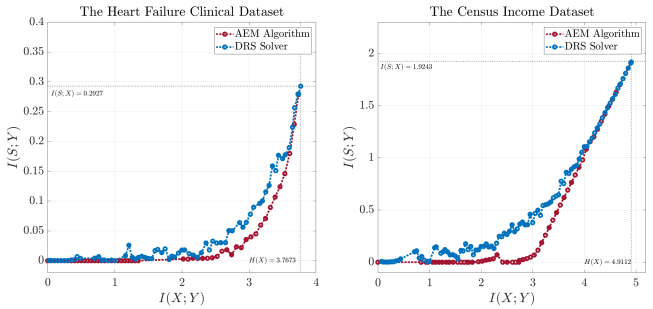

Fig. 2: Performance comparison between the AEM algorithm and the DRS method on two real-world datasets.

As shown in Fig. 2, the AEM algorithm reaches almost perfect privacy (*i.e.*, $I(S;Y) \approx 0$) for a wide range of disclosure thresholds. In contrast, we found that the DRS method performs worse in a range of disclosures. This may be due to the difficulty of striking a balance between convergence and numerical stability, as discussed in Section I. A specific explanation is attached in Appendix B.

VI. Conclusion

We propose a novel approach to solve the PF problem where the objective is replaced with an upper bound under the EM framework, and several constraints given by the Markov chain and transition probability conditions are relaxed. The equivalence between the original model and the upper bound relaxation variant is further proven. Based on the new model, we develop the AEM algorithm by analyzing the Lagrangian, which turns out to recover the relaxed constraints with theoretically guaranteed convergence. Numerical experiments on synthetic and real-world datasets demonstrate effectiveness of our approach. The extension of our approach to continuous scenarios is also interesting and worthy of further research.

REFERENCES

[1] F. du Pin Calmon and N. Fawaz, "Privacy Against Statistical Inference," in *50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2012, pp. 1401–1408.

[2] A. Makhdoumi, S. Salamatian, N. Fawaz, and M. Médard, "From the Information Bottleneck to the Privacy Funnel," in *2014 IEEE Information Theory Workshop (ITW)*, 2014, pp. 501–505.

[3] F. P. Calmon, A. Makhdoumi, and M. Médard, "Fundamental Limits of Perfect Privacy," in *2015 IEEE International Symposium on Information Theory (ISIT)*, 2015, pp. 1796–1800.

[4] F. du Pin Calmon, A. Makhdoumi, M. Médard, M. Varia, M. Christiansen, and K. R. Duffy, "Principal Inertia Components and Applications," *IEEE Transactions on Information Theory*, vol. 63, no. 8, pp. 5011–5038, 2017.

[5] S. Asoodeh, M. Diaz, F. Alajaji, and T. Linder, "Estimation Efficiency Under Privacy Constraints," *IEEE Transactions on Information Theory*, vol. 65, no. 3, pp. 1512–1534, 2019.

[6] M. Romanelli, C. Palamidessi, and K. Chatzikokolakis, "Generating Optimal Privacy-Protection Mechanisms via Machine Learning," *arXiv preprint arXiv:1904.01059*, 2019.

[7] A. Tripathy, Y. Wang, and P. Ishwar, "Privacy-Preserving Adversarial Networks," in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2019, pp. 495–505.

[8] M. Bertran, N. Martinez, A. Papadaki, Q. Qiu, M. Rodrigues, G. Reeves, and G. Sapiro, "Adversarially Learned Representations for Information Obfuscation and Inference," in *36th International Conference on Machine Learning (ICML)*, vol. 97, 2019, pp. 614–623.

[9] B. Rodríguez-Gálvez, R. Thobaben, and M. Skoglund, "A Variational Approach to Privacy and Fairness," in *2021 IEEE Information Theory Workshop (ITW)*, 2021, pp. 1–6.

[10] Y. Bu, T. Wang, and G. W. Wornell, "SDP Methods for Sensitivity-Constrained Privacy Funnel and Information Bottleneck Problems," in *2021 IEEE International Symposium on Information Theory (ISIT)*, 2021, pp. 49–54.

[11] R. Blahut, "Computation of Channel Capacity and Rate-Distortion Functions," *IEEE Transactions on Information Theory*, vol. 18, no. 4, pp. 460–473, 1972.

[12] N. Tishby, F. C. Pereira, and W. Bialek, "The Information Bottleneck Method," in *37th Annual Allerton Conference on Communications, Control and Computing (Allerton)*, 1999, pp. 368–377.

[13] L. Chen, S. Wu, J. Ye, H. Wu, W. Zhang, and H. Wu, "Efficient and Provably Convergent Computation of Information Bottleneck: A Semi-Relaxed Approach," *IEEE International Conference on Communications (ICC)*, 2024, accepted.

[14] N. Ding and P. Sadeghi, "A Submodularity-Based Agglomerative Clustering Algorithm for the Privacy Funnel," *arXiv preprint arXiv:1901.06629*, 2019.

[15] N. Slonim and N. Tishby, "Agglomerative Information Bottleneck," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 12, 1999.

[16] B. Razeghi, F. P. Calmon, D. Gunduz, and S. Voloshynovskiy, "Bottlenecks CLUB: Unifying Information-Theoretic Trade-offs Among Complexity, Leakage, and Utility," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 2060–2075, 2023.

[17] B. Razeghi, P. Rahimi, and S. Marcel, "Deep Variational Privacy Funnel: General Modeling with Applications in Face Recognition," *arXiv preprint arXiv:2401.14792*, 2024.

[18] T.-H. Huang, A. El Gamal, and H. El Gamal, "A Linearly Convergent Douglas-Rachford Splitting Solver for Markovian Information-Theoretic Optimization Problems," *IEEE Transactions on Information Theory*, vol. 69, no. 5, pp. 3372–3399, 2022.

[19] T.-H. Huang and H. E. Gamal, "An Efficient Difference-of-Convex Solver for Privacy Funnel," *arXiv preprint arXiv:2403.04778*, 2024.

[20] M. R. Gupta and Y. Chen, "Theory And Use of the EM Algorithm," *Foundations and Trends® in Signal Processing*, vol. 4, no. 3, pp. 223–296, 2011.

[21] A. Kolchinsky, B. D. Tracey, and D. H. Wolpert, "Nonlinear Information Bottleneck," *Entropy*, vol. 21, no. 12, p. 1181, 2019.

[22] W. Ye, H. Wu, S. Wu, Y. Wang, W. Zhang, H. Wu, and B. Bai, "An Optimal Transport Approach to the Computation of the LM Rate," *IEEE Global Communications Conference (GLOBECOM)*, pp. 239–244, 2022.

[23] S. Wu, W. Ye, H. Wu, H. Wu, W. Zhang, and B. Bai, "A Communication Optimal Transport Approach to the Computation of Rate Distortion Functions," *IEEE Information Theory Workshop (ITW)*, 2023.

[24] C. L. Blake, "UCI Repository of Machine Learning Databases," *https://archive.ics.uci.edu*, 1998.

APPENDIX A
CONVERGENCE ANALYSIS

We estimate the descent caused by the update of each variable in two adjacent iterations.

1) Descent caused by the update of $\boldsymbol{q}$: The update rule of $\boldsymbol{q}$ gives

$$\begin{aligned}
&\tilde{f}(\boldsymbol{u}^{(n)},\boldsymbol{r}^{(n)},\boldsymbol{q}^{(n)})-\tilde{f}(\boldsymbol{u}^{(n)},\boldsymbol{r}^{(n)},\boldsymbol{q}^{(n+1)})\\
=&\sum_{i,j,k} s_{ki}u_{ij}^{(n)}\log\frac{q_{ijk}^{(n+1)}}{q_{ijk}^{(n)}}\\
=&\sum_{j,k}\Big(\sum_i s_{ki}u_{ij}^{(n)}\Big)\sum_i q_{ijk}^{(n+1)}\log\frac{q_{ijk}^{(n+1)}}{q_{ijk}^{(n)}} \qquad (11)\\
=&\sum_{j,k}\Big(\sum_i s_{ki}u_{ij}^{(n)}\Big)D(\boldsymbol{q}_{jk}^{(n+1)}\|\boldsymbol{q}_{jk}^{(n)})\geq 0.
\end{aligned}$$

In (11) we apply the update rule of $\boldsymbol{q}$, then we change the summation order to sum over i first.

2) Descent caused by the update of $\boldsymbol{u}$: The update rule of $\boldsymbol{u}$ gives

$$\begin{aligned}
&\tilde{f}(\boldsymbol{u}^{(n)},\boldsymbol{r}^{(n)},\boldsymbol{q}^{(n+1)})-\tilde{f}(\boldsymbol{u}^{(n+1)},\boldsymbol{r}^{(n)},\boldsymbol{q}^{(n+1)})\\
=&\sum_{i,j,k} u_{ij}^{(n)}\Big(s_{ki}\log\frac{s_{ki}u_{ij}^{(n)}}{q_{ijk}^{(n+1)}r_j^{(n)}}-\lambda^{(n+1)}\log w_{ij}^{(n-1)}\Big)\\
&-\sum_{i,j,k} u_{ij}^{(n+1)}\Big(s_{ki}\log\frac{s_{ki}u_{ij}^{(n+1)}}{q_{ijk}^{(n+1)}r_j^{(n)}}-\lambda^{(n+1)}\log w_{ij}^{(n)}\Big) \qquad (12)\\
=&\sum_{i,j} u_{ij}^{(n)}\log\frac{u_{ij}^{(n)}}{e^{\lambda^{(n+1)}\log w_{ij}^{(n)}+\phi_{ij}^{(n+1)}}r_j^{(n)}}\\
&-\sum_{i,j} u_{ij}^{(n+1)}\log\frac{p_i}{\sum_{j'}e^{\lambda^{(n+1)}\log w_{ij'}^{(n)}+\phi_{ij'}^{(n+1)}}r_{j'}^{(n)}}\\
&+\lambda^{(n+1)}\sum_{i,j}u_{ij}^{(n)}\log\frac{w_{ij}^{(n)}}{w_{ij}^{(n-1)}} \qquad (13)\\
=&\sum_{i,j} u_{ij}^{(n)}\log\frac{u_{ij}^{(n)}}{e^{\lambda^{(n+1)}\log w_{ij}^{(n)}+\phi_{ij}^{(n+1)}}r_j^{(n)}}\\
&-\sum_{i,j} u_{ij}^{(n)}\log\frac{p_i}{\sum_{j'}e^{\lambda^{(n+1)}\log w_{ij'}^{(n)}+\phi_{ij'}^{(n+1)}}r_{j'}^{(n)}}\\
&+\lambda^{(n+1)}\sum_{i,j}u_{ij}^{(n)}\log\frac{w_{ij}^{(n)}}{w_{ij}^{(n-1)}} \qquad (14)\\
=&\sum_{i,j} u_{ij}^{(n)}\log\frac{u_{ij}^{(n)}}{u_{ij}^{(n+1)}}+\lambda^{(n+1)}\sum_{i,j}r_j^{(n)}w_{ij}^{(n)}\log\frac{w_{ij}^{(n)}}{w_{ij}^{(n-1)}}\\
=&D(\boldsymbol{u}^{(n)}\|\boldsymbol{u}^{(n+1)})+\lambda^{(n+1)}\sum_j r_j^{(n)}D(\boldsymbol{w}_j^{(n)}\|\boldsymbol{w}_j^{(n-1)})\geq 0.
\end{aligned}$$

Noticing that the update rule of $\boldsymbol{u}$ yields

$$\hat{R}=\sum_{i,j}u_{ij}^{(n+1)}\log w_{ij}^{(n)}=\sum_{i,j}u_{ij}^{(n)}\log w_{ij}^{(n-1)}$$

in two consecutive iterations, we add these terms to get (12) such that the update rule of $\boldsymbol{u}$ can be applied in (13). The marginal distribution $p_i=\sum_j u_{ij}^{(n)}=\sum_j u_{ij}^{(n+1)}$ implies the derivation of (14). In the final representation, $\boldsymbol{w}_j$ denotes the j-th column of $\boldsymbol{w}$.

3) Descent caused by the update of $\boldsymbol{r}$: The update rule of $\boldsymbol{r}$ yields

$$\begin{aligned}
&\tilde{f}(\boldsymbol{u}^{(n+1)},\boldsymbol{r}^{(n)},\boldsymbol{q}^{(n+1)})-\tilde{f}(\boldsymbol{u}^{(n+1)},\boldsymbol{r}^{(n+1)},\boldsymbol{q}^{(n+1)})\\
=&\sum_{j,k}\Big(\sum_i s_{ki}u_{ij}^{(n+1)}\Big)\log\frac{r_j^{(n+1)}}{r_j^{(n)}}\\
=&\sum_{i,j}u_{ij}^{(n+1)}\log\frac{r_j^{(n+1)}}{r_j^{(n)}}=\sum_j r_j^{(n+1)}\log\frac{r_j^{(n+1)}}{r_j^{(n)}} \qquad (15)\\
=&D(\boldsymbol{r}^{(n+1)}\|\boldsymbol{r}^{(n)})\geq 0.
\end{aligned}$$

We change the summation order to sum over k first, and in (15) we apply the update rule of $\boldsymbol{r}$.

With the descent estimated, we can derive Lemma 1 in the context.

APPENDIX B
EXPLANATION IN LARGE-SCALE EXPERIMENT

In [18], the objective is written as $\mathcal{L}(p,q,v)=F(p)+G(q)+\langle v,Ap-Bq\rangle+\frac{c}{2}\|Ap-Bq\|^2$, where p,q represent $P_{Y|S}$ and $P_{Y|X}$ respectively, A,B are coefficients. The function $G(q)$ is σ_G-weakly convex with $\sigma_G=2|\mathcal{Y}|M_q\Big(|\beta-1|+|\mathcal{X}|\Big)$, where $|\mathcal{X}|,|\mathcal{Y}|$ represent the cardinality of $\mathcal{X}$ and $\mathcal{Y}$ respectively, β is the tradeoff parameter, and $M_q=\epsilon_{Y|X}^{-1}$, $\epsilon_{Y|X}$ is the infimum of $P_{Y|X}$. The subproblem

$$q^{k+1}=\operatorname*{argmin}_{q\in\Omega_q}\mathcal{L}(p^{k+1},q,v^{k+1/2})$$

is convex, and hence can be solved via gradient descent if

$$\begin{aligned}
c>&M_q\frac{M_q\alpha\sigma_G+\sqrt{(M_q\alpha\sigma_G)^2+8(2-\alpha)L_q^2\lambda_B^2\mu_{BB^T}}}{4-2\alpha}\\
>&\frac{\alpha}{2-\alpha}M_q^2\sigma_G>\frac{2\alpha}{2-\alpha}M_q^3|\mathcal{X}||\mathcal{Y}|,
\end{aligned}$$

where α is the relaxation coefficient, λ_B and μ_{BB^T} are constant coefficients determined by B, and $L_q=\epsilon_{Z|X}^{-1}$, $\epsilon_{Z|X}^{-1}$ is the infimum of $P_{Z|X}$.

The lower bound is proportional to $|\mathcal{X}|$ and $|\mathcal{Y}|$, and thus increases with the growing scale of the problem. Meanwhile, too large penalty brings numerical instability since it causes gradient explosion, consequently the numerical computation exceeds the computing ability of our device. Therefore, we can only obtain a locally sub-optimal solution by applying the DRS method to a large-scale dataset.