クラウドソーシングにおけるワーカープライバシを保護した品質管理

梶野　洸† 荒井ひろみ†† 鹿島　久嗣†

† 東京大学大学院情報理工学系研究科　〒 113–8656 東京都文京区本郷 7–3–1
†† 理化学研究所 生命情報基盤研究部門　〒 230–0045 神奈川県横浜市鶴見区末広町 1–7–22
E-mail: †{hiroshi_kajino,kashima}@mist.i.u-tokyo.ac.jp, ††hiromi@base.riken.jp

あらまし　クラウドソーシングとは，不特定多数のワーカーに仕事を依頼するシステムの総称である．その特徴として人手を要するタスクを低コストかつ短時間で大量に処理をすることができる点が挙げられるが，得られる成果物の品質が保証できないという問題が指摘されており，冗長に得た成果物を統合することで品質を高めるという研究が主に行われてきた．本論文ではクラウドソーシングにおけるワーカーのプライバシ漏洩の問題に注目しその重要性を指摘するとともに，ワーカーのプライバシを保護した品質管理プロトコルを提案する．そして提案プロトコルの安全性を理論的に保証し，ワーカーのプライバシを考慮しない既存手法と比べて精度が劣らないことと，実用的な計算時間で計算可能であることを実験的に示す．

キーワード　クラウドソーシング, 機械学習, プライバシ保護データマイニング

Hiroshi KAJINO†, Hiromi ARAI††, and Hisashi KASHIMA†

† Graduate School of Information Science and Technology, The University of Tokyo,
7–3–1 Hongo, Bunkyo–ku, Tokyo 113–8656, Japan.
†† RIKEN Bioinformatics And Systems Engineering division, 1–7–22 Suehiro-cho, Tsurumi-ku, Yokohama,
Kanagawa, 230–0045, Japan.
E-mail: †{hiroshi_kajino,kashima}@mist.i.u-tokyo.ac.jp, ††hiromi@base.riken.jp

Key words　crowdsourcing, machine learning, privacy-preserving data mining

1. 序　　論

クラウドソーシングとは，不特定多数の人間 (ワーカーと呼ぶ) に仕事 (タスクと呼ぶ) を依頼するシステムの総称である．様々なクラウドソーシングが実現されているが，その多くに共通する仕組みは，ID でのみ管理される匿名化されたワーカーがクラウドソーシング上で公開されているタスクをいくつか選択し，そのタスクを完了することで対価として金銭的報酬，情報，コミュニティ内での名声などを得るというものである．クラウドソーシングを用いて仕事を依頼する利点は，大量のワーカーによって低コストでタスクを完了させることができることが挙げられる．そのため人の知能を必要とするタスクを大量に処理するのに適していると言える．

クラウドソーシングは特定のタスクに専門化したものと汎用的なものが存在する．前者の例は様々見られるが，その一例として「測ってガイガー！」(注1)という，放射線量の測定を依頼するウェブサービスが挙げられる．これは特定地点の放射線量を測定するというタスクに特化している．後者の例として挙げられるのが Amazon Mechanical Turk(注2)や CrowdFlower(注3)に代表される汎用的なクラウドソーシングである．タスクの依頼や検索のみならずタスク自体もウェブ上で行えること，どのような種類のタスクでも依頼できることが特徴である．

このように様々な形態のクラウドソーシングが実現されているが，その多くが抱えている問題として，クラウドソーシングを通じて得られる成果物の品質に関する問題が挙げられる．ワーカーの能力は未知である上に，ワーカーはタスクの成果物の品質に対して責任を負う必要がないため，得られる結果の品質は未知である．例としてデータにラベル付けを行うタスクを考え，得られる成果物の一例を図 1 の下部分に示した．図 1 ではワーカー 1,2,3 が同じデータにラベル付けを行なっているが，ワーカー 1 が正しいラベルを付けるのに対し，ワーカー 2 はデータによって誤ったラベルを返し，またワーカー 3 はデータに関係なく同じラベルを返している．クラウドソーシングで得

(注1)：http://hakatte.jp/

(注2)：https://www.mturk.com/mturk/welcome

(注3)：http://crowdflower.com/

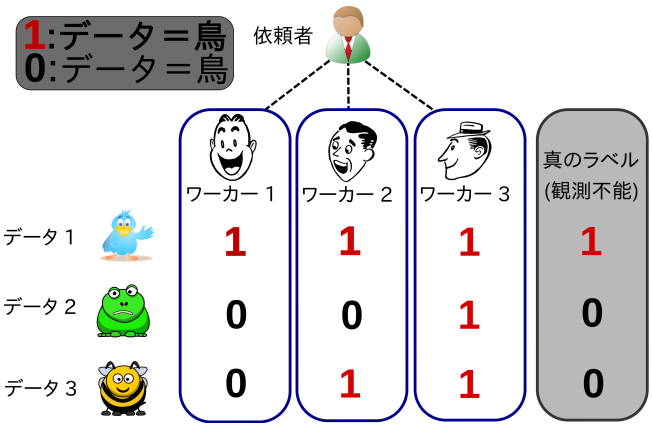

図 1　クラウドソーシングの例を示す．クラウドソーシングは依頼者とワーカーから成る．破線で参加者らの間で可能な通信を示したように，ワーカー同士で通信はできない．タスク例として画像データに対するラベル付けタスクを考える．下部の表では各データに各ワーカーが与えるラベルを表す．ワーカープライベートデータモデルでは，各ワーカーが太枠で囲まれたラベルを秘密に保持している．従来のクラウドソーシングでは依頼者が全ラベルを知ることができるのに対し，提案モデルではラベルを付けたワーカー自身のみがラベルを知ることができる．

られた品質が未知の成果物から正しい成果物を推定する問題を本論文では品質管理問題 [7] と呼ぶ．品質管理問題に対する手法は数多く提案されている．以下に紹介する手法はデータに対してラベルを付けるというタスクを取り扱っているため，以降タスクをラベル付けに限定し成果物をラベルと呼ぶことにする．

品質管理問題に対する代表的な手法は冗長化である．1 つのタスクを複数のワーカーに依頼して品質未知のラベルを複数得て，それを元に真のラベルを推定する手法である．Sheng ら [11] が品質管理問題に対して初めて冗長化法を用いた真のラベルの推定法を提案した．Sheng らは多数決を用いて冗長なラベルから真のラベルを推定した．これを発展させた手法として，ワーカーのラベル付けの過程をモデル化し，そのモデルの推定を通じて冗長なラベルから真のラベルを推定する手法がある．モデルはワーカーの能力を考慮した潜在ラベルモデル [5] が代表的だが，これを拡張したモデルも様々提案されている [16, 17]．このようなモデルを用いることで，真のラベルのみならずワーカーの能力も推定することが可能になり，能力の劣るワーカーを排除することも可能になる．

従来より品質管理問題のようなタスクの依頼者の視点からの問題が考えられてきた一方で，ワーカーの視点からの問題はほとんど考えられていない．本論文ではワーカー視点の問題の 1 つとして，品質管理問題におけるワーカーのプライバシ問題に注目する．品質管理問題では依頼者がワーカーのラベルをすべて収集及び統合することで真のラベルを推定するが，その過程で依頼者は，図 1 中の表のようなワーカーとラベルの対応の情報を取得し，それを用いて真のラベルやワーカーの能力を推定する．またタスクの内容によっては成果物自体にワーカーの位置情報や個人情報，その人の嗜好が反映される場合があり，それらを組み合わせるとワーカー個人が特定される恐れがある．ワーカー個人が特定され，それに紐付く位置情報を含む個人情報も流出することで不利益を被る恐れがある．また能力の低いワーカーは，ワーカーの能力が推定され流出することで依頼者からブロックされてタスクにアクセス出来なくなる恐れがある．以上の議論より品質管理問題に限定した場合でもワーカーのプライバシが侵害されることで様々な問題が生じ，ワーカーが不利益を被ることがわかる．クラウドソーシングにおけるプライバシ問題を取り扱った論文はほとんどなく [3, 13]，特にワーカーのプライバシ問題を取り扱ったのは本論文が初めてである．

本論文はワーカーのプライバシを保護した品質管理手法を新たに提案することで上述の問題を回避する．はじめにワーカープライバシの概念を定義し，ワーカープライバシを保護した品質管理問題を定義する．そしてその問題に対して，代表的な品質管理手法である潜在ラベル法 [5] の計算を分散化および秘匿化した手法を提案する．図 1 に示すように各ワーカーが自分の付けたラベルを秘密に保持した状態でワーカーと依頼者が協調して計算を行うことで，ワーカープライバシを保護しつつ依頼者は真のラベルの推定が可能となる．さらに提案手法の安全性について理論的評価を行い，提案手法の計算時間及び計算精度について実験的に評価及び考察を行う．

以上をまとめると，本論文における成果は次の 4 点である．(i) クラウドソーシングにおける品質管理問題に対するプライバシ問題を定義した．(ii) プライバシを保護した品質管理手法を提案した．(iii) 提案手法の安全性を理論的に保証した．(iv) 提案手法の性質について実験的に評価した．

2. クラウドソーシングと品質管理問題

本章では品質管理問題において用いられる標準的なクラウドソーシングのモデルを紹介し，冗長な品質未知のラベルから真のラベルを推定するという品質管理問題を定義した後に，この問題に対する代表的な手法である潜在ラベル法 [5] を紹介する．

2.1 クラウドソーシング

図 1 のように，クラウドソーシングは依頼者と J (≥ 1) 人のワーカー集合 $\mathcal{J} = \{1, \ldots, J\}$ から成る．クラウドソーシングでは依頼者はタスクを依頼し，またワーカーはタスクを選択し完了することで報酬を得る．本論文では簡単のため，データに対して二値のラベルを与えるタスクのみを考える．ただし本論文のアイデアは多値や連続値のラベル付けを含む様々なタスクに応用できることに注意する．各ワーカーは図 1 に示されるように依頼者とのみ通信を行うことが出来ると仮定する．実際のクラウドソーシングではワーカー間で通信を行うことは難しいため，この仮定は自然な仮定だと言える．

依頼者は I (≥ 1) 個のデータから成るデータ集合 $\mathcal{I} = \{1, \ldots, I\}$ を保持し，各データ $i \in \mathcal{I}$ に対する真のラベル $y_i \in \{0, 1\}$ を得ることを目的とする．真のラベルの集合を $\mathcal{Y}^\star = \{y_i \mid i \in \mathcal{I}\}$ とする．J 人存在するワーカーはそれぞれ $j \in \mathcal{J}$ で表される．各ワーカーはクラウドソーシング上でタス

クを探しラベルを付ける．ワーカー j がデータ i に付けるラベルを $y_{ij} \in \{0,1\}$ と置き，これをクラウドラベルと呼ぶ．このラベルの品質は未知であるとする．つまり一般に $y_i = y_{ij}$ となるとは限らないとする．このラベルの性質についてはワーカーのモデル化に伴い定義される．ワーカー j がラベルを付けたデータ集合を $\mathcal{I}_j \subseteq \mathcal{I}$ $(\mathcal{I}_j \neq \emptyset)$，ワーカー j が付けたラベル集合を $\mathcal{Y}_j = \{y_{ij} \mid i \in \mathcal{I}_j\}$，データ i にラベルを付けたワーカー集合を $\mathcal{J}_i \subseteq \mathcal{J}$ $(\mathcal{J}_i \neq \emptyset)$，データ i に付けられたラベル集合を $\mathcal{Y}_i = \{y_{ij} \mid j \in \mathcal{J}_i\}$，クラウドラベル全体を $\mathcal{Y}$ とする．

品質未知の冗長なクラウドラベルから真のラベルを推定するという問題を品質管理問題と呼ぶ．その定義を問題 1 に示す．ただしこの定義では真のラベルは定義されていないため，モデル化などを通じて定義を行うことが必要となることに注意する．

問題 1 (品質管理問題)．品質管理問題とは，クラウドラベル $\mathcal{Y}$ から，すべてのデータ $\mathcal{I}$ に対する真のラベル $\mathcal{Y}^\star$ を推定する問題である．

2.2 品質管理問題に対する既存手法

本節では品質管理問題 (問題 1) に対する既存手法である潜在ラベル法 [5] を紹介する．この手法では真のラベルとクラウドラベルの間に確率モデル (潜在ラベルモデルと呼ぶ) を仮定し，モデルの推論を通じて真のラベルを推定する．

2.2.1 潜在ラベルモデル

潜在ラベルモデルでは，各データ $i \in \mathcal{I}$ が真のラベル $y_i \in \{0,1\}$ を 1 つ持つと仮定し，各ワーカー $j \in \mathcal{J}$ がデータ i に対してラベル $y_{ij} \in \{0,1\}$ を

$$\alpha_j = \Pr[y_{ij} = 1 \mid y_i = 1, \theta_j],$$
$$\beta_j = \Pr[y_{ij} = 0 \mid y_i = 0, \theta_j].$$

というモデルに従って独立に与えるとする．ここで $\theta_j = \{\alpha_j, \beta_j\}$ をワーカー j の能力パラメタとする．また任意の $i \in \mathcal{I}$ に対して $p = \Pr[y_i = 1]$ が成り立つとする．モデルパラメタ全体を $\Theta = \{\theta_j\}_{j \in \mathcal{J}} \cup \{p\}$ とおく．

2.2.2 推論アルゴリズム

クラウドラベル $\mathcal{Y}$ が得られた時にモデルパラメタと真のラベル $\mathcal{Y}^\star$ を推論するアルゴリズムを紹介する．推論アルゴリズムは，最尤推定量を用いてモデルパラメタを推定する．ただし潜在変数 $\mathcal{Y}^\star$ がモデルに含まれているため，最尤推定量を直接計算することは困難である．そのため EM アルゴリズムを用いてモデルパラメタの推定を行う．アルゴリズムでは次に示す (E-STEP) と (M-STEP) を交互に繰り返す．

(E-STEP)　それぞれの $i \in \mathcal{I}$ に対し

$$a_i \leftarrow \prod_{j \in \mathcal{J}_i} (\alpha_j)^{y_{ij}} (1-\alpha_j)^{1-y_{ij}},$$
$$b_i \leftarrow \prod_{j \in \mathcal{J}_i} (\beta_j)^{1-y_{ij}} (1-\beta_j)^{y_{ij}}, \mu_i \leftarrow \frac{p a_i}{p a_i + (1-p) b_i},$$

と更新．

(M-STEP)　$p \leftarrow \frac{\sum_{i \in \mathcal{I}} \mu_i}{|\mathcal{I}|}$ と更新し，それぞれの $j \in \mathcal{J}$ に対し

$$\alpha_j \leftarrow \frac{\sum_{i \in \mathcal{I}_j} \mu_i y_{ij}}{\sum_{i \in \mathcal{I}_j} \mu_i}, \beta_j \leftarrow \frac{\sum_{i \in \mathcal{I}_j} (1-\mu_i)(1-y_{ij})}{\sum_{i \in \mathcal{I}_j} (1-\mu_i)},$$

と更新．

ここで μ_i, a_i, b_i はそれぞれ $\Pr[y_i = 1 \mid \mathcal{Y}, \Theta], \Pr[\mathcal{Y} \mid y_i = 1, \Theta], \Pr[\mathcal{Y} \mid y_i = 0, \Theta]$ の推定値である．以降 $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_I]$ のように，太字でパラメタの集合を表す．アルゴリズムの収束判定には Q 関数 $Q(\Theta, \boldsymbol{\mu}) = \sum_{i \in \mathcal{I}} [\mu_i \log p a_i + (1-\mu_i) \log(1-p) b_i]$ の相対誤差を用い，$|Q(\Theta^{(t+1)}, \boldsymbol{\mu}^{(t+1)}) - Q(\Theta^{(t)}, \boldsymbol{\mu}^{(t)})| / |Q(\Theta^{(t+1)}, \boldsymbol{\mu}^{(t+1)})| < \epsilon$ となった時に収束したと判断した(注4)．ここで $\Theta^{(t)}$ は t 回目の反復で得られるパラメタを表す．初期化は $\mu_i \leftarrow \frac{\sum_{j \in \mathcal{J}_i} y_{ij}}{|\mathcal{J}_i|}$ とした．

3. ワーカープライバシを保護した品質管理問題

潜在ラベル法を含む既存の品質管理手法はワーカーのプライバシを保護していない問題がある．品質管理手法を適用するためには依頼者がクラウドラベルを回収する必要があり，その結果ワーカーとラベルとの対応が依頼者に漏洩し，序論で議論したような不都合が生じうる．本章ではこのプライバシ問題を取り扱うため，ワーカーに対するプライバシモデルを定義し，ワーカープライバシを保護した品質管理問題を定義する．

プライバシモデルの定義は，誰がどのデータをどの範囲まで公開するかを定義することで行われる．ここで定義するワーカープライベートモデルは，図 1 のように，各ワーカー $j \in \mathcal{J}$ が自分の付けたラベルを秘密に保持し，それを依頼者や他のワーカー $\mathcal{J} \backslash \{j\}$ に知られないという状況をモデル化する．まずラベルがワーカープライベートである定義を次に示す．

定義 1 (ワーカープライベート)．依頼者，ワーカーら $\mathcal{J}$ から成るクラウドソーシングで，各ワーカー $j \in \mathcal{J}$ がラベル $\mathcal{Y}_j$ を付けたとする．$\mathcal{Y}_j$ に含まれるラベルが，ワーカー j のみに知られていて，依頼者，他のワーカーら $\mathcal{J} \backslash \{j\}$ に知られないとき，ラベル $\mathcal{Y}_j$ はワーカープライベートである．

各ワーカーのラベルがワーカープライベートに保たれるようなモデルをワーカープライベートモデルと定める．クラウドラベル $\mathcal{Y}$ をワーカープライベートに保ちつつ真のラベル $\mathcal{Y}^\star$ を推定する問題を，ワーカープライバシを保護した品質管理問題として次のように定義する．

問題 2 (ワーカープライバシを保護した品質管理問題)．依頼者，ワーカーら $\mathcal{J}$ から成るクラウドソーシングで，各ワーカー $j \in \mathcal{J}$ がラベル $\mathcal{Y}_j$ を付けたとする．ワーカープライバシを保護した品質管理問題とは，各ワーカー j のラベル $\mathcal{Y}_j$ をワーカープライベートに保ちつつ，依頼者が真のラベル $\mathcal{Y}^\star$ を推定するという問題である．

4. 提案プロトコル

本章では問題 2 に対する品質管理手法であるワーカープライ

(注4)：実験では $\epsilon = 10^{-8}$ とした．

プロトコル 1 ワーカープライバシ保護潜在ラベルプロトコル

参加者: 依頼者, ワーカーら $\mathcal{J}$.

ワーカー j ($j \in \mathcal{J}$) の秘匿入力: $\mathcal{Y}_j, \mathcal{I}_j$.

依頼者の秘匿入力: $\mathcal{J}$.

公開出力: 各反復 t における $\{a_i^{(t)}, b_i^{(t)}, \mu_i^{(t)} \mid i \in \mathcal{I}\}$.

ワーカー j の秘匿出力: 各反復 t における $\alpha_j^{(t)}, \beta_j^{(t)}$.

設定: 参加者は $(J+1, \theta)$-閾値暗号系を用い，すべてのワーカー $\mathcal{J}$ が公開鍵 pk を共有し，各ワーカー $j \in \mathcal{J}$ が秘密鍵 sk^j を，依頼者が秘密鍵を sk^{J+1} を保持する．

1: 依頼者が $t \leftarrow 0$ と更新し全体に配信.
2: 参加者は，それぞれの $i \in \mathcal{I}$ に対して $\mu_i^{(t)} = \left(\sum_{j \in \mathcal{J}_i} y_{ij}\right) / |\mathcal{J}_i|$ を秘匿加算プロトコル (プロトコル 2) を用いて計算.
3: **repeat**
4: 依頼者が $p^{(t)} \leftarrow \left(\sum_{i \in \mathcal{I}} \mu_i^{(t)}\right) / |\mathcal{I}|$ と更新し全体に配信.
5: 各ワーカー $j \in \mathcal{J}$ が $\alpha_j^{(t)} \leftarrow \left(\sum_{i \in \mathcal{I}_j} \mu_i^{(t)} y_{ij}\right) / \left(\sum_{i \in \mathcal{I}_j} \mu_i^{(t)}\right)$, $\beta_j^{(t)} = \left(\sum_{i \in \mathcal{I}_j} (1-\mu_i^{(t)})(1-y_{ij})\right) / \left(\sum_{i \in \mathcal{I}_j} (1-\mu_i^{(t)})\right)$ と更新.
6: 依頼者が $t \leftarrow t+1$ と更新し全体に配信.
7: 参加者が，それぞれの $i \in \mathcal{I}$ に対して $\log a_i^{(t)}$ と $\log b_i^{(t)}$ を秘匿加算プロトコル (プロトコル 2) を用いて次のように計算: $\log a_i^{(t)} \leftarrow \sum_{j \in \mathcal{J}_i} (y_{ij} \log \alpha_j^{(t-1)} + (1-y_{ij}) \log(1-\alpha_j^{(t-1)}))$, $\log b_i^{(t)} \leftarrow \sum_{j \in \mathcal{J}_i} ((1-y_{ij}) \log \beta_j^{(t-1)} + y_{ij} \log(1-\beta_j^{(t-1)}))$.
8: 依頼者がそれぞれの $i \in \mathcal{I}$ に対して $\mu_i^{(t)} \leftarrow \frac{p^{(t-1)} a_i^{(t)}}{p^{(t-1)} a_i^{(t)} + (1-p^{(t-1)}) b_i^{(t)}}$ と更新し全体に配信.
9: 依頼者が Q 関数の値を計算し，収束判定を行う. $Q^{(t)} = \sum_{i \in \mathcal{I}} \left[\mu_i^{(t)} \log p^{(t)} a_i^{(t)} + (1-\mu_i^{(t)}) \log(1-p^{(t)}) b_i^{(t)}\right]$.
10: **until** $|Q^{(t)} - Q^{(t-1)}| / |Q^{(t)}| < \epsilon$

バシ保護潜在ラベルプロトコルを提案する (以降，提案プロトコルと呼ぶ)．プロトコルとは，複数の参加者が通信を行いながら計算を行うアルゴリズムのことを指す．提案プロトコルは，潜在ラベル法 [5] の計算を分散化および秘匿化したものである．つまり依頼者とワーカーら $\mathcal{J}$ が共に計算主体となってクラウドラベルを秘匿しつつ潜在ラベル法 [5] を実行する．4.1 節でワーカープライバシ保護潜在ラベルプロトコルを提案した後，提案プロトコルで補助的に用いる秘匿加算プロトコルを 4.2 節で提案する．秘匿加算プロトコルとは，参加者 $j \in \{1, \ldots, J\}$ がそれぞれデータ v_j を秘密に保持しているとき，これを他の参加者に知られることなくデータの総和 $\sum_{j=1}^{J} v_j$ を計算するプロトコルのことを指す．

4.1 ワーカープライバシ保護潜在ラベルプロトコル

提案プロトコルをプロトコル 1 に示す．潜在ラベル法を次の 2 点で拡張して問題 2 を解決する．1 点目は計算の分散化である．従来の潜在ラベル法では，依頼者がすべてのクラウドラベルを所有し，依頼者のみが計算主体となって真のラベルの推定を行うのに対し，提案プロトコルでは，ワーカーら $\mathcal{J}$ がそれぞれクラウドラベルを秘密に所有し，ワーカーら $\mathcal{J}$ と依頼者が共に計算主体となって真のラベルの推定を行う．クラウドラベルをワーカープライベートに保つためにはワーカーが計算に参加する必要がある．2 点目は秘匿加算プロトコル (プロトコル 2) を用いる点である．潜在ラベル法の初期化と (E-STEP) では各ワーカー $j \in \mathcal{J}$ が秘密に所持する値 $\alpha_j, \beta_j, \mathcal{Y}_j$ を用いた計算するが，提案プロトコルではこの各ワーカーの値を秘匿しつつ計算結果を得るために秘匿加算プロトコルを用いる．秘匿加算プロトコルを用いずに複数の計算主体で実行する場合，潜在ラベル法と同じように情報漏洩する．なぜなら，プロトコル 1 の 2 行目の計算でクラウドラベルが漏洩し，またプロトコル 1 の 7 行目の計算で，$\{y_{ij} \log \alpha_j^{(t)} + (1-y_{ij}) \log(1-\alpha_j^{(t)}), (1-y_{ij}) \log \beta_j^{(t-1)} + y_{ij} \log(1-\beta_j^{(t-1)}) \mid i \in \mathcal{I}\}$ を依頼者に送る必要があるが，2 行目で得たラベルの情報を用いるとワーカーの能力に関するパラメタ α_j, β_j も依頼者に知られてしまうためである．以上の議論より，この 2 点の拡張は必要であると言える．またワーカープライバシ保護のためにはこの 2 点の拡張で十分であることを 5 章で議論する．

提案プロトコルは潜在ラベル法と比べると，ワーカープライバシを保護できる点で優れているが，計算時間と計算精度の面で劣る．計算の分散化により通信が発生し，また秘匿加算プロトコルで鍵生成や暗号化，復号化が必要になるため計算時間が増加する．さらに秘匿加算プロトコルでは計算誤差が生じる．これらの影響については 6 章で数値実験を用いて評価を行う．

4.2 秘匿加算プロトコル

プロトコル 1 で補助的に用いる秘匿加算プロトコルをプロトコル 2 で提案する．ここで提案する秘匿加算プロトコルは既存の秘匿加算プロトコルと比較すると参加者間の通信方式が新規である．クラウドソーシングではワーカーと依頼者間の通信のみが許容され，ワーカー間の通信を行うことができないため，これを考慮した通信方式を用いた．以降，暗号方式として採用する公開鍵暗号の性質を紹介をした後，秘匿加算プロトコルを提案する．

4.2.1 公開鍵暗号

平文を整数 $m \in \mathbb{Z}_N$ $(= \{0, 1, \ldots, N-1\})$ $(N \in \mathbb{N})$ とし，対応する暗号文を $c = \mathrm{Enc}_{\mathsf{pk}}(m; r) \in \mathbb{Z}_{N^2}^* (= \mathbb{Z}_{N^2} \backslash \{0\})$ とする．公開鍵暗号では公開鍵 pk と乱数 r を用いて暗号化を行う．公開鍵暗号では，乱数 r が $\mathbb{Z}_N^*$ から一様ランダムに選択されると暗号文 c も $\mathbb{Z}_{N^2}^*$ 上を一様ランダムに分布する．故に秘密鍵 sk を知ることなく元の平文を得ることはできない．ここで提案する秘匿加算プロトコルでは，一般化 Paillier 暗号 [4] を用いる．なぜならば，この暗号は加法準同型暗号と (n, θ)-閾値暗号の性質を兼ね備えているからである．以降その両性質と必要性を紹介する．

加法準同型暗号は暗号文を復号せずに平文の加算を実行できる．つまり平文 m_1, m_2 の加算は暗号文同士の乗算を用いて

$$\mathrm{Enc}_{\mathsf{pk}}(m_1 + m_2 \bmod N; r) = \mathrm{Enc}_{\mathsf{pk}}(m_1; r_1) \cdot \mathrm{Enc}_{\mathsf{pk}}(m_2; r_2)$$

として得られる．ここで r_1 または r_2 のどちらかが一様ランダムであれば r は一様ランダムとなるため加算後に得られる暗号文からも平文を推定できない．以降簡単のため乱数を省略する．

(n, θ)-閾値暗号とは，n 人の参加者が秘密鍵を分散して所有し，θ 人以上の参加者が協力をしない限り効率的な復号が行えない性質を持つ暗号である．秘密鍵を分散させる性質により参

プロトコル 2 秘匿加算プロトコル

参加者: 依頼者，ワーカーら $\mathcal{J} = \{1, \ldots, J\}$.

ワーカー $j \in \mathcal{J}$ の秘匿入力: $v_j \in \mathbb{R}$.

公開入力: $\forall j \in \mathcal{J}$ に対して $v_j L \in \mathbb{Z}_N$ となるような $L \in \mathbb{Z}_N$.

設定: $\theta \leqq J$ を満たす $(J+1, \theta)$-閾値加法準同型暗号を用い，全参加者が秘密鍵を分散所持し公開鍵を共有する．

(暗号文を用いた加算)

1: 各ワーカー $j \in \mathcal{J}$ は $v_j L$ の暗号文 $c_j \leftarrow \mathrm{Enc}(v_j L)$ を計算.
2: 各ワーカー $j \in \mathcal{J}$ は暗号文 c_j を依頼者に送信.
3: 依頼者は $\chi \leftarrow \prod_{j \in \mathcal{J}} c_j$ $(= \mathrm{Enc}(\sum_{j \in \mathcal{J}} v_j))$ を計算.

(復号)

1: 依頼者は $|\tilde{\mathcal{J}}| = \theta - 1$ を満たす $\mathcal{J}$ の部分集合 $\tilde{\mathcal{J}} \subseteqq \mathcal{J}$ を選択し，$\tilde{\mathcal{J}}$ に χ を送信.
2: 各ワーカー $\tilde{j} \in \tilde{\mathcal{J}}$ は依頼者に $\mathrm{Dec}_{\tilde{j}}(\chi)$ を送信.
3: 依頼者は全ての $\tilde{j} \in \tilde{\mathcal{J}}$ から復号シェア $\mathrm{Dec}_{\tilde{j}}(\chi)$ を受信.
4: 依頼者は復号シェア $\mathrm{Dec}_{\theta}(\chi)$ を計算.
5: 依頼者は復号シェア $\mathrm{Dec}_{\tilde{j}}(\chi)$ $(\forall \tilde{j} \in \tilde{\mathcal{J}})$ と $\mathrm{Dec}_{\theta}(\chi)$ から $\sum_{j \in \mathcal{J}} v_j L$ を計算.
6: 依頼者は定数 L を用いて $\sum_{j \in \mathcal{J}} v_j \leftarrow \frac{1}{L} \sum_{j \in \mathcal{J}} v_j L$ を計算.

加者が復号することを防ぐことができる．暗号文 c を復号する際には，参加者 j は自身の秘密鍵 sk^j を用いてそれぞれ復号シェア $\mathrm{Dec}_{\mathsf{sk}^j}(c)$ (平文の部分情報に相当) を計算し，θ 以上の復号シェアを組み合わせることによって暗号文が復号される．

4.2.2 提案秘匿加算プロトコル

提案する秘匿加算プロトコルをプロトコル 2 に示す．EM アルゴリズムを秘匿に実行する既存プロトコル [8, 18] で用いられている秘匿加算プロトコルとは通信方式の点で異なる．クラウドソーシングでは依頼者とワーカーの間での通信のみ許容されているため，既存プロトコルで用いられている秘匿加算プロトコルを適用することはできない．

プロトコル 2 の公開入力として，すべての $j \in \mathcal{J}$ に対して $v_j L \in \mathbb{Z}_N$ となるような L を与える必要があるが，現実にはこのような L を与えることは難しい．そのため依頼者が十分大きい L を取り，ワーカーらが $v_j L$ を整数値に丸めてプロトコルを実行する必要がある．故に得られる結果は $\sum_{j \in \mathcal{J}} v_j$ の近似値となる．この近似がワーカープライバシ保護潜在ラベルプロトコルに与える影響については 6 章で議論する．

5. 安全性評価

本章ではプロトコル 1, 2 の安全性評価を行う．その際に，各参加者は semi-honest モデルに従うとする．つまり，各参加者はプロトコルで定められた通りに行動するが，プロトコル実行中に得たすべての情報を蓄積し，秘匿情報を推測しようとするとする．本章では表記の都合上 “$\perp$” というラベルを導入する．$y_{ij} = \perp$ であるとは，ワーカー j がデータ i にラベルを付けていないことを示すラベルである．よってラベルは $\{0, 1\}$ の二値ではなく $\{0, 1, \perp\}$ の三値を取るとする．ワーカープライバシ保護潜在ラベルプロトコル (プロトコル 1) の安全性に関して，定理 1 が成立する．

定理 1 (ワーカープライバシ保護潜在ラベルプロトコルに関する安全性評価)．$|\mathcal{J}| \geqq 3$ と仮定し，それぞれの $i \in \mathcal{I}$ に対して，$y_{ij} \neq y_{ij'} \in \{0, 1, \perp\}$ となる $j, j' \in \mathcal{J}$ が存在すると仮定する．また各参加者は semi-honest モデルに従うと仮定する．これらの仮定のもとで，ワーカープライバシ保護潜在ラベルプロトコル (プロトコル 1) を実行した後，どの参加者も他の参加者の秘匿ラベルを一意に決定できない．

定理 1 の 2 番目の仮定は強い仮定ではない．なぜならば，通常では 1 つのデータに対してすべてのワーカーがラベル付けをすることはなく，いずれかのワーカーのラベルは $\perp$ となるためである．定理 1 を示すために，秘匿加算プロトコル (プロトコル 2) の安全性に関する補題 2 が必要となる．

補題 2 (秘匿加算プロトコルに関する安全性評価)．$|\mathcal{J}| \geqq 3$ とし，依頼者とワーカーらが semi-honest モデルにしたがってプロトコルを実行すると仮定する．またワーカー同士で共謀することはないと仮定する．秘匿加算プロトコルの実行後，依頼者は最終的な計算結果のみを得るが，他の情報を推定することはできない．また，ワーカーらは計算過程で得られた情報から他のワーカーに関する情報を推定することはできない．

補題 2 の略証は secure multiparty computation [6] における証明と同様にして与えられる．ここではワーカー同士の共謀がないことを仮定しているが，クラウドソーシングではワーカーは他にどのワーカーがプロトコルに参加しているか知ることができないため，共謀しないという仮定は自然である．

補題 2 を用いると，プロトコル 1 の安全性評価を行う際には，$\{\log a_i^{(t)}, \log b_i^{(t)}, \mu_i^{(0)} \mid i \in \mathcal{I}, t \in \{1, \ldots, T\}\}$ と，推測者の秘匿情報からの情報漏洩のみを考慮すればよいことがわかる．なぜなら，他の公開出力は補題 2 より保護されているか，これらの値から求めることができるからである．

定理 1 の証明. プロトコル上の反復が T 回で収束するとする $(T \geqq 1)$．どの秘匿ラベル y_{ij} も次の変数群から一意に決定することができないことを示せば良い．

$$\mu_i^{(0)} = \frac{\sum_{j \in \mathcal{J}_i} y_{ij}}{|\mathcal{J}_i|} \ (\forall i \in \mathcal{I}), \tag{1}$$

$$\log a_i^{(t)} = \sum_{j \in \mathcal{J}_i} \left(y_{ij} \log \alpha_j^{(t-1)} + (1 - y_{ij}) \log \left(1 - \alpha_j^{(t-1)}\right)\right), \tag{2}$$

$$\log b_i^{(t)} = \sum_{j \in \mathcal{J}_i} \left((1 - y_{ij}) \log \beta_j^{(t-1)} + y_{ij} \log \left(1 - \beta_j^{(t-1)}\right)\right), \tag{3}$$

$$(\forall i \in \mathcal{I},\ \forall t \in \{1, \ldots, T\}).$$

はじめに依頼者が秘匿ラベルを一意に決定できないことを示す．依頼者が，式 (1), (2), (3) を満たす解 $\{\log \alpha_j^{(t)}, \log \beta_j^{(t)} \mid j \in \mathcal{J}, t \in \{1, \ldots, T\}\}$，$\{y_{ij} \in \{0, 1, \perp\} \mid i \in \mathcal{I}, j \in \mathcal{J}\}$ を得たとする．これらの解から，他の解で同様に式 (1), (2), (3) を満たす解を導出できることを示すことで，秘匿ラベルを一意に決定することができないことを証明する．

仮定より，任意の $i \in \mathcal{I}, j \in \mathcal{J}$ に対して，$y_{ij} \neq y_{ij'}$ とな

るような $j' \in \mathcal{J}\backslash\{j\}$ が必ず存在する．$\{\log \alpha_j^{(t)}, \log \beta_j^{(t)} \mid j \in \mathcal{J}, t \in \{1,\ldots,T\}\}$ と $\{y_{ij} \in \{0,1,\perp\} \mid i \in \mathcal{I}, j \in \mathcal{J}\}$ のすべての変数に対して j と j' を入れ替えることによって，式 (1), (2), (3) を満たす解で，y_{ij} の値が異なる解が得られる．この操作は任意の $i \in \mathcal{I}, j \in \mathcal{J}$ に対して行うことができるため，すべての秘匿ラベル y_{ij} は一意に決定できないことが示された．

次にワーカー $j \in \mathcal{J}$ が秘匿ラベルを一意に決定できないことを示す．$\mathcal{J}$ は依頼者の秘匿情報であるため，ワーカー j は知り得ない．故に，式 (1), (2), (3) を満たす解とプロトコルに参加したワーカー集合 $\tilde{\mathcal{J}}$ を得たとしても，新たなワーカー $j^{\mathrm{new}} \notin \tilde{\mathcal{J}}$ を考えて，そのワーカーと $\tilde{\mathcal{J}}$ 中のワーカーとを入れ替えることで，式 (1), (2), (3) を満たす解で，y_{ij} の値が異なる解が得られる．

以上より定理 1 は示された． □

6. 数 値 実 験

提案プロトコルの性能は潜在ラベル法 [5] と比較すると計算時間と計算精度の点で異なる．提案プロトコル (プロトコル 1) で依頼者とワーカーの間で通信が必要となり，秘匿加算プロトコル (プロトコル 2) において通信，鍵生成，暗号化及び復号化が必要となるため，その分計算時間が必要となる．また 4.2.2 節で議論したように，実際には秘匿加算プロトコルでは和の近似値しか得られず丸め誤差の問題が生じる．本章ではこれらの問題について，提案プロトコルと潜在ラベル法 [5] を用いて数値実験で評価し，実用上問題とならないことを示す．

6.1 データセット

Whitehill ら [17] が作成した “Duchenne Smiles Data Set” を用いた．このデータセットでのタスクは，1 枚の顔画像に対して，愛想笑いか否かのラベルを 1 つ付けるというタスクである．すべての顔写真に対して専門家によるラベルが付いているため，専門家によるラベルを真のラベルとして手法の性能を評価した．データセット中の画像数 (本論文の用語ではデータ数) は $I = 159$ で，専門家によるラベルでは 159 枚の画像のうち 58 枚の画像は愛想笑いでないと判断された．ワーカー数は $J = 20$，全クラウドラベル数は $|\mathcal{Y}| = 3,513$，そのうち愛想笑いでないというクラウドラベルは $1,804$ である．

6.2 実験 1: 近似精度

実験 1 では，各手法で得られたパラメタの相対誤差と秘匿加算プロトコルのパラメタ L との関係を調べた．

6.2.1 実 験 設 定

パラメタ L を $L = 10^0, 10^1, \ldots, 10^{14}$ と変化させて提案プロトコルを実行し，パラメタ $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_J], \boldsymbol{\beta} = [\beta_1, \ldots, \beta_J], \boldsymbol{\mu} = [\mu_1, \ldots, \mu_I], p$ に対して次に定める相対誤差を計算して近似精度を評価した．潜在ラベル法で得られたパラメタの推定値を $\mathbf{x} \in \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\mu}, p\}$ とし，提案プロトコルで得られた対応するパラメタの推定値を $\tilde{\mathbf{x}}$ とした時に，$\|\log \mathbf{x} - \log \tilde{\mathbf{x}}\| / \|\log \mathbf{x}\|$ を相対誤差とした．

さらに各手法で推定した真のラベルの精度 (推定値と真の値が一致したデータの割合) を比較した．各手法ともにクラウドラベルが得られた元での真のラベルに関する事後確率 $\boldsymbol{\mu}$ が得られたとき，$\mu_i > 0.5$ ならば $y_i = 1$，$\mu \leqq 0.5$ ならば $y_i = 0$ と真のラベルを推定した．また多数決を用いてラベルを推定する手法 (多数決法) は秘匿加算プロトコルを用いて簡単に実現できるため比較手法として精度を計算した．

表 1 実データに対して各手法を適用した際の精度の比較．パラメタ L を $10^0, 10^1, \ldots, 10^{14}$ の範囲で変化させたが，いずれのパラメタでも精度は変化しなかった．

多数決法	潜在ラベル法 [5]	提案プロトコル
0.752	0.761	0.761

6.2.2 結　　果

図 2 にパラメタ L とモデルパラメタの推定値の相対誤差のグラフを示す．これを見ると，パラメタ L の値を大きくするに従って相対誤差が減少することがわかる．μ に関する相対誤差は $L = 10^7$ まではあまり大きく減少しなかったが，これはアルゴリズムの収束までの反復回数の違いが寄与していると考えられる．図 3 に，パラメタ L と各アルゴリズムでの反復回数のグラフを示した．図 3 から，$L = 10^7$ より小さい時には潜在ラベル法と提案手法とで反復回数が異なることがわかる．故に近似計算を行った際の計算誤差に加えて反復回数による影響が加わったためこのような結果になったと考えられる．

また表 1 にこのデータセットに各手法を適用した際の精度を示す．提案プロトコルではパラメタ L を $L = 10^0, 10^1, \ldots, 10^{14}$ と変化させたが精度は変化しなかった．

以上の実験より，真のラベルの推定値は L にはあまり左右されないということ，モデルパラメタの推定値も L を大きくすることで許容できる誤差以下にすることができることが示された．

6.3 実験 2: 計算時間

実験 2 では提案プロトコルと潜在ラベル法 [5] との計算時間を比較する．提案プロトコルは，秘匿加算プロトコルにおける鍵生成，暗号化および復号化に関する計算時間と，ワーカーと依頼者間の通信に必要な時間の分余計に計算時間が必要となる．通信時間は環境に大きく依存するため，暗号に必要な計算時間に限定して議論する．計算の分散化により α_j, β_j の計算が高速化される可能性があるが，その影響は十分無視できるものとして今回は取り扱わない．

6.3.1 実 験 設 定

秘匿加算は通常の加算と比較すると，鍵生成に必要な時間 1 回分と，暗号化および復号化に必要な時間がアルゴリズムの反復回数分余計に必要となる．どちらも秘匿加算プロトコルの参加者の人数 (本論文ではワーカー数 J) と鍵長 k に依存する．通例鍵長を $k = 1024$ と固定するため，ワーカー数 J との関係を調べた．1 タスクに関与するワーカー数 J は現実には高々100 人程度であるため $J = 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, \ldots, 100$ と変化させて，鍵生成 1 回に必要な時間と，暗号化および復号化 1 回に必要な時間を計測した．各ワーカー数で 10 回の試行を行い，その平均を計算した．$(J+1, \lceil \frac{2}{3}(J+1) \rceil)$-閾値暗号系を用いて，パラメタ L は計算時間に関係しないため $L = 10^{10}$ とした．秘匿加算プロトコルは UTD Paillier Threshold Encryption

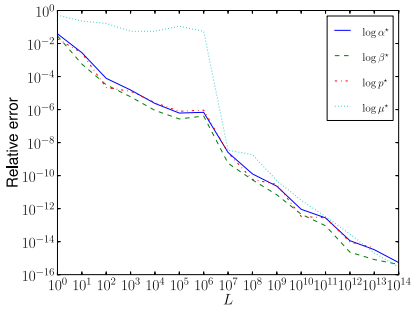

図 2　パラメタ L と，潜在ラベル法と提案プロトコルとで得られたモデルパラメタの相対誤差のグラフ．横軸がパラメタ L，縦軸が各パラメタの対数の相対誤差を表す．パラメタ L を大きくすることで相対誤差が減少する．

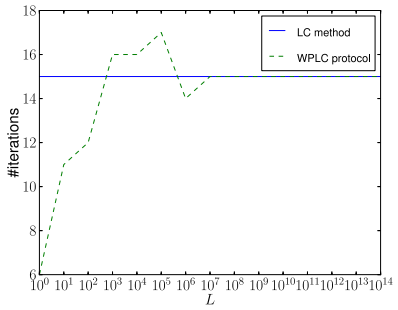

図 3　パラメタ L と，潜在ラベル法と提案プロトコルの反復回数のグラフ．横軸がパラメタ L，縦軸が反復回数を表す．$L = 10^7$ までは反復回数が一致しないため，尤度にも誤差が生じていることがわかる．

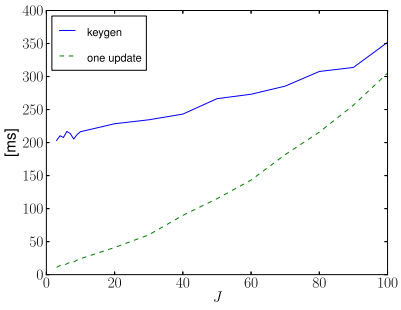

図 4　ワーカー数 J と鍵生成に必要な時間 (keygen) および，1 回の暗号化と復号化に必要な時間 (one update) とのグラフ．横軸がワーカー数 J，縦軸が計算時間をミリ秒で表す．いずれも実用的な範囲内だと言える．

Library[注5] を用いて Java 1.6.0 で実装した．実験は 3.20 GHz (CPU)，4 GB (RAM) のマシンで Linux 上で行った．

6.3.2　結　　果

図 4 にワーカー数 J と鍵生成 1 回に必要な時間 (keygen) と暗号化および復号化 1 回に必要な時間 (one update) とのグラフを示す．いずれもワーカー数 J に対してほぼ線形に大きくなることがわかる．ただしプロトコルを実行する際に鍵生成は 1 回行えばよく，また反復回数も今回用いた実データであっても 15 回程度であるため，これらの計算時間は十分実用の範囲内に収まっていると言える．実データと同じようにワーカー数 $J = 20$，反復回数 15 回としたときに，鍵生成および 15 回の反復計算に必要な計算時間は 789.3 ms であった．このことからも，提案手法を用いた場合でも計算時間は実用上問題ないことが示された．

7.　関 連 研 究

本論文はクラウドソーシングとプライバシ保護データマイニングという 2 つの研究分野と関連している．それぞれの分野と本論文との関係について論ずる．

7.1　クラウドソーシング

2005 年周辺に Amazon Mechanical Turk，ESP ゲーム [14]，reCAPTCHA [15] を含むインターネットを利用した様々なクラウドソーシングサービスが登場して以来，クラウドソーシングに関する研究が盛んに行われてきた．そのなかで最も研究されている問題は成果物の品質管理問題 [7] である．代表的な手法は，1 つのタスクを複数のワーカーに依頼して複数の品質未知の成果物を得て，その成果物を統計処理することで真の成果物を推定する手法である．Sheng ら [11] がこの手法を初めてクラウドソーシングに適用した．Sheng らの用いた統計処理は単純な多数決であったが，その後は Dawid と Skene [5] が医師の診断統合問題の文脈で提案した潜在ラベルモデルやその拡張にあたるモデル [12, 16, 17] を用いた統計処理が研究されている．本研究は品質管理問題から派生した，ワーカープライバシを保護した品質管理問題という新しい問題設定を考えている点でこれらの研究とは異なる．

またクラウドソーシングにおけるプライバシの問題を指摘したのは Bernstein ら [3] が我々の知る限りはじめてである．Varshney [13] はタスクのプライバシ問題を取り扱ったが，本研究はワーカーのプライバシ問題を取り扱ったためこの研究とは差別化される．

7.2　プライバシ保護 EM プロトコル

プライバシ保護データマイニングの概念は同時期に 2 つのグループによって提案された [1, 9]．これ以降この分野において様々な研究が盛んに行われているが，その中で本研究と関連するものは，プライバシを保護して EM アルゴリズムを実行する手法に関する研究である．

Luong と Ho [10] は，混合ガウス分布のパラメタ推定を目的とした．この手法は参加者が 2 人の場合に特化した手法であるため，本論文の問題設定には適用できない．Lin ら [8] も混合ガウス分布のパラメタ推定を目的とした．3 人以上の参加者がそれぞれデータを保持し，補助プロトコルとして準同型暗号に基づく秘匿加算プロトコル [2] を用いる．このプロトコルでは参加者 $\{1, 2, \ldots, J\}$ の中で，$1 \to 2 \to \cdots \to J \to 1$ という順番で環状に通信を行う必要がある．しかし環状の通信はクラウドソーシングでは行えないため，この手法を適用することはできない．Yang ら [18] は，ネットワーク上のノードのクラスタリングを目的とした．補助プロトコルとして Lin ら [8] と同様の通信方式の秘匿加算プロトコルを用いるため，この手法も適用することはできない．

さらに安全性評価の観点で見ると，アルゴリズムの実行中に共有される情報は本論文も他の関連研究も同様であるが，本論文では数学的な安全性評価を行なっている一方で，他の関連研究ではその安全性について明確な議論は行われていない．

(注5)：http://utdallas.edu/~mxk093120/paillier/

これら 2 つの点において本研究は既存研究と差別化される．

8. 結　　論

本論文ではクラウドソーシングで生じるワーカープライバシ問題を紹介し，ワーカープライバシを考慮した品質管理問題を定義し，潜在ラベル法を拡張することで提案した問題を解決した．更に提案プロトコルの安全性について理論的に評価を行い，その実行速度，性能ともに実データを用いて実験的に評価した．これらの評価により，提案手法の有効性を確認した．

謝　　辞

本研究の一部は、内閣府最先端研究開発プログラム (FIRST)「超巨大データベース時代に向けた最高速データベースエンジンの開発と当該エンジンを核とする戦略的社会サービスの実証・評価」の助成を受けたものである。

文　　献

[1] R. Agrawal and R. Srikant. Privacy-Preserving Data Mining. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 439–450, 2000. ISBN 1581132182.

[2] J. C. Benaloh. Secret sharing homomorphisms: keeping shares of a secret secret. In *Advances in Cryptology–CRYPTO '86*, pp. 251–260, 1987.

[3] M. Bernstein, E. H. Chi, L. Chilton, B. Hartmann, A. Kittur, and R. C. Miller. Crowdsourcing and human computation: systems, studies and platforms. In *Proceedings of 2011 ACM Conference on Human Factors in Computing Systems*, pp. 53–56, 2011.

[4] I. Damgård and M. Jurik. A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System. In *Proceedings of the 4th International Workshop on Practice and Theory in Public Key Cryptography: Public Key Cryptography*, pp. 119–136, 2001.

[5] A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):20–28, 1979.

[6] O. Goldreich. *Foundations of cryptography: basic applications*. Cambridge University Press, 2004.

[7] M. Lease. On Quality Control and Machine Learning in Crowdsourcing. In *Proceedings of the third Human Computation Workshop*, pp. 97–102, 2011.

[8] X. Lin, C. Clifton, and M. Zhu. Privacy-preserving clustering with distributed EM mixture modeling. *Knowledge and Information Systems*, 8(1):68–81, 2005.

[9] Y. Lindell and B. Pinkas. Privacy Preserving Data Mining. In *Advances in Cryptology–CRYPTO '00*, pp. 36–54, 2000.

[10] T. D. Luong and T. B. Ho. Privacy Preserving EM-Based Clustering. In *2009 IEEE-RIVF International Conference on Computing and Communication Technologies*, pp. 1–7, 2009.

[11] V. S. Sheng, F. Provost, and P. G. Ipeirotis. Get another label? Improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 614–622, 2008.

[12] Y. Tian and J. Zhu. Learning from crowds in the presence of schools of thought. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 226–234, 2012.

[13] L. R. Varshney. Privacy and Reliability in Crowdsourcing Service Delivery. In *2012 Annual SRII Global Conference*, pp. 55–60, 2012.

[14] L. von Ahn and L. Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '04*, pp. 319–326, 2004.

[15] L. von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum. reCAPTCHA: human-based character recognition via Web security measures. *Science*, 321 (5895):1465–1468, 2008.

[16] P. Welinder, S. Branson, S. Belongie, and P. Perona. The multidimensional wisdom of crowds. In *Advances in Neural Information Processing Systems 23*, pp. 2424–2432, 2010.

[17] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. Movellan. Whose vote should count more: optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems 22*, pp. 2035–2043, 2009.

[18] B. Yang, I. Sato, and H. Nakagawa. Privacy-Preserving EM Algorithm for Clustering on Social Network. In *Advances in Knowledge Discovery and Data Mining - 16th Pacific-Asia Conference, PAKDD 2012*, volume 1, pp. 542–553, 2012.