

クラウドソーシングにおけるオンライン学習

Online Learning from Crowds

チャン クワン カイ*¹ 佐久間 淳*¹

Tran Quang Khai

Jun Sakuma

*¹筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻

Dept. of Computer Science, Graduate school of SIE, University of Tsukuba

In our paper, we propose an online learning algorithm for classification problems using crowdsourcing services. By using the online learning approach, we are able to solve the problem that the existed method couldn't deal with when the reliability of workers and the tend of tasks change over time. Through experiments, we show that under this learning condition the prediction accuracy of the existed method will be decreased. And using online learning method to solve online the existed method, the batch learning, we are able to solve this problem.

1. はじめに

近年, Amazon Mechanical Turk(AMT) などのようなクラウドソーシングサービスを用いて画像, 音声, テキストなどの大量のラベル付きデータを収集する方法が増えている. データを事例とし, 事例にラベルを付けるタスクはコンピュータにとっては難しいが, 人間にとっては比較的簡単である. ここで, クラウドソーシングはインターネットを通じて複数のタスクを不特定の多数の人間に依頼することで短時間, 低コストで大量のラベル付きデータが得られる. タスクを依頼される人をワーカーと呼ぶ.

真のラベルを知らない事例に対して真のラベルとは限らない複数のラベルが与えられたとき真のラベルを求める問題は1979年に初めて Dawid&Skene[1] により研究された. 彼らの目的は単純な多数決よりよい精度を得ることである. 多数決では各ワーカーの付けたラベルから真のラベルを推定したが, Dawid&Skene らの手法では各ワーカーの能力を導入し, 真のラベルを潜在変数として EM アルゴリズムによりモデルが求められる.

2009年に Raykar[2] らは Dawid&Skene らの手法に特徴ベクトルを導入した手法を提案した. 直接に識別モデルを学習可能し, ワーカーがラベルを付けない事例に対しても予測が可能となり, 汎化が可能になった.

Raykar らの手法では真のラベルが潜在変数であるため, 最適化問題が凸最適化問題ではない. さらに, EM アルゴリズムが用いられたため, 推定は初期値に依存し最適な解が得られなくなる場合がある. 2011年に Kajino[3] らはこの問題を解決する識別器統合モデルを提案した. この手法は真のラベルを潜在変数として導入せずにワーカーのラベルから直接に識別モデルを学習するということで非凸最適化問題となることを回避している. この手法ではそれぞれのワーカーに対して識別モデルを仮定し, それらの識別モデルを関係付けることで最終的な識別モデルを得る.

これまでのクラウドソーシングでは, 一つのタスクに対してワーカーに出すタスクの傾向あるいは依頼されたワーカーの能力は変わらないことが想定されてきた. しかし実際には, 依頼者がワーカーに出すタスクの傾向あるいは依頼されたワ

ーカーは, 時間が経過するにつれ変化することは容易に想像できる. このような学習状況では既存手法の性能が低下する.

本研究では, 実験を通じて既存手法による推定精度は, 動的に変化する学習環境では低下することを指摘する. そして, バッチ学習である既存手法をオンライン学習によりオンライン化し, この問題を解決する. また, タスクの傾向やワーカーの能力が変化するという問題について十分な事例数がある場合には提案手法は元のバッチアルゴリズムと同じ性能を持つことを示す.

2. クラウドソーシングにおける分類問題

本節では, 最初にクラウドソーシングにおける分類問題の問題設定を定義し, 次に代表的な既存研究である多数決手法, Raykar らの手法, Kajino らの手法を紹介する.

2.1 問題設定

事例 $\mathbf{x}_i \in \mathbb{R}^d$ ($i \in \{1, \dots, N\}$) が与えられる. それに対し, M 人のワーカーがクラウドソーシングを通してラベルを付ける. 事例 \mathbf{x}_i に対してワーカー j が付けたラベルを $y_{ij} \in \{0, 1\}$ ($j \in \{1, \dots, M\}$) とする. すべてのワーカーがすべての事例にラベルを付けるとは限らない. そこで, すべての事例に少なくとも一人のワーカーがラベルを付け, すべてのワーカーが少なくとも一つの事例にラベルを付けると仮定する. $L_j \subseteq \{1, \dots, N\}$ をワーカー j がラベル付けをする事例の添字集合とし, $K_i \subseteq \{1, \dots, M\}$ を事例 i にラベル付けをするワーカーの添字集合とする. ワーカー j によるラベル集合を $\mathcal{Y}_j = \{y_{ij} | i \in L_j\}$ とし, $\mathcal{Y} = \bigcup_{j=1}^M \mathcal{Y}_j$ を, クラウドソーシングで得られたラベル全体の集合であるとする. また事例全体の集合を $\mathcal{X} = \{\mathbf{x}_i | i \in \{1, \dots, N\}\}$ とする.

事例 \mathbf{x}_i に対して真のラベル y_i が存在すると仮定する (一般的にはこれは観測されていない).

クラウドソーシングにおける分類問題は, この $(\mathcal{X}, \mathcal{Y})$ が訓練データとして与えられたとしたとき, 真のラベルを予測する二値分類器 $f: \mathbb{R}^d \rightarrow \{0, 1\}$ を求める問題として定式化される.

2.2 多数決手法

多数決とは, ある集団において意思決定を図る際に, 多数派の意見を採用する方法のことである. 2.1 節の問題設定より \mathbf{x}_i に対するラベル $\{y_{ij}\}_{j=1}^M$ が得られたとき, 多数決で真のラ

連絡先: Tran Quang Khai, 筑波大学システム情報工学研究科
コンピュータサイエンス専攻, 〒305-8573 茨城県つくば
市天王台 1-1-1, khai@mdl.cs.tsukuba.ac.jp

ル \hat{y}_i を予測する。つまり、次のように多数決を定式化できる。

$$\hat{y}_i = \begin{cases} 1, & \frac{1}{\|K_i\|} \sum_{j \in K_i} y_{ij} > 0.5 \text{ のとき} \\ 0, & \frac{1}{\|K_i\|} \sum_{j \in K_i} y_{ij} < 0.5 \text{ のとき} \\ \text{random,} & \text{その他} \end{cases}$$

そして、この予測結果より識別器を学習する。

2.3 Raykar らの手法による潜在ラベルモデル

Raykar らの手法は 2.1 節で定義したクラウドソーシングにおける分類問題を解決するための手法の一つである。

2.3.1 ラベル生成モデル

Raykar らの手法では Two-coins モデルがワーカーによるラベル生成モデルとして使用された。

$$\alpha_j = Pr[y_j = 1|y = 1].$$

$$\beta_j = Pr[y_j = 0|y = 0].$$

ここで、 α_j は真のラベルが 1 のとき、ワーカー j が事例に 1 を付ける確率であり、感度 (true positive rate) と呼ばれる。 β_j は真のラベルが 0 のとき、ワーカー j が事例に 0 を付ける確率であり、特異度 (1 - false positive rate) と呼ばれる。ただし、この α_j と β_j は事例 \mathbf{x} に依存しないとする。

また、真の識別モデルをロジスティック回帰でモデル化する。このとき事例 \mathbf{x} が与えられ、パラメータ \mathbf{w}_0 を持つ真の識別モデルがラベル $y = 1$ を付ける確率を以下に示す。

$$Pr[y = 1|\mathbf{x}, \mathbf{w}_0] = \sigma(\mathbf{w}_0^\top \mathbf{x}) = (1 + \exp(-\mathbf{w}_0^\top \mathbf{x}))^{-1}$$

2.3.2 目的関数

訓練データが与えられたときパラメータ $\theta = \{\mathbf{w}_0, \{\alpha_j\}_{j=1}^M, \{\beta_j\}_{j=1}^M\}$ の推定を最尤推定で行う。 \mathcal{Y} の $\mathbf{w}_0, \{\alpha_j\}_{j=1}^M, \{\beta_j\}_{j=1}^M$ に関する尤度関数は

$$Pr[\mathcal{Y}|\theta, \mathcal{X}] = \prod_{i=1}^N [p_i a_i + (1 - p_i) b_i]$$

である。ここで、 $p_i = \sigma(\mathbf{w}_0^\top, \mathbf{x}_i)$, $a_i = \prod_{j \in K_i} \alpha_j^{y_{ij}} (1 - \alpha_j)^{(1 - y_{ij})}$, $b_i = \prod_{j \in K_i} \beta_j^{(1 - y_{ij})} (1 - \beta_j)^{y_{ij}}$ とする。

2.3.3 アルゴリズム

真のラベル $\{y_i\}_{i=1}^N$ を潜在変数として以下のように対数尤度が求められる。

$$\ln Pr[\mathcal{Y}, \{y_i\}_{i=1}^N|\theta, \mathcal{X}] = \sum_{i=1}^N y_i \ln p_i a_i + (1 - y_i) \ln(1 - p_i) b_i$$

ここで、EM アルゴリズムを利用する。

2.4 Kajino らの手法による識別器統合モデル

Kajino らの手法は 2.1 節で定義したクラウドソーシングにおける分類問題を解決するための手法の一つである。

2.4.1 ラベル生成モデル

ワーカーによるラベル生成モデルを今回では真の識別モデルと同様にロジスティック回帰モデルとして与える。

事例 \mathbf{x} が与えられたとき、パラメータ \mathbf{w}_0 を持つ真の識別モデルとパラメータ $\mathbf{w}_j (j \in \{1, \dots, M\})$ を持つ各ワーカーの識別モデルが事例にラベル 1 を付ける確率はそれぞれ以下で与えられる。

$$Pr[y = 1|\mathbf{x}, \mathbf{w}_0] = \sigma(\mathbf{w}_0^\top \mathbf{x})$$

$$Pr[y_j = 1|\mathbf{x}, \mathbf{w}_j] = \sigma(\mathbf{w}_j^\top \mathbf{x})$$

真の識別モデルのパラメータは事前分布 $Pr[\mathbf{w}_0]$ から生成されるとし、ワーカー j の識別モデルのパラメータは $Pr[\mathbf{w}_j|\mathbf{w}_0]$ からそれぞれ独立に生成されるとする。これらを正規分布と仮定する。つまり、 $Pr[\mathbf{w}_0|\eta] = \mathcal{N}(\mathbf{0}, \eta^{-1} \mathbf{I})$, $Pr[\mathbf{w}_j|\mathbf{w}_0, \lambda] = \mathcal{N}(\mathbf{w}_0, \lambda^{-1} \mathbf{I})$ とする。ここで η, λ は正の定数であるとする。

2.4.2 目的関数

訓練データが与えられ、パラメータ $\mathbf{w}_0, \mathbf{W} = \{\mathbf{w}_j|j \in \{1, \dots, M\}\}$ の事後確率を最大化しパラメータを推定する。 \mathbf{w}_0, \mathbf{W} に関する負の対数事後確率 $F(\mathbf{w}_0, \mathbf{W})$ は

$$F(\mathbf{w}_0, \mathbf{W}) = \frac{\lambda}{2} \sum_{j=1}^M \|\mathbf{w}_j - \mathbf{w}_0\|^2 + \frac{\eta}{2} \|\mathbf{w}_0\|^2 - \sum_{j=1}^M \sum_{i \in L_j} [y_{ij} \log \sigma(\mathbf{w}_j^\top \mathbf{x}_i) + (1 - y_{ij}) \log(1 - \sigma(\mathbf{w}_j^\top \mathbf{x}_i))]. \quad (1)$$

である。

2.4.3 アルゴリズム

\mathbf{W} は \mathbf{w}_0 に対して独立のため、 \mathbf{w}_0 に関する最適化と \mathbf{W} に関する最適化の 2 つのステップからなる最適アルゴリズムが導かれる。詳細は [3] を参照する。

最初、単に多数決よりよい結果を得るためにクラウドソーシングの研究が始まった。ワーカーの信頼度や問題の難易度など様々な要素を考慮して学習を行ったが、Raykar らの手法を含むほとんどの既存手法は EM アルゴリズムを利用するため、非凸最適化問題となる。ここで、Kajino らが EM アルゴリズムを利用せずに識別統合モデルを提案した。これより、非凸を回避し、凸最適化問題になるため、よりよい結果が得られた。

3. クラウドソーシングにおけるオンライン学習

本節では時間とともにタスクの傾向あるいはワーカーの能力が変わる問題を解決するために 3.3 節で提案手法を導出する。その問題を 3.2 節で定義する。さらに、提案手法を導出するため必要となるオンライン凸計画を 3.1 節で説明する。

3.1 オンライン凸計画

次にオンライン凸計画に関する定義とアルゴリズムが Zinkevich[4] により記述された。

3.1.1 定義

凸実行可能集合 F で凸コスト関数 $c: F \rightarrow \mathbb{R}$ が最小となる値を求める最適化問題は凸計画問題と知らせる。ここで、オンライン凸計画問題を次に定義する。

オンライン凸計画問題は凸実行可能集合 $F \subseteq \mathbb{R}^n$ と無限シーケンス $\{c^1, c^2, \dots\}$ からなる。ただし、 $c^t: F \rightarrow \mathbb{R}$ は凸関数である。

各時刻 t において、オンライン凸計画アルゴリズムは 1 つのベクトル $\mathbf{x}^t \in F$ を選ぶ。そのベクトルを選んだあと、コスト関数 c^t が与えられる。

コスト関数で真のコストを得る前に F から 1 つのベクトルを選ばなければならない。オンラインアルゴリズムは解は見つけれないかもしれないが、その解の近い値を求めることができる。

3.1.2 アルゴリズム

次のように仮定する. (1) 実行可能集合 F は有界集合であり, 閉集合であり, 空集合ではない, (2) $\forall t, c^t$ は微分可能である, (3) $\forall t$, かつ $\forall \mathbf{x} \in F$ において, $\|\nabla c^t(\mathbf{x})\| \leq N$ を満たす $N \in \mathbb{R}$ が存在する, (4) $\forall t, \mathbf{x}$ が与えられたとき $\nabla c^t(\mathbf{x})$ を求めるアルゴリズムが存在する, (5) $\forall \mathbf{y} \in \mathbb{R}^n, \arg \min_{\mathbf{x} \in F} d(\mathbf{x}, \mathbf{y})$ を求めるアルゴリズムが存在する. ここで, $P(\mathbf{y}) = \arg \min_{\mathbf{x} \in F} d(\mathbf{x}, \mathbf{y})$ を射影と定義する.

上記の仮定に基づいて, オンライン凸計画アルゴリズムを以下に示す.

Algorithm 1 Greedy Projection

任意の $\mathbf{x}^1 \in F$ と学習率のシーケンス $\eta_1, \eta_2, \dots \in \mathbb{R}^+$ を選ぶ. 時刻 t に, コスト関数が与えられたとき, 次の時刻のベクトル \mathbf{x}^{t+1} を以下の更新式で選ぶ:

$$\mathbf{x}^{t+1} = P(\mathbf{x}^t - \eta_t \nabla c^t(\mathbf{x}^t))$$

今回は, 射影は必要はないため, 上の更新式が以下のようになる.

$$\mathbf{x}^{t+1} = \mathbf{x}^t - \eta_t \nabla c^t(\mathbf{x}^t) \quad (2)$$

3.1.3 オンライン凸計画のメリットとデメリット

オンライン凸計画のデメリットは学習した訓練データが少ないときはバッチの手法より性能が低い点である. ただし, 訓練データが十分多いときはバッチの手法と同程度の性能が持てる.

オンライン凸計画のメリットは一つずつ事例を学習しているため記憶容量が少ない. さらに, 時間とともに変化する識別モデルに対してオンライン凸計画はコスト関数を最小化する.

3.2 問題設定

本節ではオンラインのようにある時刻 $t \in \{1, \dots, T\}$ に事例 \mathbf{x}^t とそれに対するラベル $y_j^t (j \in \{1, \dots, M\})$ を含む一つのデータが入力されると仮定する. 時刻 t ごとに $(\mathbf{x}^t, \{y_j^t\}_{j=1}^M)$ が訓練データとして与えられたとしたとき, コスト関数となる二値分類器 $c^t: \mathbb{R}^d \rightarrow \{0, 1\}$ を学習する.

3.3 提案手法

式 (1) より Kajino らが提案した識別器統合モデルの目的関数が凸関数であることが分かる. オンライン凸計画の視点から見るとその目的関数はすべての事例に対するコストの総和に相当する. ここで, 3.3.1 節で識別器統合モデルをオンライン化する過程を説明し, 3.3.2 節でオンライン化した問題を解くことを説明する.

3.3.1 識別器統合モデルのオンライン化

時刻 t におけるコストは下式で与えられる.

$$\begin{aligned} c^t(\mathbf{w}_0, \mathbf{W}) = & \\ & - \sum_{j=1}^M [y_j^t \log \sigma(\mathbf{w}_j^\top \mathbf{x}^t) + (1 - y_j^t) \log(1 - \sigma(\mathbf{w}_j^\top \mathbf{x}^t))] \\ & + \frac{\lambda}{2} \sum_{j=1}^M \|\mathbf{w}_j - \mathbf{w}_0\|^2 + \frac{\eta}{2} \|\mathbf{w}_0\|^2. \end{aligned} \quad (3)$$

式 (3) よりコスト関数が凸関数であり, 3.1 節のアルゴリズムの条件を満たしているため, アルゴリズム (1) を利用することができる.

3.3.2 アルゴリズム

式 (3) のコスト関数に対してアルゴリズム (2.4.3) と同様に $\{\mathbf{w}_j\}_{j=1}^M$ と \mathbf{w}_0 との独立性を用いると, 次のように \mathbf{w}_0 と \mathbf{W} の更新は以下のように与えられる.

- ステップ 1. \mathbf{w}_0 を更新する. $\{\mathbf{w}_j\}_{j=1}^M$ を固定し, \mathbf{w}_0 を更新する.

$$\mathbf{w}_0^{t+1} = \frac{\lambda \sum_{j=1}^M \mathbf{w}_j}{\eta + M\lambda}$$

- ステップ 2. \mathbf{W} を更新する. \mathbf{w}_0 を固定すると $\{\mathbf{w}_j\}_{j=1}^M$ はお互いに独立である. そのためそれぞれの $\mathbf{w}_j (j \in \{1, \dots, M\})$ を更新することができる.

$$\mathbf{w}_j^{t+1} = \mathbf{w}_j^t - \alpha_t \mathbf{g}^t(\mathbf{w}_j, \mathbf{w}_0)$$

ここで, α_t はステップ幅であり, 時間 t とともに変わる. ただし, 勾配 $\mathbf{g}^t(\mathbf{w}_j, \mathbf{w}_0)$ は下式で与えられる.

$$\mathbf{g}^t(\mathbf{w}_j, \mathbf{w}_0) = -(y_j^t - \sigma(\mathbf{w}_j^\top \mathbf{x}^t)) \mathbf{x}^t + \lambda(\mathbf{w}_j^t - \mathbf{w}_0)$$

4. 実験

本実験では人工データとベンチマークデータの2種類のデータセットを利用する. 提案手法との性能評価のために, 比較手法として多数決手法, Raykar らの手法と Kajino らの手法を用いる. 本実験ではワーカー能力を表す Raykar らによる Two-coins モデルに従って次の (a) のようにラベルを生成する.

(a) ワーカー j が Two-coins モデルより, 真のラベル y_i が 1 のとき, $Pr[y_{ij} = 1 | y_i = 1] \geq \alpha_j$ ならば事例 \mathbf{x}_i に $y_{ij} = 1$ を付け, 真のラベル y_i が 0 のとき, $Pr[y_{ij} = 0 | y_i = 0] \geq \beta_j$ ならば $y_{ij} = 0$ を付ける.

4.1 十分なデータがある場合の提案手法の性能評価

提案手法が十分な時間の経過により Kajino らの手法と同程度の性能を持つことを示す.

データセットは人工データである. 一様分布 $\mathcal{U}([-20, 20] \times [-20, 20])$ から事例を生成する. 事例 \mathbf{x}_i に対して, ロジスティック回帰モデルを用いて $Pr[y_i = 1 | \mathbf{x}_i, \mathbf{w}_0] \geq \gamma (\gamma = 0.5)$ ならば真のラベル $y_i = 1$ を割り当て, それ以外ならば $y_i = 0$ を割り当てる. 上記の方法 (a) により各ワーカーが事例にラベルを付ける.

事例数を固定し, ワーカー数を 10 人ずつ, あるいはワーカー数を固定し, 事例数を 10 個ずつ変えながら, 実験を行う. 実験毎に 100 回行って, AUC の平均で識別器の性能を評価する.

図 1 の図より, 提案手法は Kajino らの手法との差は見られなく, 他の手法を上回ったと確認できた. これは, 提案手法は Kajino らの手法をオンライン化した手法であり, かつ十分なデータがあるからである.

4.2 タスクの傾向が変化する際の各手法の性能評価

データセットは Pendigits である. このデータは数字の画像を表す分類問題のベクトルデータである. 実験に用いるために傾向が変化するタスクを作る. タスクは 0 と 1 を分類し, $\{0, 1\}$ のように記述する. 同様に $\{0, 1\}, \{2, 3\}, \{4, 5\}, \{6, 7\}, \{8, 9\}$ の 5 つのタスクを生成する. 残る 5 つのタスクは最初の 5 つのタスクより生成される. 例えば, 元々 $\{4, 5\}$ の分類のためのデータセットだが 4 を 5 に, 5 を 4 に修正すれば $\{5, 4\}$ タスクが生成できる. ただし, 実験で扱うのは二値分類問題なので $\{2 \rightarrow 0, 3 \rightarrow 1\}, \{4 \rightarrow 0, 5 \rightarrow 1\}, \{6 \rightarrow 0, 7 \rightarrow 1\}, \{8 \rightarrow 0, 9 \rightarrow 1\}$ のように

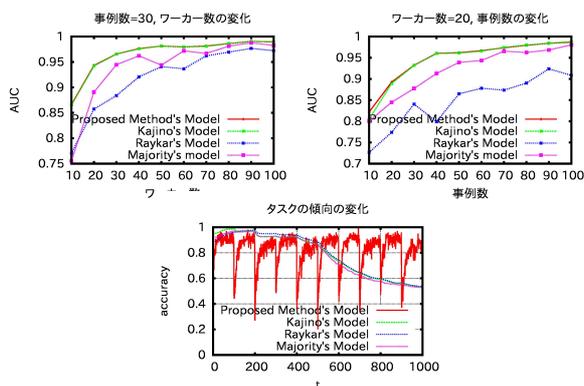


図 2: Pendigits データ:タスクの傾向の変化

変化させて $\{0, 1\}$ に修正する. そして, ワーカー数が 20 人である. すべてのワーカーが同じ能力を持つとする. 上記の方法 (a) より各ワーカーが事例にラベルを付けるシミュレーションを行う.

ある時刻 $t (t \in \{0, \dots, T\})$ に一つの事例とそれに対するラベルが入力され, 時刻 $T = 1000$ まで事例が入力されるとする. 100 事例ごとに上記のタスクの中からタスクの傾向が変化する. 実験毎に 100 回実験を行い, Accuracy の平均で識別器の性能を評価する.

図 2 の図より多数決に基づく手法, Raykar らの手法と Kajino らの手法は時間の経過につれて性能が低下し, accuracy が 0.5 (ランダム) に近づくことが確認された. 既存手法はすべての事例が同一の傾向に従うことを仮定し設計されているため, 様々なタスクの傾向が入力された状況に対応できていない. 一方, 提案手法は 100 事例毎に傾向が変化するこの設定において同一傾向内では傾向が変化した直後の時点で性能は低下するが, 直ちに性能が改善し, 他の手法を上回ることが確認された. オンライン凸計画では一つの事例ごとに学習するためデメリットとして最初は性能は高くないが, 一定数の事例を学習した後は, 提案手法の性能は改善する. その結果, 平均としては, 提案手法は動的な問題に対応できていることがわかる.

4.3 ワーカーの信頼度が変化する際の各手法の性能評価

タスクの傾向が変化する際の実験と同様である. ただし, ワーカー数が 9 人であり, それぞれ違う能力を持つ.

ある時刻 t に一つの事例とそれに対するラベルが入力され, 時刻 $T = 1000$ まで事例が入力されるとする. 100 事例ごとに傾向が変化する. ワーカーの中では 3 人は能力 $\alpha_j = \beta_j$ が $t = [300, 600)$ の間で 0.85 であり, 他の時点で 0.5 であるとし, これらのワーカーを Worker 1 と呼ぶ. また, 他の 3 人は能力 $\alpha_j = \beta_j$ が 0.5 であり, 常にランダムにラベルを付けるとし, これらのワーカーを Worker 2 と呼ぶ. 最後の 3 人は能力 $\alpha_j = \beta_j$ が $t = [0, 300)$ の間で 0.85 であり, 他の時点で 0.5 であるとし, これらのワーカーを Worker 3 と呼ぶ. 実験毎に 100 回を行い, Accuracy の平均で識別器の性能を評価する.

図 3 より Worker 1 の性能を表す右上の図, Worker 2 の性能を表す左下の図, Worker 3 の性能を表す右下の図は実験設定と比べて比較的に一致すると分かった. そして, 真のモデルの性能を表す左上の図より提案手法のモデルあるいは真のモデルの性能はワーカーのモデルの性能と比較すると $t = [0, 600)$ の間ではワーカーの能力の変化と関係せずに能力の高いワーカーの出力を重視し, そのワーカーの性能を保つことがわかる. 従って, 提案手法はワーカーの信頼度が動的に変化する場合でも能力の高いワーカーの予測能力を出力に反映できることが確認できた.

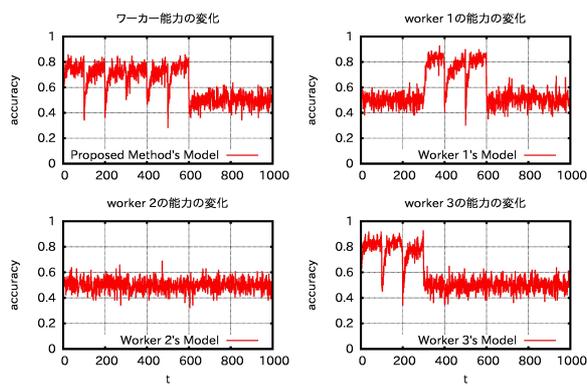


図 3: Pendigits データ:ワーカー能力の変化

5. おわりに

これまで, クラウドソーシングにより信頼度の低いラベルが得られたデータを取り扱っているクラウドソーシングにおける分類問題を解決する既存手法はタスクの傾向やワーカーの信頼度は変わらないと想定されてきた. 本論文では, 時間とともにタスクの傾向が変わると仮定し, 人工データとベンチマークデータで実験を行い, 既存手法がその状況に対応できないことを指摘した. そのため, 凸最適化に基づく Kajino らが提案した統合識別モデルをオンライン凸計画を使ってオンライン化した手法を提案した. オンライン凸計画のメリットよりその問題を解決できたことが実験を通して確認された. また, 提案手法はワーカーの信頼度が動的に変化する場合でも能力の高いワーカーの予測能力を出力に反映できることが確認できた.

謝辞

本研究は, 最先端研究開発プログラム「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」の助成を受けました.

参考文献

- [1] 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)*, Vol. 28, No. 1, pp. pp. 20–28, 1979.
- [2] Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Anna Jerebko, Charles Florin, Gerardo Hermosillo Valadez, Luca Bogoni, and Linda Moy. Supervised learning from multiple experts: whom to trust when everyone lies a bit. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09*, pp. 889–896, New York, NY, USA, 2009. ACM.
- [3] Hiroshi KAJINO and Hisashi KASHIMA. A convex formulation of learning from crowds. 電子情報通信学会技術研究報告. IBISML, 情報論的学習理論と機械学習, Vol. 111, No. 275, pp. 231–236, 2011.
- [4] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In Tom Fawcett and Nina Mishra, editors, *The Proceedings of the Twentieth International Conference on Machine Learning*, pp. 928–936. AAAI Press, 2003.