Regression with Sensor Data Containing Incomplete Observations

Takayuki Katsuki [1] **Takayuki Osogami** [1]

Abstract

This paper addresses a regression problem in which output label values are the results of sensing the magnitude of a phenomenon. A low value of such labels can mean either that the actual magnitude of the phenomenon was low or that the sensor made an incomplete observation. This leads to a bias toward lower values in labels and the resultant learning because labels may have lower values due to incomplete observations, even if the actual magnitude of the phenomenon was high. Moreover, because an incomplete observation does not provide any tags indicating incompleteness, we cannot eliminate or impute them. To address this issue, we propose a learning algorithm that explicitly models incomplete observations corrupted with an asymmetric noise that always has a negative value. We show that our algorithm is unbiased as if it were learned from uncorrupted data that does not involve incomplete observations. We demonstrate the advantages of our algorithm through numerical experiments.

1. Introduction

This paper addresses a regression problem for predicting the magnitude of a phenomenon when an observed magnitude involves a particular measurement error. The magnitude typically represents *how large a phenomenon is or how strong the nature of the phenomenon is*. Such examples of predicting the magnitude are found in several application areas, including pressure, vibration, and temperature (Vandal et al., 2017; Shi et al., 2017; Wilby et al., 2004; Tanaka et al., 2019). In medicine and healthcare, the magnitude may represent pulsation, respiration, or body movements (Inan et al., 2009; Nukaya et al., 2010; Lee et al., 2016; Alaziz et al., 2016; 2017; Carlson et al., 2018).

More specifically, we learn a regression function to predict the *label* representing the magnitude of a phenomenon from *explanatory variables*, where the label is observed with a sensor and is not necessarily in agreement with the actual magnitude. We use the term "label" even though we address the regression problem; a label is thus real-valued in this paper.

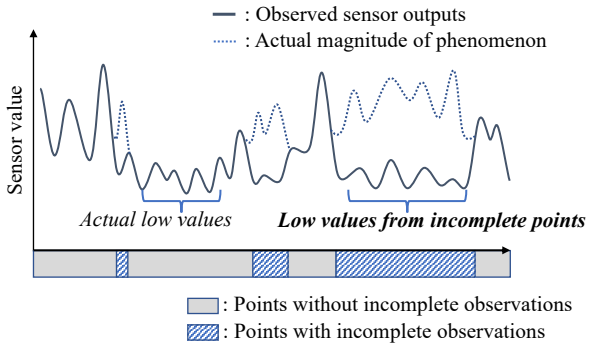


Figure 1. Sensor values with incomplete observations. Low values can mean either actual low magnitude or incomplete observation.

For example, the body movements of a patient are typically measured with an intrusive sensor attached to the chest or wrist. If we can learn a reliable regression function that predicts the magnitude of the body movements (label) from the values of non-intrusive bed sensors (explanatory variables) (Mullaney et al., 1980; Webster et al., 1982; Cole et al., 1992; Tryon, 2013), we can replace intrusive sensors with non-intrusive ones, which will reduce the burden on patients.

Although the sensors that measure the label generally have high accuracy, they often make incomplete observations, and *such incomplete observations are recorded as low values instead of missing values*. This leads to the particular challenge illustrated in Fig. 1, where a low value of the label can mean either that the actual magnitude of the phenomenon was low or that the sensor made an incomplete observation, and there are no clues to distinguish between the two.

Such incomplete observations are prevalent when measuring the magnitude of a phenomenon. For example, the phenomenon may be outside the coverage of a sensor, or

[1]IBM Research - Tokyo, Tokyo, Japan. Correspondence to: Takayuki Katsuki <kats@jp.ibm.com>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

the sensing system may experience temporary mechanical failures. In the case of body movements, the sensor may be temporarily detached from the chest or wrist. In all cases, the sensor keeps recording low values, even though the actual magnitude may be high, and no tag indicating incompleteness is provided.

This incomplete observation is particularly severe in the sensor for the label, not for the explanatory variables. The sensor for the label is often single-source and has narrow data coverage because it is intrusive or because it is expensive to produce highly accurate observations. For example, chest or wrist sensors focus on the movements of a local body part with high accuracy and often miss movements outside their coverage, such as those of parts located far from where the sensor is attached. At most, a single intrusive sensor can be attached to avoid burdening the patient. In contrast, the sensors for explanatory variables are usually multi-source and provide broader data coverage: for example, multiple sensors can be attached on various places of a bed and globally monitor the movements of all body parts, albeit with low accuracy.

One cannot simply ignore the problem that the observations of labels may be incomplete because estimated regression functions naively trained on such data would be severely biased toward lower values regardless of the amount of training data. This bias comes from the fact that incomplete observations always have lower values than the actual magnitude, and they frequently occur on label sensors, whereas explanatory variables are observed completely.

Unfortunately, since we cannot identify which observations are incomplete, we cannot eliminate or impute them by using existing methods that require the identification of incomplete observations. Such methods include thresholding, missing value detection (Pearson, 2006; Qahtan et al., 2018), imputation (Enders, 2010; Smieja et al., 2018; Ma & Chen, 2019; Sportisse et al., 2020), and semi-supervised regression (Zhou & Li, 2005; Zhu & Goldberg, 2009; Jean et al., 2018; Zhou et al., 2019).

The problems of incomplete observations also cannot be solved with robust regression (Huber et al., 1964; Narula & Wellington, 1982; Draper & Smith, 1998; Wilcox, 1997), which takes into account the possibility that the observed labels contain outliers. While robust regression is an established approach and state-of-the-art for corrupted labels in regression, it assumes that the noise is unbiased. Since incomplete observations induce noise that is severely biased toward lower values, robust regression methods still produce regression functions that are biased toward lower values when the fraction of incomplete observations exceeds its tolerance.

In this paper, to mitigate the bias toward lower values, we explicitly assume the existence of asymmetric noise in labels from incomplete observations, which always has a negative value, in addition to the ordinary symmetric noise, i.e., *asymmetric label corruption*. We refer to data with the asymmetric label corruption as *asymmetrically corrupted data*. We then formulate a regression problem from our asymmetrically corrupted data and design a principled learning algorithm.

By explicitly modeling the asymmetric label corruption, we derive a learning algorithm that has a rather bold feature: it ignores the labels that have relatively low values (lower-side labeled data). In other words, our algorithm utilizes the data whose labels have relatively high values (upper-side labeled data) and the data whose labels are ignored (unlabeled data). Hence, we refer to our algorithm as *upper and unlabeled* regression (U2 regression). This aligns with the intuition that the labels with low values are particularly unreliable, as those low values may be due to incomplete observations.

Our main result is that U2 regression produces a regression function that is, under some technical assumptions, unbiased and consistent with the one that could be produced from uncorrupted data (i.e., without incomplete observations). This counterintuitive result is achieved by considering a specific class of loss functions and deriving their gradient as a form requiring only upper-side labeled data and unlabeled data. The gradient can be computed with only the reliable parts in asymmetrically corrupted data. We prove that this gradient is asymptotically equivalent to the gradient that is computed with the uncorrupted data. The main novelty of our approach is thus in the loss function, and we will empirically demonstrate the effectiveness of the proposed class of loss functions over common existing loss functions in dealing with asymmetrically corrupted data in synthetic and six real-world regression tasks.

Contributions.

- We formulate a novel problem of learning a regression function for a sensor magnitude with asymmetrically corrupted data. This is vital for applications where the sensor is susceptible to unidentifiable incomplete observations.
- We derive an unbiased and consistent learning algorithm (U2 regression) for this problem with the new class of loss functions.
- Extensive experiments on synthetic and six real-world regression tasks including a real use case for healthcare demonstrate the effectiveness of the proposed method.

2. Regression from Asymmetrically Corrupted Data

Our goal is to derive a learning algorithm with asymmetrically corrupted data (due to incomplete observations) in a manner that is unbiased and consistent with the one that uses uncorrupted data (without involving incomplete observations). In Section 2.1, we examine the regression problem that uses the uncorrupted data, and in Section 2.2, we formulate learning from the asymmetrically corrupted data.

2.1. Regression Problem from Uncorrupted Data

Let $\boldsymbol{x} \in \mathbb{R}^D (D \in \mathbb{N})$ be a D-dimensional explanatory variable and $y \in \mathbb{R}$ be a real-valued label. We assume that, without incomplete observations, y is observed in accordance with

$$y = f^*(\boldsymbol{x}) + \epsilon_{\mathrm{s}}, \tag{1}$$

where f^* is the oracle regressor and ϵ_{s} is the symmetric noise with 0 as the center, such as additive white Gaussian noise (AWGN).

We learn a regression function $f(\boldsymbol{x})$ that computes the value of the estimation of a label $\hat{y}$ for a newly observed $\boldsymbol{x}$ as $\hat{y} = f(\boldsymbol{x})$. The optimal regression function $\hat{f}$ is given by

$$\hat{f} \equiv \operatorname*{argmin}_{f \in \mathcal{F}} \mathcal{L}(f), \tag{2}$$

where $\mathcal{F}$ is a hypothesis space for f, and $\mathcal{L}(f)$ is the expected loss when the regression function $f(\boldsymbol{x})$ is applied to data $(\boldsymbol{x}, y)$, distributed in accordance with an underlying distribution $p(\boldsymbol{x}, y)$:

$$\mathcal{L}(f) \equiv \mathbb{E}_{p(\boldsymbol{x},y)}[L(f(\boldsymbol{x}), y)], \tag{3}$$

where $\mathbb{E}_p[\bullet]$ denotes the expectation over the distribution p, and $L(f(\boldsymbol{x}), y)$ is the loss function between $f(\boldsymbol{x})$ and y, e.g., the squared loss, $L(f(\boldsymbol{x}), y) = \|f(\boldsymbol{x}) - y\|^2$. The expectation $\mathbb{E}_{p(\boldsymbol{x},y)}$ can be estimated by computing a sample average for the training data $\boldsymbol{\mathcal{D}} \equiv \{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$, which is N pairs of explanatory variables and labels.

2.2. Regression Problem from Asymmetrically Corrupted Data

In this paper, we consider a scenario in which we only have access to the asymmetrically corrupted data $\boldsymbol{\mathcal{D}}' \equiv \{(\boldsymbol{x}_n, y'_n)\}_{n=1}^N$, where a label y' may be corrupted due to incomplete observations. A corrupted label y' is observed from the uncorrupted y with an asymmetric negative-valued noise, ϵ_{a}:

$$y' = y + \epsilon_{\mathrm{a}}, \tag{4}$$

where ϵ_{a} always has a random negative value, which implies $y' \leq y$.

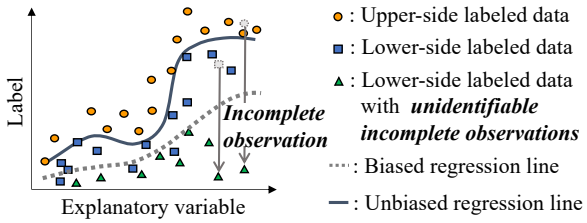


Figure 2. Asymmetrically corrupted data. Labels for incomplete observations, depicted as triangles, become lower than those of typical observations, depicted as circles or squares.

We seek to learn a regression function $f(\boldsymbol{x})$ *in an unbiased and consistent manner* (i.e., to find the solution for Eq. (2)) by using only $\boldsymbol{\mathcal{D}}'$. Although AWGN can be handled even when we use a naive regression method such as least squares, the asymmetric noise ϵ_{a}, which always has a negative value, is problematic.

Intuitively, the asymmetric noise ϵ_{a} makes *lower-side labeled data* particularly unreliable and inappropriate for learning while keeping *upper-side labeled data* reliable, where the upper-side labeled data refers to the data $\{(\boldsymbol{x}, y)\}$ whose label is above the regression line (i.e., $f(\boldsymbol{x}) \leq y$) and the lower-side labeled data refers to the data whose label is below the regression line (i.e., $y < f(\boldsymbol{x})$). The regression line represents the estimation of a regression function. Figure 2 illustrates this as a scatter plot of the value of the label against the value of an explanatory variable. Here, the data with incomplete observations appear only in the lower side of the regression line because ϵ_{a} makes observations have lower label values than those of typical observations, where the regression line represents such typical observations. This asymmetry leads to biased learning compared to learning from the uncorrupted data without incomplete observations.

To address the asymmetric noise ϵ_{a} and its resultant bias, we formalize the assumption on the observation process for the asymmetrically corrupted data $\boldsymbol{\mathcal{D}}'$ and derive a lemma representing the nature of $\boldsymbol{\mathcal{D}}'$. Then, we propose a learning algorithm based on the lemma in the next section.

We assume that $\boldsymbol{\mathcal{D}}'$ has enough information to estimate f and that the asymmetric noise ϵ_{a} is significant enough compared to the symmetric noise ϵ_{s}, which are necessary assumptions to make the learning problem solvable. Formally, the observation processes of $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{D}}'$ are characterized as follows.

Assumption 2.1. Assume $\epsilon_{\mathrm{s}} \perp\!\!\!\perp f^*(\boldsymbol{x})$, $\mathbb{E}_{p(\epsilon_{\mathrm{s}})}[\epsilon_{\mathrm{s}}] = 0$; $\epsilon_{\mathrm{a}} \perp\!\!\!\perp f^*(\boldsymbol{x})$, $\epsilon_{\mathrm{a}} \leq 0$ almost surely (a.s.); $2|\epsilon_{\mathrm{s}}| < |\epsilon_{\mathrm{a}}|$ a.s. when $\epsilon_{\mathrm{a}} < 0$; and $\{(\boldsymbol{x}_n, y_n, y'_n)\}_{n=1}^N$ are i.i.d. observations in accordance with Eqs. (1) and (4).

We then have the following lemma, which shows that ϵ_{a}

does not change the expectation for our upper-side labeled data ($f(\boldsymbol{x}) \leq y'$) before and after adding ϵ_{a}.

Lemma 2.2. *Let $\mathcal{F}' \equiv \{f \in \mathcal{F} : |f(\boldsymbol{x}) - f^*(\boldsymbol{x})| \leq |\epsilon_{\mathrm{s}}| \text{ a.s.}\}$. When $f \in \mathcal{F}'$, the following holds under Assumption 2.1:*

$$\mathbb{E}_{p(\boldsymbol{x},y'\|f(\boldsymbol{x})\leq y')}[G(\boldsymbol{x},y')] = \mathbb{E}_{p(\boldsymbol{x},y|f(\boldsymbol{x})\leq y)}[G(\boldsymbol{x},y)] \quad (5)$$

for any function $G : \mathbb{R}^D \times \mathbb{R} \to \mathbb{R}$ as long as the expectations exist.

Proof. We outline a proof here and provide a complete one in Appendix A.1. We first show that ϵ_{a} does not change the distribution for upper-side labeled data ($f^*(\boldsymbol{x}) \leq y'$) on the basis of the oracle regression function f^* before and after adding ϵ_{a}, i.e., $\epsilon_{\mathrm{a}} = 0$ when $f^*(\boldsymbol{x}) \leq y'$. With the condition $f \in \mathcal{F}'$, we can further prove that $\epsilon_{\mathrm{a}} = 0$ when $f(\boldsymbol{x}) \leq y'$, which is for upper-side labeled data on the basis of f. This establishes $p(x, y'|f(\boldsymbol{x}) \leq y') = p(x, y|f(\boldsymbol{x}) \leq y)$ and implies Lemma 2.2. □

The condition parts of the conditional distributions in Eq. (5) represent the relationships between labels and the estimations of the regression function f, e.g., $p(\boldsymbol{x}, y|f(\boldsymbol{x}) \leq y)$ is the distribution of x and y when y is higher than what is given by f. The condition $f \in \mathcal{F}'$ represents our natural expectation that f well approximates f^*.

Lemma 2.2 shows that our upper-side labeled data ($f(\boldsymbol{x}) \leq y'$) is still reliable for regression. In the next section, we derive an unbiased learning algorithm based on this lemma.

3. U2 Regression

We seek to find the minimizer of the objective $\mathcal{L}(f)$ in Eq. (2) *from the asymmetrically corrupted data* $\boldsymbol{\mathcal{D}}'$. To this end, we propose a gradient that relies only on the knowledge of the distribution of the corrupted data $p(\boldsymbol{x}, y')$ but is still equivalent to the gradient of $\mathcal{L}(f)$, which depends on the distribution of the uncorrupted data $p(\boldsymbol{x}, y)$. Specifically, based on Lemma 2.2, we will rewrite the gradient based on $p(\boldsymbol{x}, y)$ into one that only requires $p(\boldsymbol{x}, y')$.

3.1. Gradient for Regression from Asymmetrically Corrupted Data

Here, we minimize $\mathcal{L}(f)$ with the gradient descent. At step $t+1$ in the gradient descent, the gradient of $\mathcal{L}(f)$ with respect to the parameters $\boldsymbol{\theta}$ of f is represented with a regression function, f_t, which is estimated at step t, as follows:

$$\nabla\mathcal{L}(f_t) \equiv \mathbb{E}_{p(\boldsymbol{x},y)}[\nabla L(f_t(\boldsymbol{x}), y)], \quad (6)$$
$$\text{where} \quad \nabla L(f_t(\boldsymbol{x}), y) \equiv \frac{\partial L(f(\boldsymbol{x}), y)}{\partial \boldsymbol{\theta}}\Big|_{f=f_t}.$$

Note that this holds for any step in the gradient descent. When $t = 0$, f_0 is the initial value of f, and when $t = \infty$, we suppose $f_\infty = \hat{f}$. We can decompose $\nabla\mathcal{L}(f_t)$ as

$$\nabla\mathcal{L}(f_t) = p(f_t(\boldsymbol{x}) \leq y)\mathbb{E}_{p(\boldsymbol{x},y|f_t(\boldsymbol{x})\leq y)}[\nabla L(f_t(\boldsymbol{x}), y)] + p(y < f_t(\boldsymbol{x}))\mathbb{E}_{p(\boldsymbol{x},y|y<f_t(\boldsymbol{x}))}[\nabla L(f_t(\boldsymbol{x}), y)]. \quad (7)$$

We will introduce and use the class of loss functions whose gradient can depend on $f(\boldsymbol{x})$ but not on y when $y < f(\boldsymbol{x})$; thus, when $y < f(\boldsymbol{x})$, we write $\nabla L(f(\boldsymbol{x}), y)$ as $\boldsymbol{g}(f(\boldsymbol{x}))$ to emphasize this independence. Formally,

Condition 3.1. For $y < f(\boldsymbol{x})$, let $\boldsymbol{g}(f(\boldsymbol{x}))$ be defined as $\nabla L(f(\boldsymbol{x}), y)$. Then $\boldsymbol{g}(f(\boldsymbol{x}))$ depends only on $f(\boldsymbol{x})$ and is conditionally independent of y given $f(\boldsymbol{x})$.

Such common losses that satisfy Condition 3.1 include the absolute loss and pinball loss, which are respectively used in least absolute regression and quantile regression (Lee et al., 2016; Yeung et al., 2002; Wang et al., 2005; Srinivas et al., 2020). For example, the gradient of the absolute loss is

$$\frac{\partial|f(\boldsymbol{x}) - y|}{\partial\boldsymbol{\theta}} = \frac{\partial f(\boldsymbol{x})}{\partial\boldsymbol{\theta}} \qquad \text{when } y < f(\boldsymbol{x}), \quad (8)$$

which does not depend on the value of y but only on $f(\boldsymbol{x})$.

The class of loss functions satisfying Condition 3.1 is broad, since Condition 3.1 only specifies the loss function for lower-side labeled data ($y < f(x)$). The upper-side loss can be an arbitrary function. In our experiments, we will actually use squared loss for upper-side labeled data ($f(x) \leq y$), which violates Condition 3.1, and absolute loss for lower-side labeled data ($y < f(x)$).

We now propose a gradient that does not rely on the knowledge of $p(\boldsymbol{x}, y)$ but instead uses only $p(\boldsymbol{x}, y')$. Namely,

$$\nabla\tilde{\mathcal{L}}(f_t) \equiv p(f_t(\boldsymbol{x}) \leq y)\mathbb{E}_{p(\boldsymbol{x},y'|f_t(\boldsymbol{x})\leq y')}\Big[\nabla L(f_t(\boldsymbol{x}), y)\Big] + \mathbb{E}_{p(\boldsymbol{x})}\Big[\boldsymbol{g}(f_t(\boldsymbol{x}))\Big] - p(f_t(\boldsymbol{x}) \leq y)\mathbb{E}_{p(\boldsymbol{x}|f_t(\boldsymbol{x})\leq y')}\Big[\boldsymbol{g}(f_t(\boldsymbol{x}))\Big]. \quad (9)$$

In Section 3.2, we will formally establish the equivalence between the gradient in Eq. (9) and that in Eq. (6) under our assumptions. Note that, in the second and third terms of Eq. (9), we apply expectations over $p(\boldsymbol{x})$ and $p(\boldsymbol{x}|f_t(\boldsymbol{x}) \leq y')$ to $\boldsymbol{g}(f(\boldsymbol{x}))$, even though $\boldsymbol{g}(f(\boldsymbol{x}))$ is defined only for $y < f(\boldsymbol{x})$. This is tractable due to the nature of $\boldsymbol{g}(f(\boldsymbol{x}))$, which does not depend on the value of y.

Since the expectations in Eq. (9) only depend on $\boldsymbol{x}$ and y', they can be estimated by computing a sample average for

our asymmetrically corrupted data $\boldsymbol{\mathcal{D}}'$ as

$$\nabla\hat{\mathcal{L}}(f_t) = \frac{\pi_{\mathrm{up}}}{n_{\mathrm{up}}}\Bigg[\sum_{(\boldsymbol{x},y)\in\{\boldsymbol{X}_{\mathrm{up}},\boldsymbol{y}'_{\mathrm{up}}\}} \nabla L(f_t(\boldsymbol{x}),y)\Bigg] + \frac{1}{N}\Bigg[\sum_{\boldsymbol{x}\in\boldsymbol{X}_{\mathrm{un}}} \boldsymbol{g}(f_t(\boldsymbol{x}))\Bigg] - \frac{\pi_{\mathrm{up}}}{n_{\mathrm{up}}}\Bigg[\sum_{\boldsymbol{x}\in\boldsymbol{X}_{\mathrm{up}}} \boldsymbol{g}(f_t(\boldsymbol{x}))\Bigg], \tag{10}$$

where $\{\boldsymbol{X}_{\mathrm{up}}, \boldsymbol{y}'_{\mathrm{up}}\}$ represents the set of coupled pairs of $\boldsymbol{x}$ and y' in the upper-side labeled sample set, $\{x, y' : f_t(\boldsymbol{x}) \leq y'\}$, in $\boldsymbol{\mathcal{D}}'$; $\boldsymbol{X}_{\mathrm{un}}$ is a sample set of $\boldsymbol{x}$ in $\boldsymbol{\mathcal{D}}'$ ignoring labels y'; n_{up} is the number of samples in the upper-side labeled set; and π_{up} is $\pi_{\mathrm{up}} \equiv p(f_t(\boldsymbol{x}) \leq y)$. Note that π_{up} depends on the current estimation of the function f_t and the label y with complete observation. Thus, it changes at each step of the gradient descent, and we cannot determine its value in a general way. In this paper, we propose treating π_{up} as a hyperparameter and optimize it with the grid search based on the validation set. In our experiments, we will demonstrate that this simple approach can work effectively and robustly in practice.

As we will show in Section 3.2, we can use Eq. (10) to design an algorithm that gives an unbiased and consistent regression function. By using the gradient in Eq. (10), we can optimize Eq. (2) and learn the regression function with only upper-side labeled samples and unlabeled samples from $\boldsymbol{\mathcal{D}}'$ independent of lower-side labels. This addresses the problem that our lower-side labeled data is particularly unreliable and leads to overcoming the bias that stems from this unreliable part of the data. We refer to our algorithm as *upper and unlabeled* regression (U2 regression).

See Appendix B for the specific implementation of the algorithm based on stochastic optimization. The gradient in Eq. (10) can be interpreted in an intuitive manner. The first term has the effect of minimizing the upper-side loss. Recall that the upper-side data are not affected by the asymmetric noise under our assumptions from Lemma 2.2. Thus, U2 regression seeks to learn the regression function f on the basis of this reliable upper-side data. Note that the first term becomes zero when all of the data points are below f (i.e., $y' \leq f_t(\boldsymbol{x}), \forall(x, y') \in \mathcal{D}'$), since $\{\boldsymbol{X}_{\mathrm{up}}, \boldsymbol{y}'_{\mathrm{up}}\}$ then becomes empty. The second term thus has the effect of pushing down f at all of the data points so that some data points are above f. Meanwhile, the third term partially cancels out this effect of the second term for the upper-side data to control the balance between the first and second terms.

3.2. Unbiasedness and Consistency of Gradient

U2 regression is the learning algorithm based on the gradient $\nabla\hat{\mathcal{L}}(f_t)$ in Eq. (10) and uses only asymmetrically corrupted data $\boldsymbol{\mathcal{D}}'$. The use of $\nabla\hat{\mathcal{L}}(f_t)$ can be justified as follows:

Proposition 3.2. *Suppose that Assumption 2.1 holds and the loss function $L(f(\boldsymbol{x}), y)$ satisfies Condition 3.1. Then, the gradient $\nabla\tilde{\mathcal{L}}(f_t)$ in Eq. (9) and its empirical approximation $\nabla\hat{\mathcal{L}}(f_t)$ in Eq. (10) are unbiased and consistent with the gradient $\nabla\mathcal{L}(f_t)$ in Eq. (6) a.s.*

Proof. We outline a proof here and provide a complete one in Appendix A.2. First, we rewrite the decomposed gradient $\nabla\mathcal{L}(f_t)$ in Eq. (7) into a gradient that only contains the expectations over $p(\boldsymbol{x}, y|f_t(\boldsymbol{x}) \leq y)$ and $p(\boldsymbol{x})$ with Condition 3.1. Then, we apply Lemma 2.2 to the gradient, and it becomes identical to Eq. (9). □

In other words, U2 regression asymptotically produces the same result as the learning algorithm based on the gradient $\nabla\mathcal{L}(f_t)$ in Eq. (6), which requires the uncorrupted data without incomplete observations, $\boldsymbol{\mathcal{D}}$. The convergence rate of U2 regression is of the order $\mathcal{O}_p(1/\sqrt{n_{\mathrm{up}}} + 1/\sqrt{N})$ in accordance with the central limit theorem (Chung, 1968), where $\mathcal{O}_p$ denotes the order in probability.

We further justify our approach of having the specific form of Eq. (9) by showing how a straightforward variant that uses $\boldsymbol{\mathcal{D}}'$ as if it does not involve incomplete observations (i.e., $p(\boldsymbol{x}, y) \approx p(\boldsymbol{x}, y')$) can fail for our problem. To this end, we introduce an additional assumption on the observation process:

Assumption 3.3. Assume $\epsilon_{\mathrm{a}} \perp\!\!\!\perp \boldsymbol{x}$.

Then, we have

Lemma 3.4. *Let $\nabla\check{\mathcal{L}}(f_t)$ be a variant of the gradient in Eq. (7) replacing $p(\boldsymbol{x}, y)$ with $p(\boldsymbol{x}, y')$, δ be the difference between the expectations of the gradients in the upper side and the lower side $\delta \equiv \left|\mathbb{E}_{p(\boldsymbol{x},y|f(\boldsymbol{x})\leq y)}[\nabla L(f(\boldsymbol{x}), y)] - \mathbb{E}_{p(\boldsymbol{x},y|y<f(\boldsymbol{x}))}[\nabla L(f(\boldsymbol{x}), y)]\right|$, $\eta \in [0, 1]$ be the probability of being $0 \leq \epsilon_{\mathrm{s}}$, and $\xi \in [0, 1]$ be the probability of $\epsilon_{\mathrm{a}} = 0$. Then, $\nabla\check{\mathcal{L}}(f_t)$ is not consistent with the gradient $\nabla\mathcal{L}(f_t)$ in Eq. (6) a.s., and the difference (bias) between them at step $t + 1$ in the gradient descent is*

$$\frac{\eta(1-\eta)(1-\xi)}{1-\eta\xi}\delta \leq |\nabla\check{\mathcal{L}}(f_t) - \nabla\mathcal{L}(f_t)|. \tag{11}$$

Proof. We outline a proof here and provide a complete one in Appendix A.3. We first show that the bias $|\nabla\check{\mathcal{L}}(f_t) - \nabla\mathcal{L}(f_t)|$ can be represented by the difference between the expectation of $\boldsymbol{g}(f_t(\boldsymbol{x}))$ with the upper-side data and that with the lower-side data, which can be written by δ. The bias also has a coefficient that contains the proportions for the lower-side data and the original upper-side data mixed into the lower side due to incomplete observations. These values can be written by η and ξ from their definitions. □

Lemma 3.4 shows that the bias caused by the asymmetric label corruption becomes severe when there is a large difference between the expectations of the gradients in the upper

side and the lower side. δ is usually higher than zero because $\delta = 0$ implies that there is no difference between the expectations of the gradients in the upper and lower sides or that both of the expectations are zero. Furthermore, a larger $1 - \xi = p(\epsilon_{\mathrm{a}} < 0)$ makes the bias more significant, which agrees with the intuition that as the proportion of incomplete observations increases, the problem becomes more difficult.

4. Experiments

We now evaluate the proposed method through numerical experiments. We first introduce the baselines to be compared and then present the experimental results to demonstrate the effectiveness of our unbiased learning.

4.1. Baselines

Recall that the novelty of the proposed approach lies in the unbiased gradient in Eq. (10), which is derived from the new class of loss functions in Eq. (9) with Condition 3.1. The objective of our experiments is thus to validate the effectiveness of this new class of loss functions and the corresponding gradients against common loss functions in the literature. Specifically, we compare the proposed method with mean squared error (MSE), mean absolute error (MAE), and Huber losses (Huber et al., 1964; Narula & Wellington, 1982; Wilcox, 1997). In terms of robust loss functions in regression, MAE and Huber losses are considered the de facto standard and state-of-the-art in many studies and libraries. We use the same model and optimization method with all of the loss functions under consideration, and hence the only difference between the proposed method and the baselines is the loss functions. Since the loss function uniquely determines the baseline, we refer to each baseline method as `MSE`, `MAE`, or `Huber`.

4.2. Experimental Procedure and Results

The experiments are organized into four parts. In Section 4.2.1, we visually demonstrate the effectiveness of the proposed approach in giving unbiased prediction. In Section 4.2.2, we investigate the robustness of our validation set-based tuning of the hyperparameters. In Section 4.2.3, we intensively and quantitatively evaluate the predictive error of the proposed method and baselines with five real-world regression tasks. In Section 4.2.4, we demonstrate the practical benefit of our approach in a real healthcare use case, which forms the motivation for this work. See Appendix D-H for details of the experimental settings.

4.2.1. Demonstration of Unbiased Learning

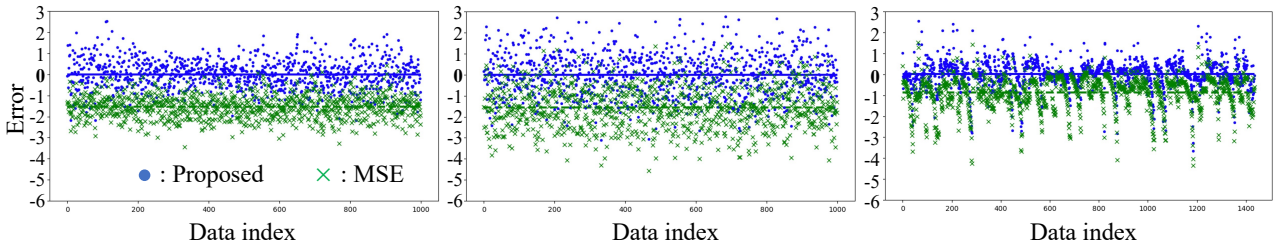
Procedure. We start by conducting the experiments with synthetic data to show the effectiveness of our method in obtaining unbiased learning results from asymmetrically corrupted data with different proportions of incomplete observations, $K = \{25, 50, 75\}\%$. We use three synthetic tasks, **LowNoise**, **HighNoise**, and **Breathing**, collected from the Kaggle dataset (Sen, 2016). We compare the proposed method against `MSE`, which assumes that both upper- and lower-side data are correctly labeled. This comparison clarifies whether our method can learn from asymmetrically corrupted data in an unbiased manner, which `MSE` cannot do. We conducted 5-fold cross-validation, each with a different randomly sampled training-testing split. For evaluation purposes, we do not include incomplete observations in these test sets. For each fold of the cross-validation, we use a randomly sampled 20% of the training set as a validation set to choose the best hyperparameters for each algorithm.

Results. Figure 3 plots the error in prediction (i.e., the predicted value minus the true value) given by the proposed method and `MSE` for each data point of the three tasks with $K = 50\%$. Since `MSE` regards both upper- and lower-side data as correctly labeled, it produces biased results due to the asymmetric label corruption, where the average error (green dashed line) is negative, which means the estimation has a negative bias. In contrast, the average error by the proposed method (blue solid line) is approximately zero. This shows that the proposed method obtained unbiased learning results. The figures for the other settings and the tables of quantitative results are provided in Appendix E.

4.2.2. Performance over different sizes of validation set

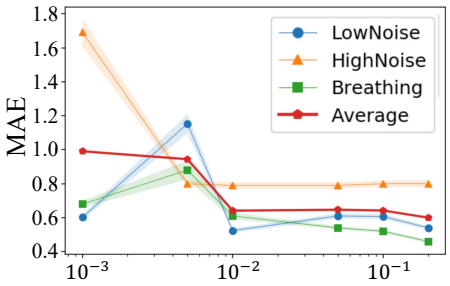
Procedure. To demonstrate the robustness of our validation set-based approach for estimating the hyperparameters, including π_{up}, we report the performance of the proposed method over different sizes of validation set. This analysis is conducted on the same tasks used in Section 4.2.1, namely, **LowNoise**, **HighNoise**, and **Breathing**, with $K = 50\%$.

Results. The results are shown in Fig. 4, where we can see that the performance of the proposed method does not degrade much even when we use only 1% of the training set as the validation set. This demonstrates that the proposed approach is robust enough for small validation sets as well as for a high proportion of incomplete validation samples, as we saw in Section 4.2.1 (e.g., $K = 50\%$ and 75%). Figure 5 shows a chart similar to the ones in Fig. 3 (the error in prediction for the **LowNoise** task with $K = 50\%$), where we used 1% of the training set as the validation set. We can see that even in this case, the proposed method achieved approximately unbiased learning (the average error shown by the blue solid line is approximately zero). The figures for the other tasks are provided in Appendix E.



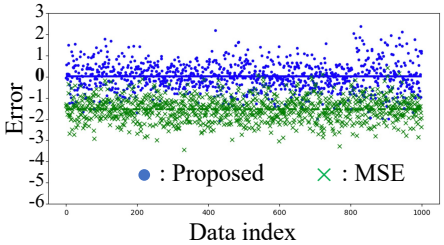
(a) LowNoise, K = 50% (b) HighNoise, K = 50% (c) Breathing, K = 50%

Figure 3. Errors (predicted value minus true value) by proposed method (blue) and by `MSE` (green) for three tasks: (a) **LowNoise**, (b) **HighNoise**, and (c) **Breathing**, with $K = 50\%$ of incomplete training samples. Error of each data point is shown by dots (for proposed method) or crosses (for `MSE`), and average error is shown by solid line (proposed method) or dashed line (`MSE`).



Size of validation set (proportion within training set)

Figure 4. Performance (MAE, lower is better) of proposed method over different sizes of validation set. Blue, orange, and green lines represent the results on **LowNoise**, **HighNoise**, and **Breathing**, respectively; red line is their average. Shaded areas are confidence intervals. Leftmost point shows results when we use 0.1% of training set as a validation set, and rightmost point shows those of 20%, which is the setting we used in all experiments throughout this paper.



LowNoise, K = 50%
Prop of val set = 1%

Figure 5. Errors in prediction for **LowNoise** task with $K = 50\%$ when we choose hyperparameters with 1% of the training set as a validation set. Other configurations are the same as those in Fig. 3.

4.2.3. Performance Comparison Among Different Loss Functions

Procedure. We next apply the proposed method and baselines to five different real-world healthcare tasks from the UCI Machine Learning Repository (Velloso, 2013; Velloso et al., 2013) to provide a more extensive comparison between the proposed method and the baselines (`MSE`, `MAE`, and `Huber`). For the proposed method, we use two implementations of $L(f(\boldsymbol{x}), y)$ for $f(\boldsymbol{x}) \leq y'$ in Eq. (10): the absolute loss (`Proposed-1`) and the squared loss (`Proposed-2`). In both implementations, we use the absolute loss, which satisfies Condition 3.1, for $L(f(\boldsymbol{x}), y)$ when $y' < f(\boldsymbol{x})$. Here, we report the *mean absolute error* (MAE), and its standard error, of the predictions $\hat{\boldsymbol{y}} = \{\hat{y}_n\}_{n=1}^N$ against the corresponding true labels $\boldsymbol{y}$ across 5-fold cross-validation, each with a different randomly sampled training-testing split. MAE is a common metric used in the healthcare domain (Lee et al., 2016; Yeung et al., 2002; Wang et al., 2005; Srinivas et al., 2020) and is defined as $\mathrm{MAE}(\hat{\boldsymbol{y}}, \boldsymbol{y}) \equiv {}^1\!/_N \sum_{n=1}^N |\hat{y}_n - y_n|$. For each fold of the cross-validation, we use a randomly sampled 20% of the training set as a validation set to choose the best hyperparameters for each algorithm, in which hyperparameters providing the highest MAE in the validation set are chosen.

Results. As shown in Table 1, `Proposed-1` and `Proposed-2` significantly outperformed the baselines. The robust regression methods (`MAE` and `Huber`) did not improve in performance against `MSE`. `Proposed-1` and `Proposed-2` respectively reduced the MAE by more than 20% and 30% on average, compared with the baselines.

4.2.4. Real Use Case for Healthcare

Procedure. Finally, we demonstrate the practical benefits of our approach in a real use case in healthcare. Here, from non-intrusive bed sensors installed under each of the four legs of a bed, we estimate the motion intensity of a subject that is measured accurately but intrusively with ActiGraph, a sensor wrapped around the wrist (Tryon, 2013; Mullaney et al., 1980; Webster et al., 1982; Cole et al., 1992). If we

Table 1. Comparison between proposed method and baselines in terms of MAE (smaller is better). Best methods are in bold. Confidence intervals are standard errors.

	Specification	Throwing A	Lifting	Lowering	Throwing B	Avg.
MSE	2.38 ± 0.03	1.54 ± 0.01	1.42 ± 0.01	1.37 ± 0.01	1.21 ± 0.01	1.58
MAE	2.14 ± 0.02	1.46 ± 0.01	1.44 ± 0.01	1.33 ± 0.01	1.31 ± 0.01	1.54
Huber	2.04 ± 0.02	1.66 ± 0.01	1.45 ± 0.01	1.50 ± 0.01	1.32 ± 0.01	1.59
Proposed-1	1.55 ± 0.02	1.18 ± 0.01	1.11 ± 0.01	1.14 ± 0.01	1.03 ± 0.01	1.20
Proposed-2	$\mathbf{1.32 \pm 0.01}$	$\mathbf{0.99 \pm 0.01}$	$\mathbf{0.94 \pm 0.01}$	$\mathbf{0.86 \pm 0.01}$	$\mathbf{0.97 \pm 0.01}$	**1.02**

Table 2. Proportion of correct prediction period and rate of false prediction in real use case for healthcare. We estimate intrusive sensor output from outputs of non-intrusive sensors.

Proportion of correct prediction period	0.89
Rate of false prediction	0.016

can mimic the outputs from ActiGraph with outputs from the bed sensors, we can measure the motion with high accuracy and high coverage, while also easing the burden on the subject. We evaluate the results with 3-fold cross-validation, where we sequentially consider data from one subject as a test set and the others as a training set. We use evaluation metrics designed for sleep-wake discrimination (Cole et al., 1992), i.e., the proportion of correct prediction period and rate of false prediction.

Results. Table 2 shows the proportion of correct prediction period and rate of false prediction, where we can see that the proposed method captured 89 percent of the total time period of the motions that were captured by ActiGraph, and false detection due to factors such as floor vibration was only 1.6 percent. Furthermore, the proposed method captured 15 additional motions that were not captured by ActiGraph. The baseline method MSE was severely underfitted, and most of the weights were zero; thus, we omitted these results. Overall, our findings here demonstrate that ActiGraph can be replaced with bed sensors, and we can also use the bed sensors for the inputs of certain ActiGraph functions, such as sleep-wake discrimination (Cole et al., 1992). See Appendix H for further details, including the actual estimation results of the motion intensity.

5. Discussion

Limitations. In this paper, we do not address ordinary outliers, where the coverage and incompleteness are consistent between a label and explanatory variables. Other established approaches can handle such cases. Only when the corruption is asymmetric does it lead to the technical challenge we address here. In that sense, we can handle the opposite asymmetric corruption, in which labels for some observations may become inconsistently *higher* than those for typical observations. This can be handled by a learning algorithm from *lower-side labeled data and unlabeled data*, i.e., LU regression. Since our derivation of U2 regression is straightforwardly applicable to this LU regression case, we show only its learning algorithm in Appendix C. Such a scenario can be found when a sensor for the label has ideal coverage but sensors for explanatory variables have smaller coverage. This may correspond to more challenging sensor replacement, which addresses critical invasiveness, cost, and lack of alternative methods. In that case, there are unidentifiable incomplete observations in explanatory variables, which leads to the opposite asymmetric data corruption.

Related problems in classification. In the classification problem setting, positive-unlabeled (PU) learning addresses a problem related to our problem setting, where it is assumed that negative data cannot be obtained, but unlabeled data are available as well as positive data (Denis, 1998; De Comité et al., 1999; Letouzey et al., 2000; Shi et al., 2018; Kato et al., 2019; Sakai & Shimizu, 2019; Li et al., 2019; Zhang et al., 2019; 2020; Chen et al., 2020b;a; Luo et al., 2021; Hu et al., 2021; Li et al., 2021; Wilton et al., 2022). An unbiased risk estimator has also been proposed (Du Plessis et al., 2014; 2015). However, PU learning cannot be used for a regression problem, where labels are real values and we need to handle order and gradation between labels. Also, these studies do not address the asymmetric noise we examine in this paper, as their motivation is rather to address the labeling cost of negative labels.

Other possible use cases. Examples of predicting the magnitude values of a sensor, which is a field of application of U2 regression, can be found in several areas. One example is estimating the wind speed or rainfall in a specific region from observable macroscopic information (Cheng & Tan, 2008; Abraham & Tan, 2010; Abraham et al., 2013; Vandal et al., 2017), known as statistical downscaling (Wilby et al., 2004). Wind speed and rainfall, which are labels in these tasks, can be sensed locally in a limited number of locations and provide incomplete observations and biased labels compared with macroscopic information, which is considered to be explanatory variables. Molecular biology

is another interesting example (Seal et al., 2020; Dizaji et al., 2021), where a low measurement of a gene expression may indicate that the gene is not present or that the sensor failed to detect it.

Future work. We showed that our approach to estimating hyperparameters based on the grid search with the validation set was effective even for the important ratio for upper-side labeled data, $p(f_t(\boldsymbol{x}) \leq y)$. It also provides the flexibility needed to handle data variation. Most studies on PU learning assume that a hyperparameter corresponding to π_{up} is given (Hammoudeh & Lowd, 2020; Sonntag et al., 2021; Lin et al., 2022), and some papers have addressed this hyperparameter estimation as their main contribution (Jain et al., 2016; Ramaswamy et al., 2016; Christoffel et al., 2016; Jain et al., 2020). Developing a method for the hyperparameter estimation to improve performance would be a worthwhile next step of our own study. Also, in Assumption 2.1, we assumed $\epsilon_{\mathrm{s}} \perp\!\!\!\perp f^*(\boldsymbol{x})$ and $\epsilon_{\mathrm{a}} \perp\!\!\!\perp f^*(\boldsymbol{x})$, which is a common noise assumption. Addressing the case where the noises are not independent of $f^*(\boldsymbol{x})$ is another possible future direction of our work.

Conclusion. We formulated a regression problem from asymmetrically corrupted data in which training data are corrupted with an asymmetric noise that always has a negative value. This causes labels for data with relatively lower label values to be particularly unreliable. To address this problem, we proposed a learning algorithm called U2 regression. Under some technical assumptions, we showed that our algorithm is unbiased and consistent with the one that uses uncorrupted data. Our analysis is based on the equivalence of the gradient between them. An experimental evaluation demonstrated that the proposed method performed significantly better than the methods without the assumption of the asymmetrical label corruption.

References

Abraham, Z. and Tan, P.-N. An integrated framework for simultaneous classification and regression of time-series data. In *SDM*, pp. 653–664, 2010.

Abraham, Z., Tan, P.-N., Perdinan, Winkler, J. A., Zhong, S., and Liszewska, M. Distribution regularized regression framework for climate modeling. In *SDM*, pp. 333–341, 2013.

Alaziz, M., Jia, Z., Liu, J., Howard, R., Chen, Y., and Zhang, Y. Motion scale: A body motion monitoring system using bed-mounted wireless load cells. In *CHASE*, pp. 183–192, 2016.

Alaziz, M., Jia, Z., Howard, R., Lin, X., and Zhang, Y. Motiontree: a tree-based in-bed body motion classification system using load-cells. In *CHASE*, pp. 127–136, 2017.

Carlson, C., Suliman, A., Alivar, A., Prakash, P., Thompson, D., Natarajan, B., and Warren, S. A pilot study of an unobtrusive bed-based sleep quality monitor for severely disabled autistic children. In *EMBC*, pp. 4343–4346, 2018.

Chen, H., Liu, F., Wang, Y., Zhao, L., and Wu, H. A variational approach for learning from positive and unlabeled data. In *NeurIPS*, 2020a.

Chen, X., Chen, W., Chen, T., Yuan, Y., Gong, C., Chen, K., and Wang, Z. Self-PU: Self boosted and calibrated positive-unlabeled training. In *ICML*, 2020b.

Cheng, H. and Tan, P.-N. Semi-supervised learning with data calibration for long-term time series forecasting. In *KDD*, pp. 133–141, 2008.

Christoffel, M., Niu, G., and Sugiyama, M. Class-prior estimation for learning from positive and unlabeled data. In *ACML*, pp. 221–236. PMLR, 2016.

Chung, K. L. *A course in probability theory*. Academic press, 1968.

Cole, R. J., Kripke, D. F., Gruen, W., Mullaney, D. J., and Gillin, J. C. Automatic sleep/wake identification from wrist activity. *Sleep*, 15(5):461–469, 1992.

De Comité, F., Denis, F., Gilleron, R., and Letouzey, F. Positive and unlabeled examples help learning. In *ALT*, pp. 219–230, 1999.

Denis, F. Pac learning from positive statistical queries. In *ALT*, pp. 112–126, 1998.

Dizaji, K. G., Chen, W., and Huang, H. Deep large-scale multitask learning network for gene expression inference. *Journal of Computational Biology*, 28(5):485–500, 2021.

Draper, N. R. and Smith, H. *Applied regression analysis*, volume 326. John Wiley & Sons, 1998.

Du Plessis, M., Niu, G., and Sugiyama, M. Convex formulation for learning from positive and unlabeled data. In *ICML*, pp. 1386–1394, 2015.

Du Plessis, M. C., Niu, G., and Sugiyama, M. Analysis of learning from positive and unlabeled data. In *NIPS*, pp. 703–711, 2014.

Enders, C. K. *Applied missing data analysis*. Guilford press, 2010.

Hammoudeh, Z. and Lowd, D. Learning from positive and unlabeled data with arbitrary positive shift. In *NeurIPS*, 2020.

Hu, W., Le, R., Liu, B., Ji, F., Ma, J., Zhao, D., and Yan, R. Predictive adversarial learning from positive and unlabeled data. In *AAAI*, pp. 7806–7814, 2021.

Huber, P. J. et al. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.

Inan, O., Etemadi, M., Paloma, A., Giovangrandi, L., and Kovacs, G. Non-invasive cardiac output trending during exercise recovery on a bathroom-scale-based ballistocardiograph. *Physiological measurement*, 30(3):261, 2009.

Jain, S., White, M., and Radivojac, P. Estimating the class prior and posterior from noisy positives and unlabeled data. In *NeurIPS*, pp. 2693–2701, 2016.

Jain, S., Delano, J., Sharma, H., and Radivojac, P. Class prior estimation with biased positives and unlabeled examples. In *AAAI*, pp. 4255–4263, 2020.

Jean, N., Xie, S. M., and Ermon, S. Semi-supervised deep kernel learning: Regression with unlabeled data by minimizing predictive variance. In *NeurIPS*, 2018.

Kato, M., Teshima, T., and Honda, J. Learning from positive and unlabeled data with a selection bias. In *ICLR*, 2019.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Lee, W., Yoon, H., Han, C., Joo, K., and Park, K. Physiological signal monitoring bed for infants based on load-cell sensors. *Sensors*, 16(3):409, 2016.

Letouzey, F., Denis, F., and Gilleron, R. Learning from positive and unlabeled examples. In *ALT*, pp. 71–85, 2000.

Li, C., Li, X., Feng, L., and Ouyang, J. Who is your right mixup partner in positive and unlabeled learning. In *ICLR*, 2021.

Li, T., Wang, C.-C., Ma, Y., Ortal, P., Zhao, Q., Stenger, B., and Hirate, Y. Learning classifiers on positive and unlabeled data with policy gradient. In *ICDM*, pp. 399–408, 2019.

Lin, X., Chen, H., Xu, Y., Xu, C., Gui, X., Deng, Y., and Wang, Y. Federated learning with positive and unlabeled data. In *ICML*, pp. 13344–13355, 2022.

Luo, C., Zhao, P., Chen, C., Qiao, B., Du, C., Zhang, H., Wu, W., Cai, S., He, B., Rajmohan, S., et al. Pulns: Positive-unlabeled learning with effective negative sample selector. In *AAAI*, pp. 8784–8792, 2021.

Ma, W. and Chen, G. H. Missing not at random in matrix completion: The effectiveness of estimating missingness probabilities under a low nuclear norm assumption. In *NeurIPS*, 2019.

Mullaney, D., Kripke, D., and Messin, S. Wrist-actigraphic estimation of sleep time. *Sleep*, 3(1):83–92, 1980.

Nair, V. and Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *ICML*, pp. 807–814, 2010.

Narula, S. C. and Wellington, J. F. The minimum sum of absolute errors regression: A state of the art survey. *International Statistical Review/Revue Internationale de Statistique*, pp. 317–326, 1982.

Nukaya, S., Shino, T., Kurihara, Y., Watanabe, K., and Tanaka, H. Noninvasive bed sensing of human biosignals via piezoceramic devices sandwiched between the floor and bed. *IEEE Sensors journal*, 12(3):431–438, 2010.

Pearson, R. K. The problem of disguised missing data. *Acm Sigkdd Explorations Newsletter*, 8(1):83–92, 2006.

Qahtan, A. A., Elmagarmid, A., Castro Fernandez, R., Ouzzani, M., and Tang, N. Fahes: A robust disguised missing values detector. In *KDD*, pp. 2100–2109, 2018.

Ramaswamy, H., Scott, C., and Tewari, A. Mixture proportion estimation via kernel embeddings of distributions. In *ICML*, pp. 2052–2060. PMLR, 2016.

Sakai, T. and Shimizu, N. Covariate shift adaptation on learning from positive and unlabeled data. In *AAAI*, pp. 4838–4845, 2019.

Seal, D. B., Das, V., Goswami, S., and De, R. K. Estimating gene expression from dna methylation and copy number variation: a deep learning regression model for multi-omics integration. *Genomics*, 112(4):2833–2841, 2020.

Sen, S. Kaggle dataset. https://www.kaggle.com/sagarsen/breathing-data-from-a-chest-belt, 2016.

Shi, H., Pan, S., Yang, J., and Gong, C. Positive and unlabeled learning via loss decomposition and centroid estimation. In *IJCAI*, pp. 2689–2695, 2018.

Shi, X., Gao, Z., Lausen, L., Wang, H., Yeung, D.-Y., Wong, W.-k., and Woo, W.-c. Deep learning for precipitation nowcasting: A benchmark and a new model. In *NIPS*, pp. 5617–5627, 2017.

Smieja, M., Struski, Ł., Tabor, J., Zieliński, B., and Spurek, P. Processing of missing data by neural networks. In *NeurIPS*, 2018.

Sonntag, J., Engel, M., and Schmidt-Thieme, L. Predicting parking availability from mobile payment transactions with positive unlabeled learning. In *AAAI*, pp. 15408–15415, 2021.

Sportisse, A., Boyer, C., and Josse, J. Estimation and imputation in probabilistic principal component analysis with missing not at random data. In *NeurIPS*, 2020.

Srinivas, K., Rani, B. K., Rao, M., Patra, R. K., Madhukar, G., and Mahendar, A. Prediction of heart disease using hybrid linear regression. *European Journal of Molecular & Clinical Medicine*, 7(5):1159–1171, 2020.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Tanaka, Y., Iwata, T., Tanaka, T., Kurashima, T., Okawa, M., and Toda, H. Refining coarse-grained spatial data using auxiliary spatial data sets with various granularities. In *AAAI*, volume 33, pp. 5091–5099, 2019.

Tryon, W. W. *Activity measurement in psychology and medicine*. Springer Science & Business Media, 2013.

Vandal, T., Kodra, E., Ganguly, S., Michaelis, A., Nemani, R., and Ganguly, A. R. Deepsd: Generating high resolution climate change projections through single image super-resolution. In *KDD*, pp. 1663–1672, 2017.

Velloso, E. UCI machine learning repository. https://archive.ics.uci.edu/ml/datasets/Weight+Lifting+Exercises+monitored+with+Inertial+Measurement+Units, 2013.

Velloso, E., Bulling, A., Gellersen, H., Ugulino, W., and Fuks, H. Qualitative activity recognition of weight lifting exercises. In *AH*, pp. 116–123, 2013.

Wang, Z., He, Z., and Chen, J. D. Robust time delay estimation of bioelectric signals using least absolute deviation neural network. *IEEE Transactions on biomedical engineering*, 52(3):454–462, 2005.

Webster, J. B., Kripke, D. F., Messin, S., Mullaney, D. J., and Wyborney, G. An activity-based sleep monitor system for ambulatory use. *Sleep*, 5(4):389–399, 1982.

Wilby, R. L., Charles, S., Zorita, E., Timbal, B., Whetton, P., and Mearns, L. Guidelines for use of climate scenarios developed from statistical downscaling methods. *Supporting material of the Intergovernmental Panel on Climate Change, available from the DDC of IPCC TGCIA*, 27, 2004.

Wilcox, R. R. *Introduction to robust estimation and hypothesis testing*. Academic Press, 1997.

Wilton, J., Koay, A., Ko, R., Xu, M., and Ye, N. Positive-unlabeled learning using random forests via recursive greedy risk minimization. In *NeurIPS*, 2022.

Yeung, M. S., Tegnér, J., and Collins, J. J. Reverse engineering gene networks using singular value decomposition and robust regression. *Proceedings of the National Academy of Sciences*, 99(9):6163–6168, 2002.

Zhang, C., Ren, D., Liu, T., Yang, J., and Gong, C. Positive and unlabeled learning with label disambiguation. In *IJCAI*, pp. 1–7, 2019.

Zhang, C., Hou, Y., and Zhang, Y. Learning from positive and unlabeled data without explicit estimation of class prior. In *AAAI*, pp. 6762–6769, 2020.

Zhou, F., Li, T., Zhou, H., Zhu, H., and Ye, J. Graph-based semi-supervised learning with non-ignorable non-response. In *NeurIPS*, 2019.

Zhou, Z.-H. and Li, M. Semi-supervised regression with co-training. In *IJCAI*, pp. 908–913, 2005.

Zhu, X. and Goldberg, A. B. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.

A. Proofs

A.1. Proof of Lemma 2.2

Proof. For the proof of Lemma 2.2, we will derive two important lemmas from Assumption 2.1. Then, we will prove Lemma 2.2 by using them.

We first show $f^*(\boldsymbol{x}) \leq y' \Rightarrow \epsilon_{\mathrm{a}} = 0$. When $f^*(\boldsymbol{x}) \leq y'$, we have the following from Eqs. (1) and (4):

$$\begin{aligned} f^*(\boldsymbol{x}) &\leq f^*(\boldsymbol{x}) + \epsilon_{\mathrm{s}} + \epsilon_{\mathrm{a}} \\ 0 &\leq \epsilon_{\mathrm{s}} + \epsilon_{\mathrm{a}} \\ -\epsilon_{\mathrm{a}} &\leq \epsilon_{\mathrm{s}}. \end{aligned} \tag{12}$$

Since $\epsilon_{\mathrm{a}} \leq 0$ by Assumption 2.1, we have

$$|\epsilon_{\mathrm{a}}| \leq \epsilon_{\mathrm{s}}. \tag{13}$$

If $\epsilon_{\mathrm{a}} < 0$, Assumption 2.1 implies $|\epsilon_{\mathrm{s}}| < |\epsilon_{\mathrm{a}}|$, which contradicts Eq. (13). Hence, we must have

$$\epsilon_{\mathrm{a}} = 0. \tag{14}$$

Since $y = y'$ when $\epsilon_{\mathrm{a}} = 0$, we have

$$\begin{aligned} p(\boldsymbol{x}, y' | f^*(\boldsymbol{x}) \leq y') &= p(\boldsymbol{x}, y' | f^*(\boldsymbol{x}) \leq y', \epsilon_{\mathrm{a}} = 0) \\ &= p(\boldsymbol{x}, y | f^*(\boldsymbol{x}) \leq y, \epsilon_{\mathrm{a}} = 0) \\ &= p(\boldsymbol{x}, y | f^*(\boldsymbol{x}) \leq y), \end{aligned} \tag{15}$$

which establishes

Lemma A.1. *Let $p(\boldsymbol{x}, y, y')$ be the underlying probability distribution for $\boldsymbol{x}$, y, and y'. Then,*

$$p(\boldsymbol{x}, y'|f^*(\boldsymbol{x}) \leq y') = p(\boldsymbol{x}, y|f^*(\boldsymbol{x}) \leq y). \tag{16}$$

Similar to Lemma A.1, we show $f(x) \leq y' \Rightarrow \epsilon_{\mathrm{a}} = 0$. Let $\mathcal{F}' \equiv \{f \in \mathcal{F} : |f(\boldsymbol{x}) - f^*(\boldsymbol{x})| \leq |\epsilon_{\mathrm{s}}| \text{ a.s.}\}$, which represents our natural expectation that the regression function f well approximates f^*. When $f(\boldsymbol{x}) \leq y'$, we have the following from Eqs. (1) and (4) with the condition $f \in \mathcal{F}'$:

$$\begin{aligned} f(\boldsymbol{x}) &\leq f^*(\boldsymbol{x}) + \epsilon_{\mathrm{s}} + \epsilon_{\mathrm{a}} \\ f(\boldsymbol{x}) &\leq f(\boldsymbol{x}) + \epsilon_{\mathrm{s}} + \epsilon_{\mathrm{a}} + |\epsilon_{\mathrm{s}}| \\ 0 &\leq \epsilon_{\mathrm{s}} + \epsilon_{\mathrm{a}} + |\epsilon_{\mathrm{s}}| \\ -\epsilon_{\mathrm{a}} &\leq \epsilon_{\mathrm{s}} + |\epsilon_{\mathrm{s}}|. \end{aligned} \tag{17}$$

Since $\epsilon_{\mathrm{a}} \leq 0$ by Assumption 2.1, we have

$$|\epsilon_{\mathrm{a}}| \leq \epsilon_{\mathrm{s}} + |\epsilon_{\mathrm{s}}|. \tag{18}$$

If $\epsilon_{\mathrm{a}} < 0$, Assumption 2.1 implies $2|\epsilon_{\mathrm{s}}| < |\epsilon_{\mathrm{a}}|$, which contradicts Eq. (18). Hence, we must have

$$\epsilon_{\mathrm{a}} = 0. \tag{19}$$

Since $y = y'$ when $\epsilon_{\mathrm{a}} = 0$, by replacing f^* with f for the argument in the derivation of Lemma A.1 in Eq. (15), we have

Lemma A.2. *Let $\mathcal{F}' \equiv \{f \in \mathcal{F} : |f(\boldsymbol{x}) - f^*(\boldsymbol{x})| \leq |\epsilon_{\mathrm{s}}|\}$. When $f \in \mathcal{F}'$, the following holds:*

$$p(x, y'|f(\boldsymbol{x}) \leq y') = p(x, y|f(\boldsymbol{x}) \leq y). \tag{20}$$

Lemma A.1 immediately implies

$$\mathbb{E}_{p(\boldsymbol{x}, y'|f^*(\boldsymbol{x}) \leq y')}[G(\boldsymbol{x}, y')] = \mathbb{E}_{p(\boldsymbol{x}, y|f^*(\boldsymbol{x}) \leq y)}[G(\boldsymbol{x}, y)] \tag{21}$$

for any function $G : \mathbb{R}^D \times \mathbb{R} \to \mathbb{R}$ as long as the expectations exist. When $f \in \mathcal{F}'$, from Lemma A.2, we then have

$$\mathbb{E}_{p(\boldsymbol{x}, y'|f(\boldsymbol{x}) \leq y')}[G(\boldsymbol{x}, y')] = \mathbb{E}_{p(\boldsymbol{x}, y|f(\boldsymbol{x}) \leq y)}[G(\boldsymbol{x}, y)]. \tag{22}$$

□

A.2. Proof of Proposition 3.2

Proof. From the decomposed gradient $\nabla\mathcal{L}(f_t)$ in Eq. (7), we derive the proposed gradient only with the expectations over $p(\boldsymbol{x}, y')$.

From Condition 3.1 for $L(f(\boldsymbol{x}), y)$, $\nabla L(f(\boldsymbol{x}), y) = \boldsymbol{g}(f(\boldsymbol{x}))$ when $y < f(\boldsymbol{x})$. Thus, Eq. (7) can be rewritten as

$$\begin{aligned} \nabla\mathcal{L}(f_t) =& p(f_t(\boldsymbol{x}) \leq y)\mathbb{E}_{p(\boldsymbol{x}, y|f_t(\boldsymbol{x}) \leq y)}[\nabla L(f_t(\boldsymbol{x}), y)] \\ &+ p(y < f_t(\boldsymbol{x}))\mathbb{E}_{p(\boldsymbol{x}|y < f_t(\boldsymbol{x}))}\Big[\boldsymbol{g}(f_t(\boldsymbol{x}))\Big], \end{aligned} \tag{23}$$

where y is marginalized out in the expectation in the second term since $\boldsymbol{g}(f_t(\boldsymbol{x}))$ does not depend on y.

Here, Eqs. (6) and (7) can be rewritten by replacing $\nabla L(f_t(\boldsymbol{x}), y)$ with $\boldsymbol{g}(f_t(\boldsymbol{x}))$, as

$$\begin{aligned} &\mathbb{E}_{p(\boldsymbol{x}, y)}[\boldsymbol{g}(f_t(\boldsymbol{x}))] \\ &\quad = p(f_t(\boldsymbol{x}) \leq y)\mathbb{E}_{p(\boldsymbol{x}, y|f_t(\boldsymbol{x}) \leq y)}\Big[\boldsymbol{g}(f_t(\boldsymbol{x}))\Big] \\ &\qquad + p(y < f_t(\boldsymbol{x}))\mathbb{E}_{p(\boldsymbol{x}, y|y < f_t(\boldsymbol{x}))}\Big[\boldsymbol{g}(f_t(\boldsymbol{x}))\Big] \end{aligned} \tag{24}$$

$$\begin{aligned} &p(y < f_t(\boldsymbol{x}))\mathbb{E}_{p(\boldsymbol{x}, y|y < f_t(\boldsymbol{x}))}\Big[\boldsymbol{g}(f_t(\boldsymbol{x}))\Big] \\ &\quad = \mathbb{E}_{p(\boldsymbol{x}, y)}\Big[\boldsymbol{g}(f_t(\boldsymbol{x}))\Big] \\ &\qquad - p(f_t(\boldsymbol{x}) \leq y)\mathbb{E}_{p(\boldsymbol{x}, y|f_t(\boldsymbol{x}) \leq y)}\Big[\boldsymbol{g}(f_t(\boldsymbol{x}))\Big]. \end{aligned} \tag{25}$$

Since $\boldsymbol{g}(f_t(\boldsymbol{x}))$ does not depend on y, we can marginalize out y in Eq. (25) as

$$\begin{aligned} &p(y < f_t(\boldsymbol{x}))\mathbb{E}_{p(\boldsymbol{x}|y < f_t(\boldsymbol{x}))}\Big[\boldsymbol{g}(f_t(\boldsymbol{x}))\Big] \\ &\quad = \mathbb{E}_{p(\boldsymbol{x})}\Big[\boldsymbol{g}(f_t(\boldsymbol{x}))\Big] \\ &\qquad - p(f_t(\boldsymbol{x}) \leq y)\mathbb{E}_{p(\boldsymbol{x}|f_t(\boldsymbol{x}) \leq y)}\Big[\boldsymbol{g}(f_t(\boldsymbol{x}))\Big]. \end{aligned} \tag{26}$$

From Eq. (26), we can express Eq. (23) as

$$\begin{aligned} \nabla\mathcal{L}(f_t) =& p(f_t(\boldsymbol{x}) \leq y)\mathbb{E}_{p(\boldsymbol{x}, y|f_t(\boldsymbol{x}) \leq y)}[\nabla L(f_t(\boldsymbol{x}), y)] \\ &+ \mathbb{E}_{p(\boldsymbol{x})}\Big[\boldsymbol{g}(f_t(\boldsymbol{x}))\Big] \\ &- p(f_t(\boldsymbol{x}) \leq y)\mathbb{E}_{p(\boldsymbol{x}|f_t(\boldsymbol{x}) \leq y)}\Big[\boldsymbol{g}(f_t(\boldsymbol{x}))\Big]. \end{aligned} \tag{27}$$

Finally, from Lemma 2.2, we can rewrite Eq. (27) as

$$\begin{aligned} \nabla\mathcal{L}(f_t) =& p(f_t(\boldsymbol{x}) \leq y)\mathbb{E}_{p(\boldsymbol{x}, y'|f_t(\boldsymbol{x}) \leq y')}[\nabla L(f_t(\boldsymbol{x}), y)] \\ &+ \mathbb{E}_{p(\boldsymbol{x})}\Big[\boldsymbol{g}(f_t(\boldsymbol{x}))\Big] \\ &- p(f_t(\boldsymbol{x}) \leq y)\mathbb{E}_{p(\boldsymbol{x}|f_t(\boldsymbol{x}) \leq y')}\Big[\boldsymbol{g}(f_t(\boldsymbol{x}))\Big], \end{aligned} \tag{28}$$

which is identical to Eq. (9). Thus, the gradient in Eq. (9) is unbiased and consistent with the gradient in Eq. (6) a.s. □

A.3. Proof of Lemma 3.4

Proof. The difference between the decomposed gradients $\nabla\check{\mathcal{L}}(f_t)$ and $\nabla\mathcal{L}(f_t)$ at step $t+1$ in the gradient descent is

$$
\begin{aligned}
&|\nabla\check{\mathcal{L}}(f_t) - \nabla\mathcal{L}(f_t)| \qquad (29)\\
&= \Big| p(f_t(\boldsymbol{x}) \le y)\mathbb{E}_{p(\boldsymbol{x},y'|f_t(\boldsymbol{x})\le y')}[\nabla L(f_t(\boldsymbol{x}),y)]\\
&\quad + p(y < f_t(\boldsymbol{x}))\mathbb{E}_{p(\boldsymbol{x},y'|y'<f_t(\boldsymbol{x}))}[\nabla L(f_t(\boldsymbol{x}),y)]\\
&\quad - p(f_t(\boldsymbol{x}) \le y)\mathbb{E}_{p(\boldsymbol{x},y|f_t(\boldsymbol{x})\le y)}[\nabla L(f_t(\boldsymbol{x}),y)]\\
&\quad - p(y < f_t(\boldsymbol{x}))\mathbb{E}_{p(\boldsymbol{x},y|y<f_t(\boldsymbol{x}))}[\nabla L(f_t(\boldsymbol{x}),y)]\Big|.
\end{aligned}
$$

From Lemma 2.2 and Condition 3.1,

$$
\begin{aligned}
&|\nabla\check{\mathcal{L}}(f_t) - \nabla\mathcal{L}(f_t)| \qquad (30)\\
&= \Big| p(y < f_t(\boldsymbol{x}))\mathbb{E}_{p(\boldsymbol{x},y'|y'<f_t(\boldsymbol{x}))}[\nabla L(f_t(\boldsymbol{x}),y)]\\
&\quad - p(y < f_t(\boldsymbol{x}))\mathbb{E}_{p(\boldsymbol{x},y|y<f_t(\boldsymbol{x}))}[\nabla L(f_t(\boldsymbol{x}),y)]\Big|\\
&= \Big| p(y < f_t(\boldsymbol{x}))\mathbb{E}_{p(\boldsymbol{x}|y'<f_t(\boldsymbol{x}))}[\boldsymbol{g}(f_t(\boldsymbol{x}))]\\
&\quad - p(y < f_t(\boldsymbol{x}))\mathbb{E}_{p(\boldsymbol{x}|y<f_t(\boldsymbol{x}))}[\boldsymbol{g}(f_t(\boldsymbol{x}))]\Big|.
\end{aligned}
$$

We decompose $\mathbb{E}_{p(\boldsymbol{x}|y'<f_t(\boldsymbol{x}))}[\boldsymbol{g}(f_t(\boldsymbol{x}))]$ again as

$$
\begin{aligned}
&|\nabla\check{\mathcal{L}}(f_t) - \nabla\mathcal{L}(f_t)| \qquad (31)\\
&= \Big| p(y < f_t(\boldsymbol{x}))\Big(\\
&p(f_t(\boldsymbol{x}) \le y|y' < f_t(\boldsymbol{x}))\mathbb{E}_{p(\boldsymbol{x}|y'<f_t(\boldsymbol{x})\wedge f_t(\boldsymbol{x})\le y)}[\boldsymbol{g}(f_t(\boldsymbol{x}))]\\
&+ p(y < f_t(\boldsymbol{x})|y' < f_t(\boldsymbol{x}))\\
&\qquad \mathbb{E}_{p(\boldsymbol{x}|y'<f_t(\boldsymbol{x})\wedge y<f_t(\boldsymbol{x}))}[\boldsymbol{g}(f_t(\boldsymbol{x}))]\Big)\\
&\quad - p(y < f_t(\boldsymbol{x}))\mathbb{E}_{p(\boldsymbol{x}|y<f_t(\boldsymbol{x}))}[\boldsymbol{g}(f_t(\boldsymbol{x}))]\Big|.
\end{aligned}
$$

The condition $y' < f_t(x) \wedge y < f_t(x)$ is equivalent to the condition $y < f_t(x)$ since $y' \le y$ from Assumption 2.1. Then, we have

$$
\begin{aligned}
&|\nabla\check{\mathcal{L}}(f_t) - \nabla\mathcal{L}(f_t)| \qquad (32)\\
&= \Big| p(y < f_t(\boldsymbol{x}))\Big(\\
&p(f_t(\boldsymbol{x}) \le y|y' < f_t(\boldsymbol{x}))\mathbb{E}_{p(\boldsymbol{x}|y'<f_t(\boldsymbol{x})\wedge f_t(\boldsymbol{x})\le y)}[\boldsymbol{g}(f_t(\boldsymbol{x}))]\\
&+ p(y < f_t(\boldsymbol{x})|y' < f_t(\boldsymbol{x}))\\
&\qquad \mathbb{E}_{p(\boldsymbol{x}|y<f_t(x))}[\boldsymbol{g}(f_t(\boldsymbol{x}))]\Big)\\
&\quad - p(y < f_t(\boldsymbol{x}))\mathbb{E}_{p(\boldsymbol{x}|y<f_t(\boldsymbol{x}))}[\boldsymbol{g}(f_t(\boldsymbol{x}))]\Big|.
\end{aligned}
$$

Additionally, since $p(y < f_t(\boldsymbol{x})|y' < f_t(\boldsymbol{x})) = 1 - p(f_t(\boldsymbol{x}) \le y|y' < f_t(\boldsymbol{x}))$,

$$
\begin{aligned}
&|\nabla\check{\mathcal{L}}(f_t) - \nabla\mathcal{L}(f_t)| \qquad (33)\\
&= \Big| p(y < f_t(\boldsymbol{x}))p(f_t(\boldsymbol{x}) \le y|y' < f_t(\boldsymbol{x}))\Big(\\
&\qquad \mathbb{E}_{p(\boldsymbol{x}|y'<f_t(\boldsymbol{x})\wedge f_t(\boldsymbol{x})\le y)}[\boldsymbol{g}(f_t(\boldsymbol{x}))]\\
&\qquad - \mathbb{E}_{p(\boldsymbol{x}|y<f_t(\boldsymbol{x}))}[\boldsymbol{g}(f_t(\boldsymbol{x}))]\Big)\Big|.
\end{aligned}
$$

This equation shows that the bias is represented by the difference between the expectation of $\boldsymbol{g}(f_t(\boldsymbol{x}))$ with the lower-side data and that with the original upper-side data mixed into the lower side due to the asymmetric noise and the corresponding proportions.

From Assumption 3.3, since $\epsilon_{\mathrm{a}} \perp\!\!\!\perp \boldsymbol{x}$,

$$
\begin{aligned}
&|\nabla\check{\mathcal{L}}(f_t) - \nabla\mathcal{L}(f_t)| \qquad (34)\\
&= \Big| p(y < f_t(\boldsymbol{x}))p(f_t(\boldsymbol{x}) \le y|y' < f_t(\boldsymbol{x}))\Big(\\
&\mathbb{E}_{p(\boldsymbol{x}|f_t(\boldsymbol{x})\le y)}[\boldsymbol{g}(f_t(\boldsymbol{x}))] - \mathbb{E}_{p(\boldsymbol{x}|y<f_t(\boldsymbol{x}))}[\boldsymbol{g}(f_t(\boldsymbol{x}))]\Big)\Big|.
\end{aligned}
$$

Since $|f - f^*| \le |\epsilon_{\mathrm{s}}|$ a.s., $p(f_t(\boldsymbol{x}) \le y) = \eta$ and $p(y < f_t(\boldsymbol{x})) = 1 - \eta$ from their definition,

$$
\begin{aligned}
&p(f_t(\boldsymbol{x}) \le y|y' < f_t(\boldsymbol{x})) =\\
&\qquad \frac{p(f_t(\boldsymbol{x}) \le y)p(\epsilon_{\mathrm{a}} < 0)}{p(y < f_t(\boldsymbol{x})) + p(f_t(\boldsymbol{x}) \le y)p(\epsilon_{\mathrm{a}} < 0)}\\
&\qquad = \frac{\eta(1-\xi)}{(1-\eta)+\eta(1-\xi)}. \qquad (35)
\end{aligned}
$$

Therefore, from the definition of δ,

$$
\begin{aligned}
|\nabla\check{\mathcal{L}}(f_t) - \nabla\mathcal{L}(f_t)| &\ge \frac{\eta(1-\eta)(1-\xi)}{(1-\eta)+\eta(1-\xi)}\delta\\
&= \frac{\eta(1-\eta)(1-\xi)}{1-\eta\xi}\delta. \qquad (36)
\end{aligned}
$$

□

B. Implementation of Learning Algorithm Based on Stochastic Optimization

We scale up our U2 regression algorithm by stochastic approximation with M mini-batches and add a regularization term, $R(f)$:

$$
\begin{aligned}
\nabla\hat{\mathcal{L}}^{\{m\}}(f_t) = &\sum_{(\boldsymbol{x},y)\in\left\{\boldsymbol{X}_{\mathrm{up}}^{\{m\}},\boldsymbol{y}_{\mathrm{up}}^{\{m\}}\right\}} \nabla L(f_t(\boldsymbol{x}),y) \qquad (37)\\
&+ \rho\left[\sum_{\boldsymbol{x}\in\boldsymbol{X}_{\mathrm{un}}^{\{m\}}} \boldsymbol{g}(f_t(\boldsymbol{x}))\right] - \sum_{\boldsymbol{x}\in\boldsymbol{X}_{\mathrm{up}}^{\{m\}}} \boldsymbol{g}(f_t(\boldsymbol{x})) + \lambda\frac{\partial R(f_t)}{\partial\boldsymbol{\theta}},
\end{aligned}
$$

where $\nabla\hat{\mathcal{L}}^{\{m\}}(f_t)$ is the gradient for the m-th mini-batch, $\{\boldsymbol{X}_{\mathrm{up}}^{\{m\}}, \boldsymbol{y}_{\mathrm{up}}^{\{m\}}\}$ and $\boldsymbol{X}_{\mathrm{un}}^{\{m\}}$ are the upper-side and unlabeled sets in the m-th mini-batch based on the current f_t, respectively, λ is a regularization parameter, and the regularization term $R(f)$ is, for example, the L1 or L2 norm of the parameter vector $\boldsymbol{\theta}$ of f. We also convert $n_{\mathrm{up}}/(\pi_{\mathrm{up}}N)$ to a hyperparameter ρ, ignoring constant coefficients instead of directly handling π_{up}. The hyperparameters ρ and λ are optimized in training based on the grid-search with the validation set.

The U2 regression algorithm based on stochastic optimization is described in Algorithm 1. We learn the regression function with the gradient in Eq. (37) by utilizing any stochastic gradient method. By using the learned f, we can estimate $\hat{y} = f(\boldsymbol{x})$ for new data $\boldsymbol{x}$.

C. Algorithm for LU Regression

We show the algorithm for the *lower and unlabeled* regression (LU regression), where labels for some observations may become inconsistently *higher* than those for typical observations. Let $L_{\mathrm{LU}}(f(\boldsymbol{x}), y)$ be a loss function for LU regression and $\boldsymbol{g}_{\mathrm{LU}}(f(\boldsymbol{x}))$ be a gradient of $\nabla L_{\mathrm{LU}}(f(\boldsymbol{x}), y)$ when $f(\boldsymbol{x}) \leq y$. Similar to Condition 3.1 for U2 regression, we assume that the class of $L_{\mathrm{LU}}(f(\boldsymbol{x}), y)$ satisfies the condition that $\boldsymbol{g}_{\mathrm{LU}}(f(\boldsymbol{x}))$ is a gradient function depending only on $f(\boldsymbol{x})$ and not on the value of y. Then, LU regression is derived as Algorithm 1, with the following gradient, $\nabla\hat{\mathcal{L}}_{\mathrm{LU}}^{\{m\}}(f_t)$, instead of $\nabla\hat{\mathcal{L}}^{\{m\}}(f_t)$ in Eq. (37), as

$$\nabla\hat{\mathcal{L}}_{\mathrm{LU}}^{\{m\}}(f_t) = \sum_{\{\boldsymbol{x},y\}\in\left\{\boldsymbol{X}_{\mathrm{lo}}^{\{m\}}, \boldsymbol{y}_{\mathrm{lo}}^{\{m\}}\right\}} \nabla L_{\mathrm{LU}}(f_t(\boldsymbol{x}), y) + \rho\left[\sum_{\boldsymbol{x}\in\boldsymbol{X}_{\mathrm{un}}^{\{m\}}} \boldsymbol{g}_{\mathrm{LU}}(f_t(\boldsymbol{x}))\right] - \sum_{\boldsymbol{x}\in\boldsymbol{X}_{\mathrm{lo}}^{\{m\}}} \boldsymbol{g}_{\mathrm{LU}}(f_t(\boldsymbol{x})) + \lambda\frac{\partial R(f_t)}{\partial\boldsymbol{\theta}}, \quad (38)$$

where $\{\boldsymbol{X}_{\mathrm{lo}}^{\{m\}}, \boldsymbol{y}_{\mathrm{lo}}^{\{m\}}\}$ and $\boldsymbol{X}_{\mathrm{un}}^{\{m\}}$ are the lower-side and unlabeled sets in the m-th mini-batch based on the current f_t, respectively.

D. Computing Infrastructure

All of the experiments were carried out with a Python and TensorFlow implementation on workstations having 80 GB of memory, a 4.0 GHz CPU, and an Nvidia Titan X GPU. In this environment, the computational time to produce the results was a few hours.

Algorithm 1 U2 regression based on stochastic gradient method.

input: Training data $\mathcal{D}' = \{\boldsymbol{x}_n, y'_n\}_{n=1}^N$; hyperparameters $\rho, \lambda \geq 0$; an external stochastic gradient method $\mathcal{A}$
output: Model parameters $\boldsymbol{\theta}$ for f

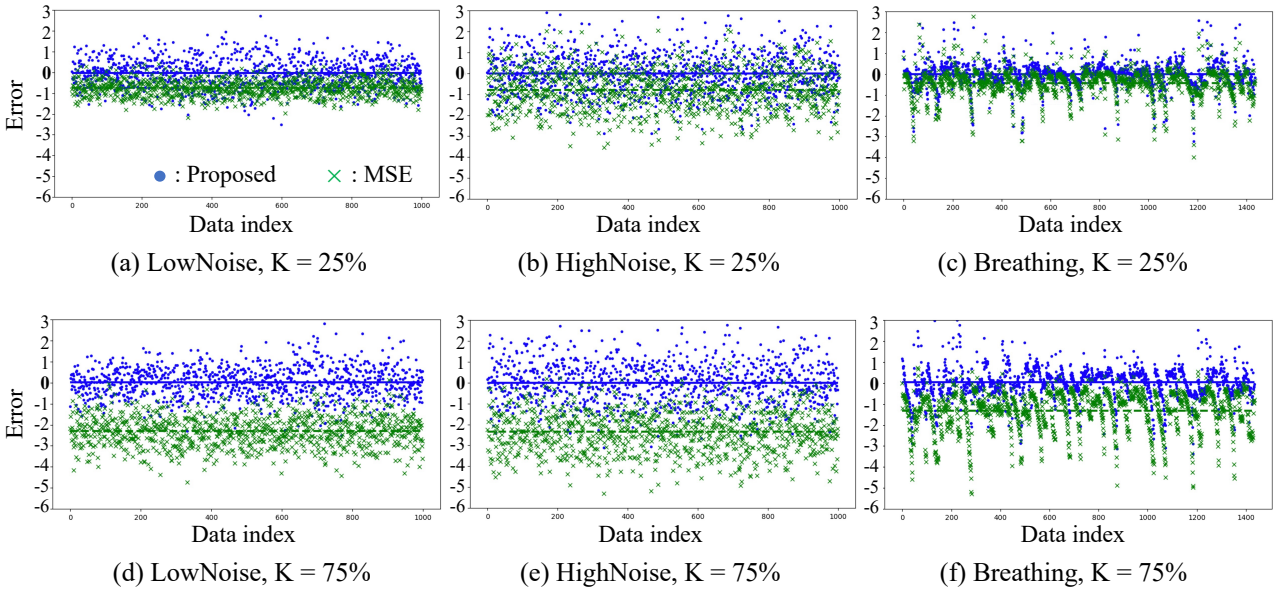
```
while No stopping criterion has been met do
    Shuffle D' into M mini-batches: {X^{m}, y^{m}}_{m=1}^{M}
    for m = 1 to M do
        Compute the gradient ∇L̂^{m}(f_t) in Eq. (37) with {X^{m}, y^{m}}
        Update θ by A with ∇L̂^{m}(f_t)
    end for
end while
```

E. Details of Experiments in Section 4.2.1

E.1. Details of LowNoise and HighNoise tasks

We conducted the experiments on synthetic data to evaluate the feasibility of our method for obtaining unbiased learning results from asymmetrically corrupted data containing different proportions of incomplete observations. We generated synthetic data on the basis of Assumption 2.1 and Eq. (4). We randomly generated $N = 1,000$ training samples, $\boldsymbol{X} = \{\boldsymbol{x}_n\}_{n=1}^N$, from the standard Gaussian distribution $\mathcal{N}(\boldsymbol{x}_n; 0, \boldsymbol{I})$, where the number of features in $\boldsymbol{x}$ was $D = 10$, and $\boldsymbol{I}$ is the identity matrix. Then, using $\boldsymbol{X}$, we generated the corresponding N sets of true labels $\boldsymbol{y} = \{y_n\}_{n=1}^N$ from the distribution $\mathcal{N}(y_n; \boldsymbol{w}^\top\boldsymbol{x}_n, \beta)$, where $\boldsymbol{w}$ are coefficients that were also randomly generated from the standard Gaussian distribution $\mathcal{N}(\boldsymbol{w}; 0, \boldsymbol{I})$, β is the noise precision, and $\top$ denotes the transpose. For simulating the situation in which a label has incomplete observations, we created corrupted labels $\boldsymbol{y}' = \{y'_n\}_{n=1}^N$ by randomly selecting the K percent of data in $\boldsymbol{y}$ and subtracting the absolute value of white Gaussian noise with the standard deviation having twice the value of that for generating $\boldsymbol{y}$ from their values. We repeatedly evaluated the proposed method for each of the following settings. The noise precision was $\beta = \{10^0, 10^{-1}\}$, which corresponds to a low-noise setting task (**LowNoise**) and a high-noise setting task (**HighNoise**), and the proportion of incomplete training samples was $K = \{25, 50, 75\}\%$. In the case of $K = 75\%$, only 25 percent of the samples correctly corresponded to labels, and all of the other samples were attached with labels that were lower than the corresponding true values. It is quite difficult to learn regression functions using such data.

Implementation. In these tasks, we used a linear model, $\boldsymbol{\theta}^\top\boldsymbol{x}$, for $f(\boldsymbol{x})$ and an implementation for Eq. (37) with the absolute loss, which satisfies Condition 3.1, for the loss function L and L1-regularization for the regularization term. We set the candidates of the hyperparameters, ρ and



(a) LowNoise, K = 25% (b) HighNoise, K = 25% (c) Breathing, K = 25%

(d) LowNoise, K = 75% (e) HighNoise, K = 75% (f) Breathing, K = 75%

Figure 6. Errors in prediction (predicted value minus true value) by proposed method (blue) and by MSE (green) for tasks (a) **LowNoise**, $K = 25\%$, (b) **HighNoise**, $K = 25\%$, (c) **Breathing**, $K = 75\%$, (d) **LowNoise**, $K = 75\%$, (e) **HighNoise**, $K = 75\%$, and (f) **Breathing**, $K = 75\%$. Error of each data point is shown by a dot (for proposed method) or a cross mark (for MSE), and average error is shown by a solid line (for proposed method) or dashed line (for MSE).

λ, to $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$. We standardized the data by subtracting their mean and dividing by their standard deviation in the training split. We used Adam with the hyperparameters recommended in (Kingma & Ba, 2015), and the number of samples in the mini-batches was set to 32.

E.2. Details of Breathing Task

We also used a real-world sensor dataset collected from the Kaggle dataset (Sen, 2016) that contains breathing signals (**Breathing**). The dataset consisted of $N = 1,432$ samples. We utilized signals from a chest belt as $\boldsymbol{X} = \{\boldsymbol{x}_n\}_{n=1}^N$, and $\boldsymbol{x}$ in each sample had $D = 2$ number of features, i.e., the period and height of the expansion/contraction of the chest. We utilized signals obtained by the Douglas bag (DB) method, which is the gold standard for measuring ventilation, as true labels $\boldsymbol{y} = \{y_n\}_{n=1}^N$. For our problem setting, we created corrupted labels $\boldsymbol{y}' = \{y'_n\}_{n=1}^N$ through the same procedure for synthetic corruption as that for LowNoise and HighNoise with $K = \{25, 50, 75\}\%$.

Implementation. In the experiment on Breathing, for its non-linearity, we used $\boldsymbol{\theta}^\top \boldsymbol{\phi}(\boldsymbol{x}, \sigma)$ for $f(\boldsymbol{x})$, where $\boldsymbol{\phi}$ is a radial basis function with the training set as its bases, and σ is a hyperparameter representing the kernel width that is also optimized by using the validation set. We set the candidates of the hyperparameter σ to $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$. The other implementation details were the same as those for **LowNoise** and **HighNoise**.

E.3. Detailed Results

Figure 6 shows the error between the estimation results of the proposed method and their true values and those of MSE for **LowNoise**, **HighNoise**, and **Breathing** with 25 and 75 percent of incomplete training samples. Table 3 shows the performance on **LowNoise**, **HighNoise**, and **Breathing** for the proposed method and MSE. As shown in Fig. 6, the proposed method obtained unbiased learning results in all cases, while MSE produced biased results. From Table 3, we can see that the proposed method outperformed MSE overall with $0 < K$. We found that the performance of our method was not significantly affected by the increase in the proportion of incomplete training samples K even for $K = 75\%$, unlike that of MSE. We also show results with $K = 0\%$, which means there is no corruption. Except for the **LowNoise** task, the proposed method worked comparably to MSE even when $K = 0\%$, which shows that there is almost no drawback if we use the proposed method even for the uncorrupted data without incomplete observations. Since **LowNoise** with $K = 0\%$ is very clean and simple data with a linear relationship between $\boldsymbol{x}$ and y and little additive white Gaussian noise, MSE can perform much better than robust methods, including the proposed method. We again note that this paper addresses cases where there is the asymmetric label corruption. If there is no corruption, we can use simple loss functions, such as squared loss.

Table 3. Comparison of proposed method and MSE in terms of MAE (smaller is better). Best methods are in bold. Confidence intervals are standard errors.

(a) LowNoise

	$K = 0\%$	$K = 25\%$	$K = 50\%$	$K = 75\%$
MSE	$\mathbf{0.24 \pm 0.01}$	0.77 ± 0.01	1.53 ± 0.02	2.30 ± 0.02
Proposed	0.57 ± 0.01	$\mathbf{0.55 \pm 0.01}$	$\mathbf{0.54 \pm 0.01}$	$\mathbf{0.58 \pm 0.01}$

(b) HighNoise

	$K = 0\%$	$K = 25\%$	$K = 50\%$	$K = 75\%$
MSE	0.77 ± 0.02	1.03 ± 0.02	1.62 ± 0.03	2.36 ± 0.03
Proposed	0.79 ± 0.02	$\mathbf{0.79 \pm 0.02}$	$\mathbf{0.80 \pm 0.02}$	$\mathbf{0.80 \pm 0.02}$

(c) Breathing

	$K = 0\%$	$K = 25\%$	$K = 50\%$	$K = 75\%$
MSE	0.41 ± 0.02	0.55 ± 0.02	0.91 ± 0.02	1.32 ± 0.02
Proposed	0.45 ± 0.01	$\mathbf{0.43 \pm 0.01}$	$\mathbf{0.46 \pm 0.01}$	$\mathbf{0.59 \pm 0.01}$

F. Details of Experiments in Section 4.2.2

F.1. Detailed Results

In Fig. 7, we show charts similar to Fig. 3 (the error in prediction for **LowNoise**, **HighNoise**, and **Breathing** tasks with $K = 50\%$) when we used 1% of the training set as the validation set. We can see that even in this case, the proposed method achieved unbiased learning (the average error shown by the blue solid line is approximately zero).

G. Details of Experiments in Section 4.2.3

G.1. Details of Task

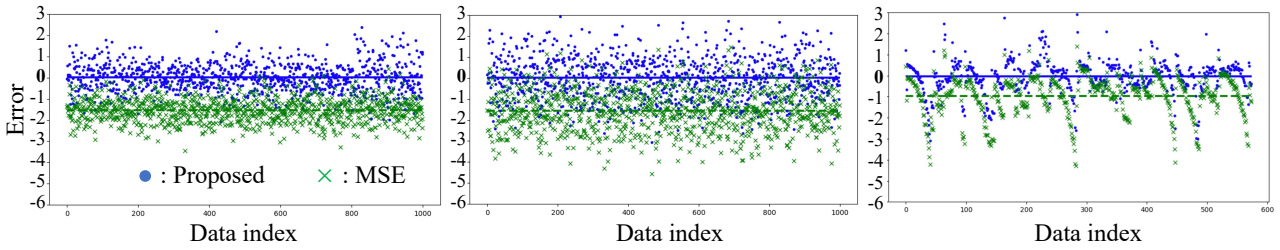
We applied the algorithm to five different real-world healthcare tasks recorded in the datasets from the UCI Machine Learning Repository (Velloso, 2013; Velloso et al., 2013), which contains sensor outputs from wearable devices attached to the arm while subjects exercised. From the non-intrusive sensors attached to gym equipment, we estimated the motion intensity of a subject that was measured accurately with an intrusive sensor wrapped around the arm. If we can mimic outputs from the arm sensor with outputs from the equipment sensor, it could contribute to the subjects' comfort, as they would not need to wear sensors to measure their motion intensity. We utilized all of the features from the equipment sensor that took "None" values less than ten times as $\boldsymbol{X} = \{\boldsymbol{x}_n\}_{n=1}^N$, where each sample had $D = 13$ number of features. The corrupted labels $\boldsymbol{y}' = \{y'_n\}_{n=1}^N$ were the magnitude of acceleration from the arm sensor, which can accurately sense motion intensity on the arm, but it had insufficient data coverage and incomplete observations for the movements of other body parts. For performance evaluation, we used the magnitude of acceleration for the entire body as true labels $\boldsymbol{y} = \{y_n\}_{n=1}^N$. The number of samples was $N = 11,159$, $N = 7,593$, $N = 6,844$, $N = 6,432$, and $N = 7,214$ for the **Specification**, **Throwing A**, **Lifting**, **Lowering**, and **Throwing B** tasks, respectively.

Implementation. For the complex nature of the tasks, we used a 6-layer multilayer perceptron with ReLU (Nair & Hinton, 2010) (more specifically, D-100-100-100-100-1) as $f(\boldsymbol{x})$, which demonstrates the usefulness of the proposed method for training deep neural networks. We also used a dropout (Srivastava et al., 2014) with a rate of 50% after each fully connected layer. We used two implementations of $L(f(\boldsymbol{x}), y)$ for $f(\boldsymbol{x}) \leq y'$ in Eq. (10): the absolute loss (`Proposed-1`) and the squared loss (`Proposed-2`). In both implementations, we use the absolute loss, which satisfies Condition 3.1, for $L(f(\boldsymbol{x}), y)$ when $y' < f(\boldsymbol{x})$. We used L1-regularization for the regularization term. The other implementation details were the same as those for **LowNoise**, **HighNoise**, and **Breathing**.

H. Details of Experiments in Section 4.2.4

H.1. Details of Task

We demonstrate the practicality of our approach in a real use case in healthcare. From non-intrusive bed sensors installed under each of the four legs of a bed, we estimated the motion intensity of a subject that was measured accurately with ActiGraph, a gold standard intrusive sensor wrapped around the wrist (Tryon, 2013; Mullaney et al., 1980; Webster et al., 1982; Cole et al., 1992). The sensing results of ActiGraph



(a) LowNoise, K = 50% Prop of val set = 1%

(b) HighNoise, K = 50% Prop of val set = 1%

(c) Breathing, K = 50% Prop of val set = 1%

Figure 7. Errors in prediction when we choose hyperparameters with 1% of the training set as a validation set. Other configurations were the same as those in Fig. 3.

are used for tasks such as discriminating whether a subject is asleep or awake (Cole et al., 1992). While ActiGraph can accurately sense motion on the forearm, it has insufficient data coverage in other areas and often causes observations of movements on other body parts to be missing. The bed sensors have a broader data coverage since they can sense global motion on all body parts; however, the sensing accuracy is limited due to their non-intrusiveness. If we can mimic the outputs from ActiGraph with outputs from the bed sensors, we can expect to achieve sufficient accuracy and coverage while also easing the burden on the subject. The dataset we used included three pieces of data, Data (i), (ii), and (iii), which were respectively recorded over 20, 18, and 18.5 minutes. Each piece of data consisted of pairs of bed-sensor-data sequences and the corresponding motion intensity sequence obtained by ActiGraph. We used the "magnitude" attribute of ActiGraph as corrupted labels $\boldsymbol{y}'$ for the motion intensity, whose sampling rate was about one sample per second. For true labels $\boldsymbol{y}$, we manually measured the motion intensity every minute under the management of a domain expert. For $\boldsymbol{X}$, we first computed the gravity center of the four sensor outputs that were obtained from the bed sensors under the four legs of a bed. Then, we computed the time derivatives and cross terms of the raw sensor outputs and the gravity center. The sampling rate of the bed sensors was different from that of ActiGraph (about one sample per five milliseconds). Thus, $\boldsymbol{X}$ was finally generated as a sliding window of statistics in $1,000$-millisecond (1-second) subsequences of the time series of the above computed variables, where 1 second was the same as the sampling interval of ActiGraph. The statistics were means, standard deviations, and $\{0.05, 0.25, 0.5, 0.75, 0.95\}$ quantiles. In the end, the numbers of samples and features were $N = 3,390$ and $D = 84$.

Implementation. In this task, we used the linear model $\boldsymbol{\theta}^\top \boldsymbol{x}$ for $f(\boldsymbol{x})$ due to its interpretability, which is inevitable in real-world healthcare and medical applications. The other implementation details were the same as those for **LowNoise** and **HighNoise**.

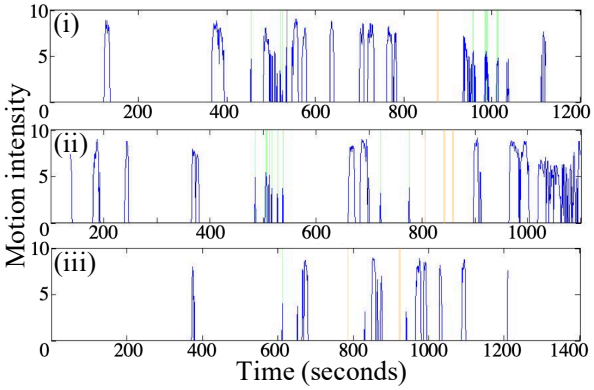


Figure 8. Experiment on real use case for healthcare. Blue line represents our estimation results for motion intensity. White (non-colored) area shows that both proposed method and ActiGraph correctly estimated motion intensity of the subject at this duration. Green area shows that our method could capture motion at this duration while ActiGraph could not. Orange area shows that our method could not capture motion at this duration but ActiGraph could. Gray area shows that our method mistakenly captured noise as subject's motion.

H.2. Estimation Results for Motion Intensity

Figure 8 compares our estimation results for motion intensity with the output of ActiGraph and true labels.

H.3. Important Features for Estimating Motion Intensity

The important features selected by L1 regularization were the statistics of the gravity center and the cross terms and time derivatives of the raw sensor outputs. The largest weight was assigned to the standard deviation of the gravity center, which represents the amplitude of the gravity center, so it is directly related to the motion of subjects.