Supplementary Material
Extended version of Online and Stochastic Learning with a Human Cognitive Bias

Hidekazu Oiwa
The University of Tokyo and JSPS Research Fellow
7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan
hidekazu.oiwa@gmail.com

Hiroshi Nakagawa
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan
nakagawa@dl.itc.u-tokyo.ac.jp

Abstract

Sequential learning for classification tasks is an effective tool in the machine learning community. In sequential learning settings, algorithms sometimes make incorrect predictions on data that were correctly classified in the past. This paper explicitly deals with such inconsistent prediction behavior. Our main contributions are 1) to experimentally show its effect for user utilities as a human cognitive bias, 2) to formalize a new framework by internalizing this bias into the optimization problem, 3) to develop new algorithms without memorization of the past prediction history, and 4) to show some theoretical guarantees of our derived algorithm for both online and stochastic learning settings. Our experimental results show the superiority of the derived algorithm for problems involving human cognition.

1 Introduction

Online learning and stochastic learning are advantageous for large-scale learning. Sequential processing of data is the key of these methods. For classification tasks, these learning algorithms process a bunch of data one by one and change its classification rule at every round. We call these methods sequential learning in this paper.

Sequential learning algorithms sometimes make wrong predictions on data that were correctly classified in the past. While classical performance evaluation measures for sequential learning, such as the expected loss, do not reflect the history of the past prediction results, previous algorithms have not considered this inconsistent behavior as a crucial factor. The key statement in this paper is that this phenomenon has a crucial impact on the evaluation of algorithms on the condition that humans are evaluators. Humans have a cognitive bias that they attach a higher value to the data that were correctly classified in the past than the other data. This effect originates from the endowment effect that had been widely analyzed in the field of behavior economics. There are motivating examples in which this cognitive bias has important roles:

- **User utility maximization**: Sequential learning has been used in many services such as image object recognition and email filtering (Aberdeen, Pacovsky, and Slater 2010). Many users continuously utilize services whose prediction rules have been changed over time. Furthermore, some users check prediction results of previously seen data. Negative flips may drastically decrease the utilities of these users.
- **Interactive annotation**: There are many human-computer interaction systems based on sequential learning such as active learning-based annotations (Settles 2011). Encouraging people to make annotations is crucial for more data generation and better performance. Some annotators may feel frustrated annotating the data correctly classified in the past as wrong ones.

To maximize the availability of machine learning, algorithms which interact with humans need to adjust update rules to heal the bias derived from the past prediction history. We explicitly deal with this cost as the divestiture loss. We first conducted an experiment to verify whether the endowment effect negatively affects human's evaluations. Next, we set new evaluation measures for sequential learning by incorporating the endowment loss. This measure imposes an additional objective on sequential learning, minimizing the divestiture loss. We note that this new problem setting can be easily dealt with if algorithms could store all previous examples and its prediction results in the memory; however, this memorization is unpractical for large-scale learning setting due to the memory constraint. To solve this problem, we derived new variants of Online Gradient Descent (OGD). Our derived algorithms enable to retain reasonable convergence guarantees for both online learning and stochastic learning settings without data memorization. We lastly conducted experiments and the results showed advantages of our algorithm compared with the conventional ones in the sequential learning framework with a human cognitive bias.

1.1 Notations

Scalars are denoted by lower-case x and vectors are denoted by bold lower-case $\mathbf{x}$. t-th training input vectors and labels are denoted $\mathbf{x}_t$ and y_t. Input vectors are n-dimensional and taken from the input space $\mathcal{X} \subset \mathbb{R}^n$. Output labels are taken from the output space $\mathcal{Y}$. For simplicity, we define $z_t = (\mathbf{x}_t, y_t)$ to describe t-th datum. $\mathbf{x}_{s:t}$ describes a sequence of vectors from s-th to t-th and $\mathbf{x}_{1:0}$ is a empty set. $1_{a=b}$ is a boolean function which becomes 1 only if $a = b$.

Algorithm 1 Sequential learning framework

Initialize $\mathbf{w}_1 = \mathbf{0} \in \mathbb{R}^n$, the size of data : T
for $t = 1, \ldots, T$ **do**
  Receive $\mathbf{x}_t \in \mathcal{X}$
  Predict corresponding output $\hat{y}_t = \mathrm{sgn}(\langle \mathbf{w}_t, \mathbf{x}_t \rangle)$
  Unveil true output $y_t \in \{-1, 1\}$
  Incur loss $\ell(\mathbf{w}_t; z_t)$
  Update weight vector and obtain $\mathbf{w}_{t+1} \in \mathcal{W}$
end for

2 Sequential Learning

Let us begin by outlining a general sequential learning setting for binary classification tasks, that is, $\mathcal{Y} = \{-1, 1\}$. Furthermore, we focus on linear prediction models in this paper. In this setting, the prediction is performed through the sign of the inner product of $\mathbf{w}$ and $\mathbf{x}$, that is, $\hat{y} = \mathrm{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle)$. The basic iterative procedure is as follows:

1. At round t, receive an input vector $\mathbf{x}_t$.
2. Predict the corresponding output $\hat{y}_t \in \{-1, 1\}$ through the current weight vector $\mathbf{w}_t$.
3. The true label, $y_t \in \{-1, 1\}$, is revealed and incur a cost through the loss function $\ell(\mathbf{w}_t; z_t)$. Loss functions measure the predictability of the weight vector for a specific datum.
4. Update the weight vector to $\mathbf{w}_{t+1}$ in the convex set $\mathcal{W} \subset \mathbb{R}^n$ according to the prediction result.
5. Increment the round number t. The used datum cannot be accessed in the following procedure. Repeat this process until no labeled data remains.

Algorithm 1 summarizes this general framework.

As famous examples of the sequential learning framework, online learning and stochastic learning have recently gained attentions due to its memory efficiency, easiness to re-learning, and adaptation to streaming data.

2.1 Online Learning

Online learning has a great advantage for large-scale data processing. Although the data loading time becomes the dominant factor in the batch learning framework on a large-scale data due to memory constraints (Yu et al. 2012), online learning algorithms can run with a limited memory space. Standard online learning algorithms do not assume any distribution on the data. This framework can be applied under not only an i.i.d. assumption but also an adversarial one wherein an adversary assign a label after algorithms estimate it. As a novel performance measure, the regret is well used. For any $\mathbf{u} \in \mathcal{W}$ and any sequence $z_{1:T}$, regret is defined as:

$$\mathrm{Regret}(T) = \sum_{t=1}^{T} \ell(\mathbf{w}_t; z_t) - \sum_{t=1}^{T} \ell(\mathbf{u}; z_t) \ . \tag{1}$$

The regret is formalized as the difference between two terms; 1) the cumulative loss incurred by the algorithm and 2) the one produced by the fixed optimal weight vector. While no assumption is put on the sequence, it can be measured even in an adversarial setting. If the upper bound of regret is sublinear ($o(T)$), the loss per datum becomes the same as the one of the best fixed strategy.

2.2 Stochastic Learning

In the standard stochastic learning setting, the final goal is the minimization of the expected loss. Let us assume that a certain data distribution $\mathcal{D}$ exists and a sequence of data $z_{1:T}$ is i.i.d. sampled from this distribution. The objective function is the difference between the expected loss evaluated at the final output of the algorithm and the optimal one. For any $\mathbf{u} \in \mathcal{W}$,

$$E_{z\sim\mathcal{D}}\left[\ell(\mathbf{w}; z)\right] - E_{z\sim\mathcal{D}}\left[\ell(\mathbf{u}; z)\right] \ . \tag{2}$$

If the value of this function converges to 0, the algorithm will minimize the expected loss as the best fixed strategy do.

2.3 Online (Stochastic) Gradient Descent

Online gradient descent (OGD)[1] is a simple algorithm for sequential learning. OGD is an iterative algorithm and uses only one datum in each round. OGD updates the weight vector for the reverse direction of the gradient. Therefore, OGD works with any differentiable loss function. The update formula is

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}\left(\mathbf{w}_t - \eta_t \nabla \ell(\mathbf{w}_t; z_t)\right) \ . \tag{3}$$

$\nabla \ell(\mathbf{w}_t; z_t)$ means the gradient of the loss function with respect to a weight vector $\mathbf{w}_t$ evaluated at z_t. $\Pi_{\mathcal{W}}(\cdot)$ is a projection function onto a convex set $\mathcal{W}$ such that $\Pi_{\mathcal{W}}(\mathbf{w}) = \arg\min_{\mathbf{w}' \in \mathcal{W}} \|\mathbf{w} - \mathbf{w}'\|_2$. We can see from this update formula that the weight vector is projected onto $\mathcal{W}$ if it moves to the outside of $\mathcal{W}$. $\eta_{1:T}$ is a sequence of positive learning rates. The weight vector is continuously updated according to formula (3) whenever OGD receives new one datum.

OGD uses a first-order approximation of loss functions to update the weight vector for the sake of faster calculation. Therefore, OGD is well used when computational constraints are crucial concerns. OGD has been experimentally shown to have good performances even if its theoretical properties are worse than other algorithms (Bottou and Bousquet 2011). OGD has been the topic of extensive theoretical analysis. OGD obtains a sublinear regret upper bound under practical constraints.

Theorem 1. *(Zinkevich 2003) Let $\mathbf{w}_{1:T+1}$ be derived according to OGD's update formula (3). Assume that for all $\mathbf{w} \in \mathcal{W}$, $\|\mathbf{w}\|_2 \leq R$ and for all t, $\|\nabla \ell(\mathbf{w}_t; z_t)\|_2 \leq G$. When $\eta_t = \sqrt{2}R/G\sqrt{t}$, the upper regret bound is*

$$\mathrm{Regret}(T) \leq 2\sqrt{2}RG\sqrt{T} = O(\sqrt{T}) \ . \tag{4}$$

From this result, we see that OGD is guaranteed to converge to obtain the optimal average loss. If T is known in advance, OGD can achieve a tighter bound by setting an appropriate fixed learning rate. When the loss function is

[1] In a stochastic learning setting, this algorithm is called stochastic gradient descent (SGD). Though these two algorithms have distinct objectives, the skeleton of their update procedures is almost the same. We use the term OGD for describing both types of algorithms if there is no explicit statement.

strongly convex, OGD converges to the optimal solution in $O(\log T)$ (Hazan, Agarwal, and Kale 2007; Shalev-Shwartz and Kakade 2008).

When OGD is used in a stochastic optimization setting, the average weight vector is guaranteed to converge to the optimal weight vector. We define $\mathcal{D}^t$ as a sequence of labeled data $z_{1:t}$ i.i.d. sampled from a distribution $\mathcal{D}$ and define an average weight vector as $\bar{\mathbf{w}} = \sum_{t=1}^{T} \mathbf{w}_t / T$.

Theorem 2. *(Cesa-Bianchi, Conconi, and Gentile 2004) Assume that the conditions in Theorem 1 are satisfied. If loss functions are convex, for any* $\mathbf{u} \in \mathcal{W}$*,*

$$E_{\mathcal{D}^T}\left[E_{z\sim\mathcal{D}}\left[\ell(\bar{\mathbf{w}}; z)\right]\right] - E_{z\sim\mathcal{D}}\left[\ell(\mathbf{u}; z)\right] \leq \frac{2\sqrt{2}RG}{\sqrt{T}} . \quad (5)$$

The convergence rate is $O(1/\sqrt{T})$ and OGD is guaranteed to converge to the optimal weight vector.

3 Sequential learning with a cognitive bias

From the nature of sequential update, algorithms sometimes make mistakes on the data that were correctly classified in the past. We first show that this event largely affects user utilities neglected in the context of the standard sequential learning setting. Next, we propose a new objective taking in this human cognitive bias. The endowment effect is a key component to analyze this bias.

3.1 Endowment Effect

The endowment effect (Thaler 1980) induces in humans a cognitive bias to prevent rational decision-making. The endowment effect states that people tend to put a higher value on preventing the loss of an item they already possess than on buying the same item they does not possess. This human psychological bias has an important role for utility maximization and human engagements. There are many work on theoretical explanations and experimental tests of the endowment effect (Kahneman, Knetsch, and Thaler 1990).

The endowment effect suggests that the cost of compensation is larger than the cost of paying. Here, the notion of the endowment effect is that people would pay more in order to sustain the correct prediction result for past data than to pay for a correct prediction on new data. As a result, the sequential learner must take the data received in the past into consideration when updating the model. This notion should be incorporated into the objective as an additional cost.

3.2 Experiment on the endowment effect

Here, we describe an experiment and results demonstrating that the endowment effect is prominent in the user utilities. We conducted a subjective experiment using a crowdsourcing market place to assign tasks to humans. We set up a synthetic scene recognition task as a binary classification problem using indoor recognition datasets[2]. We used pictures of bookstores and restaurants from this large dataset.

[2] http://web.mit.edu/torralba/www/indoor.html

Table 1: Experimental result on the endowment effect. The table shows the number of people who evaluate each session's predictability.

	1	2	3	4	5	Average
type-I	2	3	15	73	7	3.80
type-II (duplicate)	2	11	26	54	7	3.53

We assigned a certain amount of tasks to each worker. Each session consisted of two phases, training phase and evaluation phase. In the training phase, workers received eight pairs of a picture and its predicted label. After seeing the given pairs, workers checked whether each label was correct and then sent their answers to the system as user feedback. In the evaluation phase, the system showed eight different pairs of a picture and its prediction result to workers. Workers were told that the previous user feedback was used to classify samples in the evaluation phase. Workers evaluated the learnability of this system on a five-star scale.

Each worker dealt with two types of sessions. In the type-I, the same picture did not appear in both the training and evaluation phases. Therefore, the endowment effect was not activated in this session. In the type-II, two pictures were redisplayed in the evaluation phase. These pictures were correctly classified in the training phase but misclassified in the evaluation phase. The number of correctly classified pictures in both phases was fixed; there were four correctly classified pictures in both phases. Therefore, if worker evaluations were largely different between two sessions, we can see that the endowment effect influenced workers' evaluations. In each session, 100 workers evaluated its learnability and verified whether there is any difference of workers' cognition between these two types or not.

Table 1 shows an experimental result. The result in the table indicates that the type-II sessions have a lower evaluation in comparison with the type-I. The p-value calculated by Mann-Whitney test is less than a 1% level of significance ($p = 0.0093$). This result shows that the endowment effect largely affects workers' evaluations.

3.3 Sequential learning and the endowment effect

We define this negative side-effect as a divestiture loss. The divestiture loss is actualized when the classifier makes an inaccurate prediction but it correctly classifies the same data in the past. To internalize this loss explicitly, we integrate this loss into the optimization problem. When an algorithm already processed S data ($z_{1:S}$) and predicted labels for them ($\hat{y}_{1:S}$), the divestiture loss is defines as:

$$C(\mathbf{w}; z, \hat{y}_{1:S}, z_{1:S}) = 1_{\text{prev}} \gamma \ell(\mathbf{w}; z)$$
$$\text{where } \; 1_{\text{prev}} = \min\left(1, \sum_{s=1}^{S} 1_{z=z_s} 1_{y_s=\hat{y}_s}\right) . \quad (6)$$

γ is a non-negative trade-off parameter between the original objective and the divestiture loss. γ is chosen according to the stakeholder's preference. If $\gamma = 0$, the divestiture loss disappears and the objective function becomes the conventional one. 1_{prev} indicates whether the algorithm correctly classified z in the past. When it correctly classified, 1_{prev} becomes 1 and the algorithm incur an additional loss $\gamma\ell(\mathbf{w}; z)$

from this function. Otherwise, 1_{prev} becomes 0 and this loss will not be activated. We assume that if we correctly classify the same datum more than once, the divestiture loss does not change. New objective functions consist of the sum of the original losses and the divestiture loss. A new regret is defined as follows; For any $\mathbf{u} \in \mathcal{W}$,

$$\text{Regret}(T) = \sum_{t=1}^{T} F_t(\mathbf{w}_t) - \sum_{t=1}^{T} F_t(\mathbf{u})$$
$$\text{where } F_t(\mathbf{w}) = \ell(\mathbf{w}; z_t) + C(\mathbf{w}; z_t, \hat{y}_{1:t-1}, z_{1:t-1}) \ , \quad (7)$$

and for any $\mathbf{u} \in \mathcal{W}$, a new expected loss is

$$E_{\mathcal{D}^T}\left[E_{z\sim\mathcal{D}}\left[G(\mathbf{w}; z)\right]\right] - E_{\mathcal{D}^T}\left[E_{z\sim\mathcal{D}}\left[G(\mathbf{u}; z)\right]\right]$$
$$\text{s.t. } G(\mathbf{w}; z) = \ell(\mathbf{w}; z) + \frac{1}{T}\sum_{t=1}^{T} C(\mathbf{w}; z, \hat{y}_{1:t-1}, z_{1:t-1}) \ . \quad (8)$$

4 Endowment-induced OGD

In the sequential learning setting with a human cognitive bias, the original OGD does not achieve a good experimental result because of the existence of divestiture loss. Although the divestiture loss appears only when the corresponding examples were correctly classified, the original OGD treats all examples the same without referring to on-the-fly prediction results. The original OGD cannot capture this skewness.

We devised the Endowment-induced Online Gradient Descent (E-OGD) to incorporate the notion of the endowment effect into the original OGD. The key idea is to heavily weight correct examples in order to absorb the skewness.

E-OGD divides all examples into two categories:

1. correctly classified examples, i.e., $\hat{y}_t = y_t$
2. wrongly classified examples, i.e., $\hat{y}_t \neq y_t$

We see that the loss corresponding to the former examples is bigger than that of the latter examples due to the divestiture loss. Therefore, correct examples should be treated as being more important than wrong ones. E-OGD first classifies each example into one of two types it should belong to. After the type identification, E-OGD updates parameters heavily with the trade-off parameter γ with respect to correctly classified examples. In summary, the weight vector is updated as follows:

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}\left(\mathbf{w}_t - \eta_t \nabla\ell(\mathbf{w}_t; z_t)\right)$$
$$\text{where } \eta_t = \begin{cases} c(1+\gamma)/\sqrt{t} & \text{if } \hat{y}_t = y_t \\ c/\sqrt{t} & \text{if } \hat{y}_t \neq y_t \end{cases} \ . \quad (9)$$

Algorithm 2 is the pseudo-code of E-OGD. We note that this E-OGD can update parameter by using only the currently received datum. We show that an appropriate step width setting makes the algorithm adaptive to the endowment effect in the following theoretical analysis.

Algorithm 2 Endowment-induced Online Gradient Descent

Require: scaling constant c, trade-off parameter γ
  Initialize $\mathbf{w}_1 = \mathbf{0}$
  for $t = 1, \ldots, T$ **do**
    Receive $\mathbf{x}_t$
    Predict corresponding output $\hat{y} = \text{sgn}\left(\langle \mathbf{w}_t, \mathbf{x}_t \rangle\right)$
    Unveil true output y_t
    if $y_t = \hat{y}_t$ **then**
      $\mathbf{v}_{t+1} = \mathbf{w}_t - c(1+\gamma)\nabla\ell(\mathbf{w}_t; z_t)/\sqrt{t}$
    else
      $\mathbf{v}_{t+1} = \mathbf{w}_t - c\nabla\ell(\mathbf{w}_t; z_t)/\sqrt{t}$
    end if
    $\mathbf{w}_{t+1} = \operatorname{argmin}_{\mathbf{w}\in\mathcal{W}} \|\mathbf{w} - \mathbf{v}_{t+1}\|_2$
  end for

4.1 Theoretical Analysis of E-OGD

Let us analyze the theoretical aspects of E-OGD. For simplifying the following discussions, we introduce a new term:

$$r_t(z) = 1 + \gamma \min\left(1, \sum_{s=1}^{t-1} 1_{z=z_s} 1_{y_s=\hat{y}_s}\right) \ , \quad (10)$$

and denote $r_t(z_t)$ as r_t and $\ell(\cdot; z_t)$ as $\ell_t(\cdot)$. We analyze the upper regret bound and the upper bound of the expected loss of E-OGD in this section. All proofs of theorems and lemmas are written in the Appendix. Furthermore, we indicate another option of step widths and its theoretical analysis in the Appendix.

First, we show relationship between a sequence of step widths in E-OGD and the endowment effect. We set a sequence of step widths as:

$$\eta_t = \begin{cases} c(1+\gamma)/\sqrt{t} & \text{if } \hat{y}_t = y_t \\ c/\sqrt{t} & \text{if } \hat{y}_t \neq y_t \end{cases} \ , \quad (11)$$

where c is some positive constant. Regret (7) is rewritten as:

$$\text{Regret}(T) = \sum_{t=1}^{T} r_t \ell_t(\mathbf{w}_t) - \min_{\mathbf{u}\in\mathcal{W}} \sum_{t=1}^{T} r_t \ell_t(\mathbf{u}) \ . \quad (12)$$

The next theorem gives the regret upper bound of E-OGD.

Theorem 3. *Let $\mathbf{w}_1, \ldots, \mathbf{w}_{T+1}$ be derived according to E-OGD's update rule. Assume that for all $\mathbf{w}_t$, $\|\mathbf{w}_t\|_2 \leq R$, $\|\nabla\ell_t(\mathbf{w}_t)\|_2 \leq G$. When we set a sequence of step widths $\eta_{1:T}$ and assume the condition as in Lemma 3, the upper bound of regret is obtained by setting $c = \sqrt{2}R/G(1+\gamma)$ as follows:*

$$\text{Regret}(T) \leq 2\sqrt{2}RG(1+\gamma)\sqrt{T} \ . \quad (13)$$

From this theorem, E-OGD is guaranteed to converge to obtain the optimal average loss with respect to the online learning setting with a human cognitive bias.

For stochastic learning setting, we assume that the data is i.i.d. sampled from a distribution $\mathcal{D}$. The final goal is to minimize the sum of the expected loss and divestiture loss, as described by formula (8). Lemma 1 reformulates the optimization problem into an easily analyzable form.

Lemma 1. *The optimization problem in the stochastic learning setting can be reformulated through $r_t(z)$.*

$$E_{\mathcal{D}^T}\left[E_{z\sim\mathcal{D}}\left[\frac{1}{T}\sum_{t=1}^{T} r_t(z)\ell(\mathbf{w}; z)\right]\right] \ . \quad (14)$$

Furthermore, it can be reformulated through a new distribution $\mathcal{D}_P$ and an appropriate constant value $H_{\mathcal{D}^T}$ conditioned on $z_{1:T}$ as

$$E_{\mathcal{D}^T}\left[H_{\mathcal{D}^T} E_{z\sim\mathcal{D}_P}\left[\ell(\mathbf{w};z)\right]\right] \ . \tag{15}$$

The following theorem is derived from Theorem 3 and Lemma 1 in order to upper bound the expected loss with a human cognitive bias (8).

Theorem 4. *Assume that the conditions in Theorem 3 are satisfied and there is an integer t_p such that $r_t(z) = r_{t_p}(z)$ for any $t \geq t_p$. In this setting, the following formula is satisfied for any $\mathbf{u} \in \mathcal{W}$.*

$$E_{\mathcal{D}^T}\left[E_{z\sim\mathcal{D}_P}\left[\ell(\bar{\mathbf{w}};z)\right]\right] - E_{\mathcal{D}^T}\left[E_{z\sim\mathcal{D}_P}\left[\ell(\mathbf{u};z)\right]\right] \leq \frac{\sqrt{2}RG(1+\gamma)}{(\sqrt{T}-(t_p+1)/\sqrt{T})/(2-\sqrt{t_p-1}/\sqrt{T})} \ , \tag{16}$$

where $\bar{\mathbf{w}} = \sum_{t=t_p}^{T} \mathbf{w}_t/(T - t_p + 1)$.

Lemma 1 derives that the left-hand side of the formula (16) equals the original objective function (8). From this theoretical result, the average weight vector converges to the optimal one that minimizes the sum of the expected loss and the divestiture loss. If $t_p \ll T$, the convergence speed is $O(1/\sqrt{T})$. And, when the number of data is finite, there is some constant t_p such that $r_t(z) = r_{t_p}(z)$ for any $t \geq t_p$.

4.2 Importance-aware Update

When E-OGD receives a correctly classified example, the weight vector is updated by $1+\gamma$ scaling. This update can be viewed as an approximate update of the original OGD at $1+\gamma$ times. An importance-aware update can be established in order to make an exact $1+\gamma$ times update through an one-time update (Karampatziakis and Langford 2011).

The original OGD and E-OGD do not hold some important properties such as invariance and safety. The invariance property guarantees that the parameter updates per example with an importance weight h should be the same as regular updates that appear h times in a row. The safety property guarantees that the magnitude relationship between $\hat{y}$ and y does not change by the update using the received datum. When the endowment effect is strong (r is large), the plain E-OGD might overshoot when the prediction is correct because the step width becomes large. The safety property guarantees to prevent this type of overshooting. The importance-aware update framework provides a closed-form update formula for calculating the weight vector by solving an ordinary differential equation. The weight vector can be updated with one closed-form formula for many major convex loss functions such as the hinge-loss and logistic loss. Besides the importance-aware update framework, there are other methods for obtaining invariance, such as rejection sampling (Zadrozny, Langford, and Abe 2003).

5 Experiments

We conducted experiments to test the performance of the conventional OGD and E-OGD in the online learning framework with a human cognitive bias. We used five large-scale data sets from the LIBSVM binary data collections[3]. The specifications of these dataset are listed in Table 2. news20 and rcv1 are news category classification tasks. algebra and BtA (Bridge to Algebra) are KDD Cup 2010 datasets to predict whether students correctly answer algebra problems. webspam-t is a tri-gram webspam classification dataset used in the Pascal Large Scale Learning Challenge. The original webspam-t dataset is not splited to two sets, therefore, we randomly sample 90% data from the dataset and used them as a training set and remaining data as a test set.

Table 2: Dataset Specifications. T is the number of training data. S is test data size. N is the number of features.

	T	S	N
news20	15,000	4,996	1,335,191
rcv1	20,242	677,399	47,236
algebra	8,407,752	510,302	20,216,830
BtA	19,264,097	748,401	29,890,095
webspam-t	315,000	35,000	16,609,143

We used these datasets to compare the performances of OGD and E-OGD in a new stochastic learning setting. We incur both expected loss and divestiture loss. To evaluate the divestiture loss, we replaced some examples in the test data with some training examples at a specific rate. The training examples are randomly extracted from the training set. If the algorithm correctly classified in the training phase but it misclassified the same example in the test phase, they incur a divestiture loss. We conducted experiments by setting the replacement rate of the test examples by training examples as 5, 10, and 30%. We quantified the performance as

$$\frac{1}{S}\sum_{s=1}^{S}\ell(\mathbf{w};z_s) + \frac{\gamma}{S}\sum_{z_p\in P}\ell(\mathbf{w};z_p) \ . \tag{17}$$

The first term corresponds to the expected loss, and each datum z_s corresponds to one datum in the test set or a replaced training example. S is the number of test data. The second term corresponds to the divestiture loss, and each datum z_p corresponds to the example regarding the divestiture loss. P is an example set that satisfies two conditions: (1) the example was extracted from the training dataset in exchange for test examples; (2) the example was correctly classified when the example appeared in the training phase. The cumulative loss is defined as the sum of these two losses.

Let the weight vector spaces $\mathcal{W}$ be a N-dimensional Euclidean space where N is the number of features. We used logistic loss as a loss function. Each algorithm learned the weight vector from training set through 1 iteration. Learning rates are $\eta_t = \eta/\sqrt{t}$. We varied η from 10^3 to 1.91×10^{-3} with common ratio $1/2$ to minimize cumulative loss.

In addition to the normal setting, we performed several experiments. We show a brief result here. First, we verified that E-OGD outperformed OGD in most datasets when we set the hinge-loss as a loss function. Next, we made the value of γ bigger and verified that E-OGD has maintained an advantage over OGD. These results indicate that the advantage

[3] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html

Table 3: Experimental results compared to the conventional OGD: the expected loss, divestiture loss, and cumulative loss (Iteration: 1). The lowest values in each replace rate r, loss type, and dataset are written in **bold**.

		$r = 0.05$		$r = 0.1$		$r = 0.3$	
	Loss Type	E-OGD	OGD	E-OGD	OGD	E-OGD	OGD
news20	Expected	$\mathbf{5.10 \times 10^{-2}}$	5.31×10^{-2}	$\mathbf{5.67 \times 10^{-2}}$	5.84×10^{-2}	$\mathbf{9.18 \times 10^{-2}}$	9.32×10^{-2}
	Divestiture	$\mathbf{8.71 \times 10^{-3}}$	1.39×10^{-2}	$\mathbf{7.73 \times 10^{-3}}$	1.27×10^{-2}	$\mathbf{6.71 \times 10^{-3}}$	1.07×10^{-2}
	Cumulative	$\mathbf{5.97 \times 10^{-2}}$	6.70×10^{-2}	$\mathbf{6.44 \times 10^{-2}}$	7.11×10^{-2}	$\mathbf{9.85 \times 10^{-2}}$	1.04×10^{-1}
rcv1	Expected	7.39×10^{-2}	$\mathbf{7.37 \times 10^{-2}}$	7.83×10^{-2}	$\mathbf{7.80 \times 10^{-2}}$	9.71×10^{-2}	$\mathbf{9.65 \times 10^{-2}}$
	Divestiture	$\mathbf{1.10 \times 10^{-2}}$	1.84×10^{-2}	$\mathbf{1.04 \times 10^{-2}}$	1.74×10^{-2}	$\mathbf{8.04 \times 10^{-3}}$	1.35×10^{-2}
	Cumulative	$\mathbf{8.49 \times 10^{-2}}$	9.21×10^{-2}	$\mathbf{8.87 \times 10^{-2}}$	9.54×10^{-2}	$\mathbf{1.05 \times 10^{-1}}$	1.10×10^{-1}
algebra	Expected	3.31×10^{-1}	$\mathbf{3.06 \times 10^{-1}}$	3.30×10^{-1}	$\mathbf{3.06 \times 10^{-1}}$	3.26×10^{-1}	$\mathbf{3.03 \times 10^{-1}}$
	Divestiture	$\mathbf{6.00 \times 10^{-2}}$	1.01×10^{-1}	$\mathbf{5.68 \times 10^{-2}}$	9.60×10^{-2}	$\mathbf{4.41 \times 10^{-2}}$	7.46×10^{-2}
	Cumulative	$\mathbf{3.91 \times 10^{-1}}$	4.08×10^{-1}	$\mathbf{3.87 \times 10^{-1}}$	4.02×10^{-1}	$\mathbf{3.70 \times 10^{-1}}$	3.78×10^{-1}
BtA	Expected	3.29×10^{-1}	$\mathbf{3.11 \times 10^{-1}}$	3.27×10^{-1}	$\mathbf{3.10 \times 10^{-1}}$	3.18×10^{-1}	$\mathbf{3.03 \times 10^{-1}}$
	Divestiture	$\mathbf{7.64 \times 10^{-2}}$	1.17×10^{-1}	$\mathbf{7.24 \times 10^{-2}}$	1.11×10^{-1}	$\mathbf{5.63 \times 10^{-2}}$	8.62×10^{-2}
	Cumulative	$\mathbf{4.05 \times 10^{-1}}$	4.28×10^{-1}	$\mathbf{3.99 \times 10^{-1}}$	4.21×10^{-1}	$\mathbf{3.74 \times 10^{-1}}$	3.90×10^{-1}
webspam-t	Expected	$\mathbf{3.45 \times 10^{-2}}$	3.51×10^{-2}	$\mathbf{3.49 \times 10^{-2}}$	3.53×10^{-2}	$\mathbf{3.75 \times 10^{-2}}$	3.76×10^{-2}
	Divestiture	$\mathbf{8.11 \times 10^{-3}}$	1.09×10^{-2}	$\mathbf{7.72 \times 10^{-3}}$	1.04×10^{-2}	$\mathbf{5.63 \times 10^{-3}}$	7.73×10^{-3}
	Cumulative	$\mathbf{4.29 \times 10^{-2}}$	4.60×10^{-2}	$\mathbf{4.26 \times 10^{-2}}$	4.57×10^{-2}	$\mathbf{4.32 \times 10^{-2}}$	4.54×10^{-2}

of E-OGD becomes more crucial as the importance of divestiture loss becomes larger.

5.1 Experimental Results

Table 3 shows the experimental results when we apply OGD and E-OGD to five datasets. These results indicate E-OGD has a crucial advantage to make divestiture losses lower in all settings, and this effect contributes to low cumulative losses. As a result, E-OGD outperforms OGD on all datasets. Figure 1 plots loss values in each 10,000 rounds when we used BtA dataset to evaluate the performance. These results denote that E-OGD has obtained significantly lower divestiture losses than OGD during most rounds. Low divestiture loss leads to low cumulative loss, and E-OGD has constantly outperformed OGD with respect to cumulative loss. The difference of expected losses between two algorithms becomes smaller while the number of received data increases. On the other hand, the difference of divestiture losses between two algorithms becomes bigger. This result means that E-OGD becomes superior to the normal OGD with respect to the cumulative loss while the data increases.

Table 4 shows the results of importance-aware update versions. These results indicate that the importance-aware update improves the performance of E-OGD in most experimental settings. Moreover, E-OGD largely outperforms OGD in terms of cumulative losses.

6 Related Work

Researchers have developed many online and stochastic learning algorithms as a natural response to the desires of large-scale learning systems (Shalev-Shwartz 2012). Many algorithms pursue to minimize the regret upper bound or the expected loss by using convex (surrogate) loss functions as a major objective. Follow-The-Regularized Leader (FTRL) (Shalev-Shwartz and Singer 2007) is a fundamental template for online convex optimization. Theoretically speaking, FTRL has desirable properties, including a tighter regret bound. A number of cutting-edge algorithms have been derived from FTRL; OGD is one of famous examples. FTRL has been extended to enable it to deal with other problem structures besides online and stochastic learning frameworks (Duchi et al. 2010; Xiao 2010; McMahan 2011). These frameworks enable sparsity-inducing regularization to be integrated into FTRL while preserving the advantages of sequential learning. They derived the sublinear regret upper bound and the convergence property to the optimal point in the stochastic learning setting. The extension to regularized objectives is one of our future research directions.

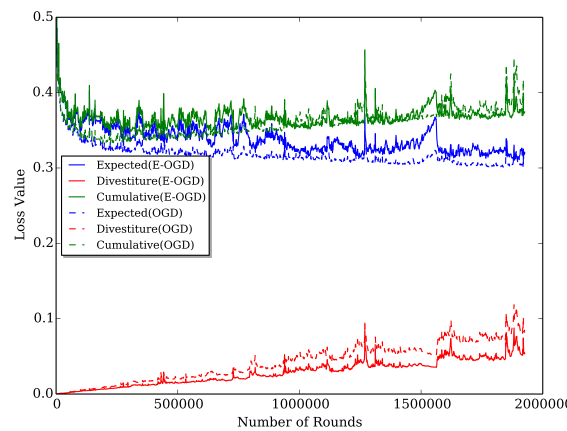

Figure 1: Experimental results on BtA dataset in each 1,000 rounds: the expected loss, divestiture loss, and cumulative loss. The x-axis is the number of rounds. The y-axis denotes the value of each loss. The solid curves are the results obtained by E-OGD. The dotted curves are the results by OGD.

Our framework is similar to the cost-sensitive learning framework wherein the loss of false positives is different from the loss of false negatives. Langford and Beygelzimer (2005) provides a reduction technique that works for clas-

Table 4: Experimental results among importance-aware update family: the expected loss, divestiture loss, and cumulative loss (Iteration: 1). The lowest value in each replace rate r, loss type, and dataset are written in **bold**.

		$r = 0.05$		$r = 0.1$		$r = 0.3$	
	Loss Type	E-OGD	OGD	E-OGD	OGD	E-OGD	OGD
news20	Expected	$\mathbf{3.11 \times 10^{-2}}$	3.85×10^{-2}	$\mathbf{3.40 \times 10^{-2}}$	4.14×10^{-2}	$\mathbf{5.20 \times 10^{-2}}$	5.86×10^{-2}
	Divestiture	$\mathbf{1.37 \times 10^{-2}}$	2.20×10^{-2}	$\mathbf{1.31 \times 10^{-2}}$	2.08×10^{-2}	$\mathbf{9.84 \times 10^{-3}}$	1.59×10^{-2}
	Cumulative	$\mathbf{4.48 \times 10^{-2}}$	6.05×10^{-2}	$\mathbf{4.70 \times 10^{-2}}$	6.22×10^{-2}	$\mathbf{6.18 \times 10^{-2}}$	7.46×10^{-2}
rcv1	Expected	$\mathbf{3.53 \times 10^{-2}}$	3.80×10^{-2}	$\mathbf{3.95 \times 10^{-2}}$	4.18×10^{-2}	$\mathbf{5.69 \times 10^{-2}}$	5.79×10^{-2}
	Divestiture	$\mathbf{1.25 \times 10^{-2}}$	1.78×10^{-2}	$\mathbf{1.18 \times 10^{-2}}$	1.68×10^{-2}	$\mathbf{9.10 \times 10^{-3}}$	1.30×10^{-2}
	Cumulative	$\mathbf{4.79 \times 10^{-2}}$	5.58×10^{-2}	$\mathbf{5.13 \times 10^{-2}}$	5.87×10^{-2}	$\mathbf{6.60 \times 10^{-2}}$	7.09×10^{-2}
algebra	Expected	3.35×10^{-1}	$\mathbf{3.13 \times 10^{-1}}$	3.34×10^{-1}	$\mathbf{3.13 \times 10^{-1}}$	3.30×10^{-1}	$\mathbf{3.09 \times 10^{-1}}$
	Divestiture	$\mathbf{5.89 \times 10^{-2}}$	1.02×10^{-1}	$\mathbf{5.58 \times 10^{-2}}$	9.68×10^{-2}	$\mathbf{4.33 \times 10^{-2}}$	7.52×10^{-2}
	Cumulative	$\mathbf{3.94 \times 10^{-1}}$	4.16×10^{-1}	$\mathbf{3.90 \times 10^{-1}}$	4.09×10^{-1}	$\mathbf{3.73 \times 10^{-1}}$	3.85×10^{-1}
BtA	Expected	3.29×10^{-1}	$\mathbf{3.11 \times 10^{-1}}$	3.27×10^{-1}	$\mathbf{3.10 \times 10^{-1}}$	3.18×10^{-1}	$\mathbf{3.03 \times 10^{-1}}$
	Divestiture	$\mathbf{7.64 \times 10^{-2}}$	1.17×10^{-1}	$\mathbf{7.24 \times 10^{-2}}$	1.11×10^{-1}	$\mathbf{5.62 \times 10^{-2}}$	8.61×10^{-2}
	Cumulative	$\mathbf{4.05 \times 10^{-1}}$	4.28×10^{-1}	$\mathbf{3.99 \times 10^{-1}}$	4.21×10^{-1}	$\mathbf{3.74 \times 10^{-1}}$	3.89×10^{-1}
webspam-t	Expected	$\mathbf{2.62 \times 10^{-2}}$	2.75×10^{-2}	$\mathbf{2.61 \times 10^{-2}}$	2.74×10^{-2}	$\mathbf{2.79 \times 10^{-2}}$	2.90×10^{-2}
	Divestiture	$\mathbf{8.29 \times 10^{-3}}$	1.16×10^{-2}	$\mathbf{7.94 \times 10^{-3}}$	1.11×10^{-2}	$\mathbf{5.87 \times 10^{-3}}$	8.30×10^{-3}
	Cumulative	$\mathbf{3.45 \times 10^{-2}}$	3.91×10^{-2}	$\mathbf{3.40 \times 10^{-2}}$	3.85×10^{-2}	$\mathbf{3.38 \times 10^{-2}}$	3.73×10^{-2}

sification ranging from cost-sensitive to simple binary and Wang, Zhao, and Hoi (2012) proposes a cost-sensitive online classification framework. In our framework, the cost of each example dynamically changes depending on the history of prediction results. Therefore, the problem becomes more complicated than these cost-sensitive frameworks.

7 Conclusion and Open Problems

We established an online and stochastic learning framework with a human cognitive bias by incorporating the notion of the endowment effect. We established an online and stochastic learning framework with a human cognitive bias by incorporating the notion of the endowment effect. In this framework, algorithms need to focus on minimizing not only the original loss but also the divestiture loss. We developed new algorithms applicable to this framework; Endowment-induced Online Gradient Descent (E-OGD). We theoretically showed that E-OGD is guaranteed to have some desirable properties for both online and stochastic learning frameworks with a human cognitive bias. Finally, we experimentally showed that our derived algorithms are effective at a large number of tasks involving human engagements in this framework.

The sequential learning framework with a human cognitive bias has several open issues. The first challenge is a more sophisticated choice of η_t. To obtain the lowest convergence rate, it is the best to set η_t as proportional to $r_t(z_t)$. However, exact matching is almost impossible because the parameter $r_t(z_t)$ is conditioned on a sequence of previous predictions. In the standard sequential learning setting, algorithms cannot preserve the history of observations and its prediction results.

The second issue is the similarity of data. When we use the machine learning algorithm, sometimes we encounter situations in which several data are similar to but not exactly the same as previously seen data. We will verify whether the endowment effect is activated even when data are quite similar to each other. Furthermore, we will incorporate this effect into the optimization problem as a modified version of this framework.

A Proofs of Theorems and Lemmas

We show some proofs of Lemmas and Theorems.

First, we prove Theorem 3. To prove this, we first introduce two lemmas and these proofs.

Lemma 2. *When we set $\eta_{1:T}$ as following the rule of E-OGD and the condition is satisfied, values of r_t and η_t become one of the following two types: (1) $r_t = 1, \eta_t = c/\sqrt{t}$, (2) $r_t = 1 + \gamma, \eta_t = c(1 + \gamma)/\sqrt{t}$.*

Proof. This Lemma is proved directly from the definition of r_t and η_t. When $y_t = \hat{y}_t$ is satisfied, $r_t = 1 + \gamma$ and $\eta_t = c(1 + \gamma)/\sqrt{t}$. When $y_t \neq \hat{y}_t$ and it has never been correctly classified in the past, $r_t = 1$ and $\eta_t = c/\sqrt{t}$. In summary, r_t and η_t becomes one of two pairs of values. □

Lemma 3. $(r_t/\eta_t) - (r_{t-1}/\eta_{t-1}) \geq 0$ *for all $t \geq 2$ when the algorithm has not misclassified the data that were correctly classified in the past.*

Proof. This result can be obtain directly from the results of Lemma 2. We see that r_t/η_t becomes $\sqrt{t}/c$ in both cases in Lemma 2. Therefore, $(r_t/\eta_t) - (r_{t-1}/\eta_{t-1}) = \sqrt{t}/c - \sqrt{t-1}/c > 0$. □

From these lemmas, we can analyze the upper bound of regret of E-OGD in the online learning setting with a human cognitive bias. Below is the proof of Theorem 3.

Proof. For simplicity, let us denote $\nabla \ell(\mathbf{w}_t; z_t)$ as $\mathbf{g}_t$. The convexity of loss functions guarantees that the first-order approximation inequality is satisfied for all $\mathbf{u}$, i.e.,

$$\ell_t(\mathbf{u}) \geq \langle \mathbf{g}_t, \mathbf{u} - \mathbf{w}_t \rangle + \ell_t(\mathbf{w}_t) \ . \qquad (18)$$

This convexity property reformulates the regret bound.

$$\begin{aligned}\text{Regret}(T) &= \sum_{t=1}^{T} r_t(\ell_t(\mathbf{w}_t) - \min_{\mathbf{u}} \ell_t(\mathbf{u})) \\ &\leq \max_{\mathbf{u}} \sum_{t=1}^{T} r_t(\ell_t(\mathbf{w}_t) + \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u}\rangle - \ell_t(\mathbf{w}_t)) \\ &= \max_{\mathbf{u}} \sum_{t=1}^{T} r_t \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u}\rangle \ . \end{aligned} \tag{19}$$

We define for all t, $\mathbf{v}_{t+1} = \mathbf{w_t} - \eta_t \nabla \ell_t(\mathbf{w}_t)$. From this definition, we obtain the following equations.

$$\begin{aligned} \mathbf{v}_{t+1} - \mathbf{u} &= (\mathbf{w}_t - \mathbf{u}) - \eta_t \mathbf{g}_t \\ \|\mathbf{v}_{t+1} - \mathbf{u}\|_2^2 &= \|\mathbf{w}_t - \mathbf{u}\|_2^2 - 2\eta_t \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u}\rangle + \eta_t^2 \|\mathbf{g}_t\|_2^2 \\ \|\mathbf{w}_{t+1} - \mathbf{u}\|_2^2 &\leq \|\mathbf{w}_t - \mathbf{u}\|_2^2 - 2\eta_t \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u}\rangle + \eta_t^2 \|\mathbf{g}_t\|_2^2 \\ \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u}\rangle &\leq \frac{1}{2\eta_t}(\|\mathbf{w}_t - \mathbf{u}\|_2^2 - \|\mathbf{w}_{t+1} - \mathbf{u}\|_2^2) + \frac{\eta_t}{2}\|\mathbf{g}_t\|_2^2 \end{aligned}$$

Through these inequalities and some assumptions on the norms of weight vectors and gradients, the following reformulation can be applied for any $\mathbf{u}$,

$$\begin{aligned} &\sum_{t=1}^{T} r_t \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u}\rangle \\ &\leq \sum_{t=1}^{T} \frac{r_t}{2\eta_t}(\|\mathbf{w}_t - \mathbf{u}\|_2^2 - \|\mathbf{w}_{t+1} - \mathbf{u}\|_2^2) + \frac{\eta_t r_t}{2}\|\mathbf{g}_t\|_2^2 \\ &\leq \frac{r_1}{2\eta_1}\|\mathbf{w}_1 - \mathbf{u}\|_2^2 - \frac{r_T}{2\eta_T}\|\mathbf{w}_{T+1} - \mathbf{u}\|_2^2 \\ &\quad + \frac{1}{2}\sum_{t=2}^{T}\left(\frac{r_t}{\eta_t} - \frac{r_{t-1}}{\eta_{t-1}}\right)\|\mathbf{w}_t - \mathbf{u}\|_2^2 + \frac{G^2}{2}\sum_{t=1}^{T}\eta_t r_t \\ &\leq 2R^2\left(\frac{r_1}{\eta_1} + \sum_{t=2}^{T}\left(\frac{r_t}{\eta_t} - \frac{r_{t-1}}{\eta_{t-1}}\right)\right) + \frac{G^2}{2}\sum_{t=1}^{T}\eta_t r_t \\ &= \frac{2R^2 r_T}{\eta_T} + \frac{G^2}{2}\sum_{t=1}^{T}\eta_t r_t \ . \end{aligned}$$

The assumption $\|\mathbf{g}\|_2 < G$ is used in the second inequality and $\|\mathbf{w}\|_2 < R$ is used in the third inequality.

The following inequality is derived by setting $r_t = 1 + \gamma$ and $\eta_t = c(1+\gamma)/\sqrt{t}$ for all t in the second term.

$$\begin{aligned} \text{Regret}(T) &\leq \left(\frac{2R^2\sqrt{T}}{c} + \frac{G^2(1+\gamma)^2}{2}\sum_{t=1}^{T}\frac{c}{\sqrt{t}}\right) \\ &\leq \left(\frac{2R^2}{c} + cG^2(1+\gamma)^2\right)\sqrt{T} \ . \end{aligned} \tag{20}$$

by using $\sum_{t=1}^{T}\frac{1}{\sqrt{t}} \leq 2\sqrt{T}$. When $c = \sqrt{2}R/G(1+\gamma)$, we have

$$\text{Regret}(T) \leq 2\sqrt{2}RG(1+\gamma)\sqrt{T} \ .$$

□

Next, we show the proof of Lemma 1.

Proof. When E-OGD runs in this setting, the algorithm automatically constructs a set of correctly classified examples P. These examples are sampled from $\mathcal{D}$ and chosen with an on-the-fly prediction. For simplifying the discussion, we define

$$\bar{r}(z) = \frac{1}{T}\sum_{t=1}^{T} r_t(z) \ . \tag{21}$$

Let us define $p(\mathbf{x}, y)$ as a probability density function according to a distribution $\mathcal{D}$. The set P is constructed from examples sampled from $\mathcal{D}$. For any datum in P, the datum's occurrence probability in D becomes necessarily greater than 0. The optimization formula can be reformulated according to basic probabilistic properties:

$$\begin{aligned} &E_{z\sim\mathcal{D}}\left[\ell(\mathbf{w}; z) + \bar{C}(\mathbf{w}; z)\right] \\ &= E_{z\sim\mathcal{D}}\Bigg[\ell(\mathbf{w}; z) \\ &\qquad + \frac{\gamma}{T}\sum_{t=1}^{T}\min\left(1, \sum_{s=1}^{t-1} 1_{z=z_s} 1_{y_s=\hat{y}_s}\right)\ell(\mathbf{w}; z)\Bigg] \\ &= E_{z\sim\mathcal{D}}\left[\bar{r}(z)\ell(\mathbf{w}; z)\right] \ . \end{aligned}$$

Furthermore, we can construct a new distribution $\mathcal{D}_P$ as follows: The probability function q of $\mathcal{D}_P$ is defined as

$$q(z) = \frac{\bar{r}(z)p(z)}{\int \bar{r}(z)p(z)dz} \ . \tag{22}$$

We denote the denominator of formula (22) as $H_{\mathcal{D}^T}$. In this case, the optimization problem can be reformulated according to basic probabilistic properties:

$$E_{z\sim\mathcal{D}}\left[\bar{r}(z)\ell(\mathbf{w}; z)\right] = H_{\mathcal{D}^T} E_{z\sim\mathcal{D}_P}\left[\ell(\mathbf{w}; z)\right] \ . \tag{23}$$

□

In the last, we prove Theorem 4.

Proof. First, we analyze the regret bound from round t_p to T.

$$\begin{aligned} \text{Regret}(t_p : T) &\leq \left(\frac{2R^2\sqrt{T}}{c} + \frac{G^2(1+\gamma)^2}{2}\sum_{t=t_p}^{T}\frac{c}{\sqrt{t}}\right) \\ &\leq \left(\frac{2R^2}{c} + cG^2(1+\gamma)^2\right)\sqrt{T} \\ &\quad - cG^2(1+\gamma)^2\sqrt{t_p - 1} \ . \end{aligned} \tag{24}$$

When we set $c = \sqrt{2}R/G(1+\gamma)$,

$$\text{Regret}(t_p : T) \leq \sqrt{2}RG(1+\gamma)\left(2\sqrt{T} - \sqrt{t_p - 1}\right) \ . \tag{25}$$

For simplicity, we will describe $H_{\mathcal{D}^T} E_{z\sim\mathcal{D}_P}[\ell(\cdot; z)]$ as $\mathcal{D}_P(\cdot)$. Let us take the expectation of the regret in the online learning framework with a human cognitive bias. Now let us analyze the first term and the second term of the expectation of the regret. Note that this reformulation uses Lemma

4.3. The expectation of the second term is reformulated as follows:

$$\begin{aligned}E_{\mathcal{D}^T}\left[\sum_{t=t_p}^T r_t(z_t)\ell_t(\mathbf{u})\right] &= \sum_{t=t_p}^T E_{\mathcal{D}^T}\left[r_t(z_t)\ell_t(\mathbf{u})\right] \\ &= \sum_{t=t_p}^T E_{\mathcal{D}^{t-1}} E_{\mathcal{D}^t}\left[r_t(z_t)\ell_t(\mathbf{u})|\mathcal{D}^{t-1}\right] \\ &= \sum_{t=t_p}^T E_{\mathcal{D}^{t-1}} E_{z\sim\mathcal{D}}\left[r_t(z)\ell(\mathbf{u};z)\right] \\ &= E_{\mathcal{D}^T} E_{z\sim\mathcal{D}}\left[\sum_{t=t_p}^T r_t(z)\ell(\mathbf{u};z)\right] \\ &= (T-t_p+1)E_{\mathcal{D}^T}\left[\mathcal{D}_P(\mathbf{u})\right] \ .\end{aligned}$$

The expectation of the first term can be reformulated with a similar procedure.

$$\begin{aligned}E_{\mathcal{D}^T}\left[\sum_{t=t_p}^T r_t(z_t)\ell_t(\mathbf{w}_t)\right] &= \sum_{t=t_p}^T E_{\mathcal{D}^T}\left[r_t(z_t)\ell_t(\mathbf{w}_t))\right] \\ &= \sum_{t=t_p}^T E_{\mathcal{D}^{t-1}} E_{z\sim\mathcal{D}}\left[r_t(z)\ell(\mathbf{w}_t;z)\right] \\ &= E_{\mathcal{D}^T} E_{z\sim\mathcal{D}}\left[\sum_{t=t_p}^T r_t(z)\ell(\mathbf{w}_t;z)\right] \\ &= E_{\mathcal{D}^T}\left[H_{\mathcal{D}^T} E_{z\sim\mathcal{D}_P}\left[\sum_{t=t_p}^T \ell(\mathbf{w}_t;z)\right]\right] \\ &\geq (T-t_p+1)E_{\mathcal{D}^T}\left[H_{\mathcal{D}^T} E_{z\sim\mathcal{D}_P}\left[\ell(\bar{\mathbf{w}}_t;z)\right]\right] \\ &= (T-t_p+1)E_{\mathcal{D}^T}\left[\mathcal{D}_P(\bar{\mathbf{w}})\right] \ .\end{aligned}$$

The inequality is satisfied from the convexity of loss functions. The following formula is obtained by combining these two formula with Lemma 4.3:

$$\begin{aligned}&E_{\mathcal{D}^T}\left[\mathcal{D}_P(\bar{\mathbf{w}})\right] - E_{\mathcal{D}^T}\left[\mathcal{D}_P(\mathbf{u})\right] \\ &\quad\leq \frac{1}{T-t_p+1} E_{\mathcal{D}^T}\left[\text{Regret}(t_p:T)\right] \\ &\quad\leq \frac{\sqrt{2}RG(1+\gamma)}{(\sqrt{T}-(t_p+1)/\sqrt{T})/(2-\sqrt{t_p-1}/\sqrt{T})} \ .\end{aligned}$$

□

B Another step width setting

Lemma 4. *We set the step width as follows:*

$$\eta_t = \frac{c}{\sqrt{t}}\frac{1}{(1+\gamma)^{N_t}} \tag{26}$$

where N_t is the number of wrong prediction from round 1 to round t. In this case, $(r_t/\eta_t) - (r_{t-1}/\eta_{t-1}) \geq 0$ for all $t \geq 2$.

Proof. It is straightforward to show that for any $t \geq 2$, $\eta_t < \eta_{t-1}$ is satisfied from the definition of η_t. From this property, we see that $(r_t/\eta_t) - (r_{t-1}/\eta_{t-1}) \geq 0$ is always satisfied when $r_t = 1+\gamma$ (regardless of whether $r_{t-1} = 1$ or $1+\gamma$). When $r_t = 1$, the algorithm misclassified the example at t-th round. Therefore, N_t is incremented and $r_t/\eta_t > (1+\gamma)/\eta_{t-1}$ is satisfied. In summary, $(r_t/\eta_t)-(r_{t-1}/\eta_{t-1}) \geq 0$ is always satisfied. □

The next theorem gives the regret upper bound of E-OGD.

Theorem 5. *Let $\mathbf{w}_1,\dots,\mathbf{w}_{T+1}$ be derived according to E-OGD's update rule. Assume that for all $\mathbf{w}_t$, $\|\mathbf{w}_t\|_2 \leq R$, $\|\nabla\ell_t(\mathbf{w}_t)\|_2 \leq G$. When we set $\eta_{1:T}$ as in Lemma 4 and $c = \sqrt{2}R/G$, the upper bound of regret becomes*

$$\textit{Regret}(T) \leq \sqrt{2}RG\left((1+\gamma)^{N_T+1} + (1+\gamma)\right)\sqrt{T} \ .$$

If $N_T \ll T$, the regret bound becomes sublinear. In general, the probability of mistakes gets lower when the algorithm converges to a well-performed point and N_T would not get larger.

Proof. From the proof of Theorem 3, we obtain

$$\sum_{t=1}^T r_t\langle\mathbf{g}_t, \mathbf{w}_t - \mathbf{u}\rangle \leq \frac{2R^2 r_T}{\eta_T} + \frac{G^2}{2}\sum_{t=1}^T \eta_t r_t \ . \tag{27}$$

In addition,

$$\sum_{t=1}^T \eta_t \leq \sum_{t=1}^T \frac{c}{\sqrt{t}} \leq 2c\sqrt{T} \tag{28}$$

The following inequality is derived by setting $r_t = 1+\gamma$ for all t in the second term.

$$\begin{aligned}\text{Regret}(T) &\leq (1+\gamma)\left(\frac{2R^2}{\eta_T} + \frac{G^2}{2}\sum_{t=1}^T \eta_t\right) \\ &\leq \left(\frac{2R^2(1+\gamma)^{N_T+1}}{c} + cG^2(1+\gamma)\right)\sqrt{T} \ .\end{aligned} \tag{29}$$

When $c = \sqrt{2}R/G$, we have

$$\text{Regret}(T) \leq \sqrt{2}RG\left((1+\gamma)^{N_T+1} + (1+\gamma)\right)\sqrt{T} \ .$$

□

Theorem 6. *Assume that the conditions set in Theorem 3 are satisfied and there is an integer t_p such that $r_t(z) = r_{t_p}(z)$ for any $t \geq t_p$. The following formula is satisfied for any $\mathbf{u} \in \mathcal{W}$,*

$$\begin{aligned}&E_{\mathcal{D}^T}\left[E_{z\sim\mathcal{D}_P}\left[\ell(\bar{\mathbf{w}};z)\right]\right] - E_{z\sim\mathcal{D}_P}\left[\ell(\mathbf{u};z)\right] \\ &\quad\leq \frac{\sqrt{2}RG\left((1+\gamma)^{N_T+1} + (1+\gamma)\right)}{\sqrt{T}-(t_p-1)/\sqrt{T}} \ .\end{aligned} \tag{30}$$

If $N_T \ll T$ and $t_p \ll T$, the convergence speed is $O(1/\sqrt{T})$. When the number of data is finite, there exists a constant t_p such that $r_t(z) = r_{t_p}(z)$ for any $t \geq t_p$.

Proof. First, we analyze the regret bound from round t_p to T.

$$\begin{aligned}\text{Regret}(t_p : T) &\leq (1+\gamma)\left(\frac{2R^2}{\eta_T} + \frac{G^2}{2}\sum_{t=t_p}^{T}\eta_t\right) \\ &\leq (1+\gamma)\left(\frac{2R^2(1+\gamma)^{N_T}}{c} + cG^2\right)\sqrt{T} \qquad (31)\end{aligned}$$

When $c = \sqrt{2}R/G$,

$$\text{Regret}(T) \leq \sqrt{2}RG\left((1+\gamma)^{N_T+1} + (1+\gamma)\right)\sqrt{T}\ .$$

For simplicity, we will describe $H_{\mathcal{D}^T} E_{z\sim\mathcal{D}_P}[\ell(\cdot; z)]$ as $\mathcal{D}_P(\cdot)$. By the reformulation as in the same case of Theorem 4, the expectation of the second term is reformulated as follows:

$$E_{\mathcal{D}^T}\left[\sum_{t=t_p}^{T} r_t(z_t)\ell_t(\mathbf{u})\right] = (T - t_p + 1)E_{\mathcal{D}^T}\left[\mathcal{D}_P(\mathbf{u})\right]\ .$$

The expectation of the first term can be reformulated as:

$$E_{\mathcal{D}^T}\left[\sum_{t=t_p}^{T} r_t(z_t)\ell_t(\mathbf{w}_t)\right] = (T - t_p + 1)E_{\mathcal{D}^T}\left[\mathcal{D}_P(\bar{\mathbf{w}})\right]\ .$$

The following formula is obtained by combining these formulas with Lemma 1

$$\begin{aligned}&E_{\mathcal{D}^T}\left[\mathcal{D}_P(\bar{\mathbf{w}})\right] - E_{\mathcal{D}^T}\left[\mathcal{D}_P(\mathbf{u})\right] \\ &\leq \frac{1}{T - t_p + 1}E_{\mathcal{D}^T}\left[\text{Regret}(t_p : T)\right] \\ &\leq \frac{\sqrt{2}RG\left((1+\gamma)^{N_T+1} + (1+\gamma)\right)}{\sqrt{T} - (t_p - 1)/\sqrt{T}}\ . \qquad (32)\end{aligned}$$

□

References

Aberdeen, D.; Pacovsky, O.; and Slater, A. 2010. The learning behind gmail priority inbox. In *LCCC: NIPS 2010 Workshop on Learning on Cores, Clusters and Clouds*.

Bottou, L., and Bousquet, O. 2011. The tradeoffs of large scale learning. In *Optimization for Machine Learning*. MIT Press. 351–368.

Cesa-Bianchi, N.; Conconi, A.; and Gentile, C. 2004. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory* 50(9):2050–2057.

Duchi, J. C.; Shalev-Shwartz, S.; Singer, Y.; and Tewari, A. 2010. Composite objective mirror descent. In *COLT*, 14–26. Omnipress.

Hazan, E.; Agarwal, A.; and Kale, S. 2007. Logarithmic regret algorithms for online convex optimization. *Machine Learning* 69(2-3):169–192.

Kahneman, D.; Knetsch, J.; and Thaler, R. H. 1990. Experimental tests of the endowment effect and the coase theorem. *Journal of Political Economy* 98(6):1325–48.

Karampatziakis, N., and Langford, J. 2011. Online importance weight aware updates. In *UAI*, 392–399. AUAI Press.

Langford, J., and Beygelzimer, A. 2005. Sensitive error correcting output codes. In *COLT*, 158–172. Springer-Verlag.

McMahan, H. B. 2011. Follow-the-regularized-leader and mirror descent: Equivalence theorems and l1 regularization. *Journal of Machine Learning Research* 15:525–533.

Settles, B. 2011. Closing the loop: fast, interactive semi-supervised annotation with queries on features and instances. In *EMNLP*, 1467–1478. Association for Computational Linguistics.

Shalev-Shwartz, S., and Kakade, S. M. 2008. Mind the duality gap: Logarithmic regret algorithms for online optimization. In *NIPS*, 1457–1464. MIT Press.

Shalev-Shwartz, S., and Singer, Y. 2007. A primal-dual perspective of online learning algorithms. *Machine Learning* 69:115–142.

Shalev-Shwartz, S.; Singer, Y.; Srebro, N.; and Cotter, A. 2011. Pegasos: primal estimated sub-gradient solver for svm. *Mathematical Programming* 127(1):3–30.

Shalev-Shwartz, S. 2012. Online learning and online convex optimization. *Foundations and Trends in Machine Learning* 4(2):107–194.

Thaler, R. H. 1980. Toward a positive theory of consumer choice. *Journal of Economic Behavior & Organization* 1(1):39–60.

Wang, J.; Zhao, P.; and Hoi, S. C. H. 2012. Cost-sensitive online classification. In *ICDM*, 1140–1145. IEEE Computer Society.

Xiao, L. 2010. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research* 11:2543–2596.

Yu, H.-F.; Hsieh, C.-J.; Chang, K.-W.; and Lin, C.-J. 2012. Large linear classification when data cannot fit in memory. *ACM Transactions on Knowledge Discovery from Data* 5(4):23:1–23:23.

Zadrozny, B.; Langford, J.; and Abe, N. 2003. Cost-sensitive learning by cost-proportionate example weighting. In *ICDM*, 435–442. IEEE Computer Society.

Zinkevich, M. 2003. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, 928–936. AAAI Press.