International Journal of Networking and Computing – www.ijnc.org, ISSN 2185-2847
Volume 8, Number 2, pages 153–165, July 2018

Label Estimation Method with Modifications for Unreliable Examples in Taming

Yasutake KOISHI
Graduate School of Life Science and Systems Engineering
Kyushu Institute of Technology, Kyutech
2-4 Hibikino, Wakamatsu-ku,
Kitakyushu, Fukuoka, Japan
Email: q899007y@edu.brain.kyutech.ac.jp

Shuichi ISHIDA
Advanced Manufacturing Research Institute
National Institute of Advanced Industrial Science and Technology
807-1, Shuku, Tosu, Saga, Japan

Tatsuo TABARU
Advanced Manufacturing Research Institute
National Institute of Advanced Industrial Science and Technology
807-1, Shuku, Tosu, Saga, Japan

Hiroyuki MIYAMOTO
Graduate School of Life Science and Systems Engineering
Kyushu Institute of Technology, Kyutech
2-4 Hibikino, Wakamatsu-ku,
Kitakyushu, Fukuoka, Japan

Received: February 1, 2018
Accepted: March 17, 2018
Communicated by Shuich Ichikawa

Abstract

Methods for improving learning accuracy by utilizing a plurality of data sets with different reliabilities have been studied extensively. Unreliable data sets often include data with incorrect labels, and the accuracy of learning from such data sets is thus affected. Here, we focused on a learning problem, Taming, which deals with two kinds of data sets with different reliabilities. We propose a label estimation method for use in data sets that include data with incorrect labels. The proposed method is an extension of BaggTaming, which has been proposed as a solution to Taming. We conducted experiments to verify the effectiveness of the proposed method by using a benchmark data set in which the labels were intentionally changed to make them incorrect. We confirmed that learning accuracy could be improved by using the proposed method and data sets with modified labels.

Keywords: Ensemble learning, Bagging, Label estimation, Taming

1 Introduction

In recent years there have been attempts to obtain new knowledge from large scale-data sets by using machine learning. Deep learning has attracted particular attention as a learning method for acquiring meaningful knowledge from large-scale data sets. By using deep learning, it becomes possible to learn with high accuracy from large amounts of labeled data ; the resulting performance exceeds those of existing methods in various fields. For example, speech recognition using DNN-HMM has realized a reduction in error rate of more than 10% compared with those of existing methods [1]. These techniques are used for voice operation of today's mobile phones. In image recognition using DCNN, the error rate has been reduced by 10% or more compared with those of other methods [2]. Thanks to the success in these fields, demand for new knowledge from large-scale data is increasing. However, there are some obstacles to utilizing deep learning in the field, for example at manufacturing sites. A problem that often arises in reality is difficulty in managing the large-scale data that are acquired. To obtain a reliable learning result, the data used for learning must have a high degree of reliability, such that labeling is accurate. Furthermore, various data must be acquired to judge various products. To obtain highly reliable knowledge, we need to learn from a wide variety of data with high reliability. However, to accurately label all of a large number of data is not realistic, because it is extremely expensive in terms of human resources and time. In actual environments, such as at manufacturing sites, it is also expensive in terms of cost. To lower the cost of data collection it is possible to obtain web-based data-for example, by web scraping. The use of web scraping makes the cost of data collection very low. However, because there are many subjective opinions and items of false information mixed in with the data on the web, it is very difficult to collect only highly reliable data. Therefore, the trade-off between data quality and quantity is a big problem.

To resolve this trade-off, methods of dealing with data sets with different reliabilities have been studied widely. Taming [3] has been proposed as a method of handling data sets with different reliabilities. Taming deals with two types of data : tame data and wild data. Tame data are labeled by using strict standards. Therefore, tame data sets are highly reliable. On the other hand, strictly no label management is used in the case of wild data. In other words, training examples with correct labels and training examples with incorrect labels are mixed in together in the case of wild data. It is therefore not known which items of wild data are training examples with correct labels. It is difficult to secure an abundance of training examples from tame data, because the costs of managing these data are high. On the other hand, it is easy to secure an abundance of training examples from wild data. The aim of Taming is to achieve high learning accuracy by using wild data and tame data in combination. Taming deals with the situation in which there are

- A small number of training example sets that have reliable labels and take a long time to collect; and
- An abundance of training example sets that have unreliable labels and take little time to collect.

This situation occurs frequently, not only in web scraping or in actual environments such as manufacturing, but also in natural science experiments. BaggTaming has been proposed as a solution to Taming. BaggTaming is a method based on the ensemble learning technique of Bagging [4]. BaggTaming creates multiple weak classifiers by bootstrap sampling from wild data. The weak classifiers are filtered by using the accuracy rate on the tame data. BaggTaming improves learning accuracy by aggregating the adopted classifiers. To achieve high learning accuracy, it is necessary to aggregate a variety of weak classifiers. When learning is performed by using data that have incorrect labels, learning accuracy is adversely affected. If we can modify wild data with incorrect labels, then we can learn from a greater variety of training examples. Therefore, we can expect to improve the learning accuracy of BaggTaming by modifying labels.

Here, we propose a method of making effective use of wild data. Specifically, we 1) estimate some training examples by using wild data with incorrect labels; and 2) modify the labels of the estimated training examples.

The paper is organized as follows: in section 2, we describe Taming and BaggTaming proposed as solution Taming. In section 3, we extend the BaggTaming algorithm described in section 2 and propose a method to identify and modify training examples with incorrect labels from wild data. In section 4, we evaluate the effectiveness of the proposed method by experiments using a benchmark dataset in both scenes of binary and multi-value classification problems. In section 5, we survey the related works for Taming. In section 6, we present the summary of the paper and describe some possible future works.

2 Taming-Learning problem

2.1 Taming

Taming [3] has been proposed by Kamishima et al. as a learning problem that combines two kinds of training examples, namely tame data and wild data. Training examples of tame data are carefully labeled, with strict standards; these examples and their labels are thus highly reliable. Here, we call the label associated with the target concept that we want to learn the correct label. Conversely, a label associated with a criterion different from the target concept is called an incorrect label.

Unlike in the case of tame data, training examples of wild data have labels that are not strictly managed. In other words, two types of training examplesthose with correct labels and those with incorrect labelsare mixed together in wild data. It is not known which training examples in the wild data are correctly labeled. Furthermore, it is also not known how many of the data are correct. It is difficult to prepare an abundance of tame data, but it is easy to prepare an abundance of wild data. The aim of Taming is to make it possible to achieve high learning accuracy by using these data in combination.

2.2 BaggTaming: the Solution to Taming

BaggTamingan algorithm based on the principle of Bagginghas been proposed as a solution to Taming. In BaggTaming, the first step is bootstrap sampling from wild data, and a plurality of classifiers is generated. The learned classifiers are filtered by the accuracy rate of the tame data, and a judgment on adoption or non-adoption is made. The final result is computed by aggregation of the adopted classifier. The algorithm and points reached in BaggTaming are shown in Figure 1.

Normally, machine learning is performed only by using highly reliable examples. Generally, examples with high reliability are expensive to collect and manage, so it is difficult to secure an abundance of such examples. Examples with high reliability are called tame data in Taming. The Figure 2 is a pseudo data set obtained by adding a random number to the XOR distribution in the two-dimensional feature amount. Red represents positive examples, and blue represents negative examples. Circle represents tame data, and cross represents wild data. When learning is performed by using only tame data, the decision boundary shown in Figure 2a is plotted. This boundary is fitting for only learning data, however it is different from an intuitive decision boundary. Also, because the decision boundary is too close to the learning examples, the generalization performance in regard to unknown data is low.

Next, the decision boundary in the case of learning using an abundance of secured data is shown in the Figure 2b. When many data are used, correct decision boundaries can be obtained more intuitively than when learning from less data. By using more data, BaggTaming aims to obtain decision boundaries with high generalization performance.

In ensemble learning, the planning of improvements in generalization performance is based on biasvariance theory [5]. Biasvariance theory is expressed in terms of an error derived from each classifier, a variance representing the error derived from the training example, and a covariance term that models the correlation between individual learners. According to biasvariance theory, the squared error $Err(H)$ of ensemble learning (H) for classifier T is calculated as follows by using the bias $b(H)$, variance $v(H)$, and covariance $c(H)$:

$$Err(H) = b(H)^2 + \frac{1}{T}v(H) + (1 - \frac{1}{T})c(H) \tag{1}$$

1. Calculate the Accuracy rate Acc_{T} using the standard classifier learned from tame data.
2. Bootstrap sampling from wild data and generate N data sets as $W_1 \cdots W_{\mathrm{N}}$.
3. a = 0
4. for i in 1:N
   - Generate a classifier f_{i} learned from W_{i}.
   - Calculate the Accuracy rate Acc_{i} using the classifier fi for tame data.
   - if $Acc_{\mathrm{i}} > Acc_{\mathrm{T}}$

     a = a + 1

     Adopt f_{i} as F_{a}
   - else

     Discard f_{i}
5. Calculate the final result by aggregating the classifiers $F_1 \cdots F_{\mathrm{a}}$.

Figure 1: Algorithm of BaggTaming

Where,

$$b(H) = \frac{1}{T}\sum_{i=1}^{T}(\mathbb{E}[h_i] - f) \tag{2}$$

$$v(H) = \frac{1}{T}\sum_{i=1}^{T}\mathbb{E}(h_i - \mathbb{E}[h_i])^2 \tag{3}$$

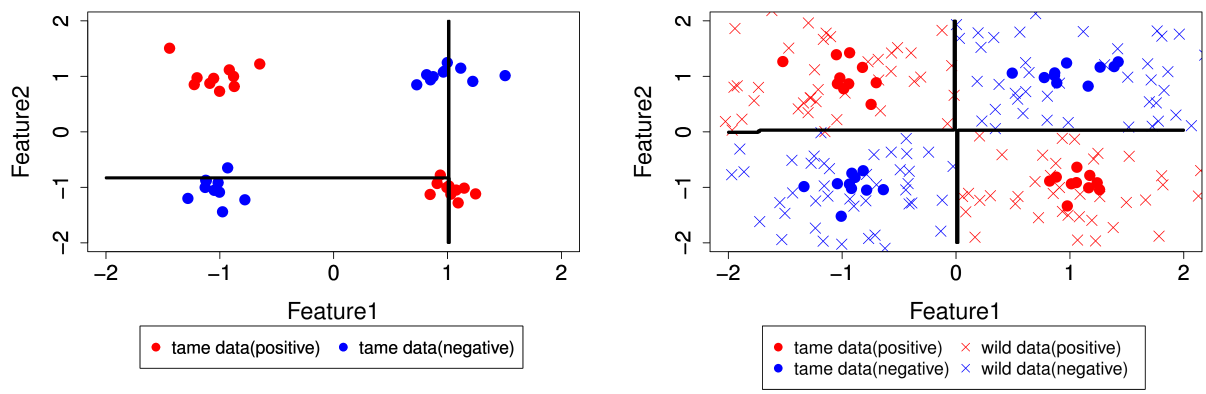
$$c(H) = \frac{1}{T(T-1)}\sum_{i=1}^{T}\sum_{j=1}^{T}\mathbb{E}(h_i - \mathbb{E}[h_i])\mathbb{E}(h_j - \mathbb{E}[h_j]) \tag{4}$$

Here, $\mathbb{E}[h]$ represents an expected value at which the function f becomes h. From equation (1), we see that the squared error of ensemble learning depends strongly on the covariance term modeling the correlation between individual learners. Therefore, by constructing more diverse learners, it is possible to reduce the squared error of ensemble learning. BaggTaming generates more diverse learners by learning from wild data, which contain a more diverse set of training examples than tame data. Furthermore, training examples with incorrect labels are included in the wild data, so if the data are aggregated by using all learning instruments, the frequency of errors will increase. Therefore, if filtering using the accuracy rate on the tame data is applied, only those classifiers effective for learning tame data will be used for learning.

3 Proposed Method

3.1 Motivation

Through bootstrap sampling of wild data, BaggTaming has succeeded in generating a greater variety of classifier than learned only from tame data. However, the classifiers learned from examples with incorrect labels included in the wild data have a lower accuracy rate. When such a classifier is used, the learning accuracy decreases. Therefore, by using the accuracy rate of the tame data to filter the learners learned from the wild data, the deterioration of learning accuracy is suppressed. However,



(a) Only tame data

(b) Leaned wild data

Figure 2: Taming's concept : (a) shows the decision boundary learned from tame data. (b) shows the decision boundary learned from wild data.

if there are many training examples with incorrect labels included in the wild data, we cannot make full use of the benefits of the wild data (such as the opportunity to learn from a large number of training examples). Therefore, the performance of BaggTaming in improving learning accuracy is lowered. It would be possible to encourage the use of wild data if appropriate labels were to be given to those training examples that were incorrectly labeled and were included in the wild data.

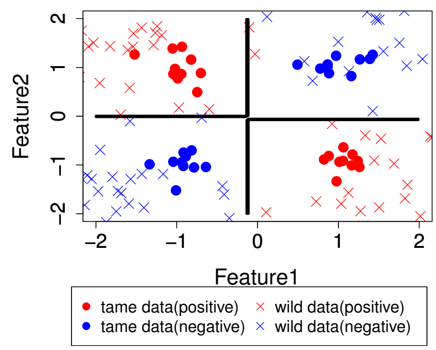
Here, we proposed a method of modifying training examples with incorrect labels in wild data by expanding BaggTaming. In the proposed method, we focus on classifiers that have been rejected. For such classifiers, the accuracy rate of tame data is bad. There are two reasons for this. One is the situation in which the training examples from bootstrap sampling of wild data are biased. The other is the situation in which the training examples from bootstrap sampling of wild data include many with incorrect labels.

As an example, consider the situation in which tame data are distributed as shown in Figure 3. As a simple example, we take the classification of two classes of two-dimensional feature quantities as in the Figure 2. When the wild data are uniformly sampled, the discrimination boundary learned from the sampled wild data is shown by the solid line in the Figure 3a. As can be seen, the classification of tame data works well.

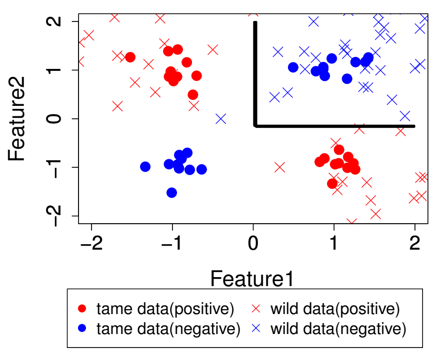
Next, consider the situation in which the sampled wild data are biased. In the example shown in the Figure 3b, the positive example of tame data has two sets, whereas the positive example of sampled wild data is biased toward one side. In this case, the discrimination boundary learned from the sampled wild data is shown by the solid line in the Figure 3b. Despite all the labels of the wild data being correct, it can be seen that positive examples and negative examples are not well classified. Although this example shows an extreme situation, the same can be said even if the data dimension and class label increase. Therefore, if there is bias in the sampled wild data, the accuracy rate of the tame data will not improve, even if the labels of the wild data are correct.

Consider the case in which many of the training examples bootstrap-sampled from the wild data have incorrect labels. The Figure 3c shows a case where approximately half of the training examples with incorrect labels are included in the sampled wild data. As with the phenomenon called "negative transfer" [6] in transfer learning, this may have occurred because we have been unable to correctly learn the target concept that we wanted to learn from the wild data. Thus the accuracy rate of the tame data has decreased because of "negative transfer".

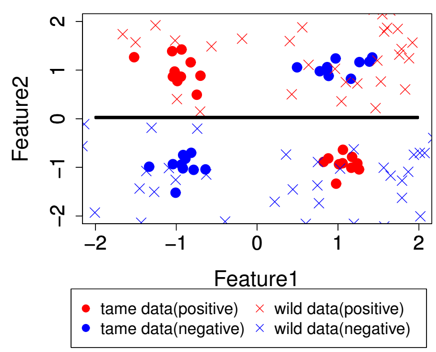
The most noteworthy problem is that training examples with correct labels and training examples with incorrect labels coexist in learners that have been rejected. Therefore, an algorithm that modifies only training examples with incorrect labels is desirable.



(a) Normally



(b) Biased labels



(c) Include incorrect labels

Figure 3: Decision boundary obtained from wild data : (a) represents a state in which wild data is evenly distribute. (b) represents a state in which wild data is biased toward one side. (c) indicates that many wild data have incorrect labels.

In the proposed method, training examples of learners that have been rejected are preserved, and labels are estimated for them in an attempt at label correction. As a result, training examples with correct labels are not changed, and only training examples with labels considered to be incorrect can be modified. Learning from label-modified training examples in wild data makes it possible to utilize more training examples than with the conventional method, BaggTaming. In addition, wild data can be corrected efficiently by investigating the training examples used for learning classifiers that have been rejected. Therefore, it is possible to construct a variety of classifiers and to thus improve learning accuracy.

3.2 Algorithm

Here, we explain the specific flow of the proposed method. First, similar to what is done in the conventional method, BaggTaming, bootstrap sampling is performed on wild data to create multiple classifiers. The accuracy rate for the tame data of each classifier is calculated, and an adoption or non-adoption judgment is made. In BaggTaming, classifiers that are rejected are discarded and do not affect future procedures. In contrast, in the proposed method, a set of training examples used for the learning of learners that have become unused is stored. Next, label estimation is performed on these sets of stored training examples through aggregation of the adopted learners. Aggregation of adopted learners better expresses the target concept that we want to learn, because it can reduce the squared error with respect to the tame data, compared with that of the classifier learned from the tame data by the ensemble.

1. Calculate the Accuracy rate Acc_{T} using the standard classifier learned from tame data.
2. Bootstrap sampling from wild data and generate N data sets as $W_1 \cdots W_{\mathrm{N}}$.
3. $\mathrm{a} = 0, \mathrm{d} = 0$
4. for i in 1:N
   - Generate a classifier f_{i} learned from W_{i}.
   - Calculate the Accuracy rate Acc_{i} using the classifier fi for tame data.
   - if $Acc_{\mathrm{i}} > Acc_{\mathrm{T}}$
     $\mathrm{a} = \mathrm{a} + 1$
     Adopt f_{i} as F_{a}
   - else
     $\mathrm{d} = \mathrm{d} + 1$
     discardf_{i}**, save** W_{i} as W_{d}
5. Label estimate of $W_{\mathrm{f}} = \{W_1 \cdots W_{\mathrm{d}}\}$ by aggregation of $F_1 \cdots F_{\mathrm{a}}$
6. Learning again using $W_1 \cdots W_{\mathrm{N}}$ given the estimated label, and repeat steps 3) to 4)
7. Calculate the final result by aggregating the classifiers $F_1 \cdots F_{\mathrm{a}}$.

Figure 4: Algorithm of proposed method

Therefore, by performing label estimation using aggregation of the adopted learners, it is possible to label by using criteria close to the target concept. Next, all learners are discarded, and learning is done again using the wild data including those training examples in which the labels have been modified. The learned classifiers are then filtered by using the accuracy rate of the tame data and a judgment of adoption or non-adoption is made. Thus, by performing aggregation of the adopted classifiers, the final result is calculated. The specific algorithm of the proposed method is shown in Figure 4.

4 Experiment

To confirm the effectiveness of the proposed method, we performed experiments using a benchmark data set. For this purpose, a quantitative data set is desirable; we used the abalone data set [7]. This data set summarizes the physical characteristics and age of abalone. The physical features are 7 dimensions. Items measured as physical features, together with age ranges, are shown in Table 2. Age is classified by using physical features as input variables. Acc_{T} is obtained by using the leave-one-out method. The number of data extracted from the wild data differs with the design, depending on the data set. However, because the enough sampling size is usually unknown, it is set to 50% of the number of tame data, N_{T}, as an intermediate value. A decision tree is used for learning. The number of samplings is set to 100 times. As a criterion for adopting classifiers, we use the accuracy rate on training data (tame data).

The calculation environment used for experiments is as shown in the table 1.

4.1 Binary classification

We examined the effectiveness of the proposed method in binary classification. The above data set was classified as binary and the proposed method was applied. The data set is shown in the table.

Table 1: Calculation environment used for experiments

CPU	Core i-7
OS	windows 10 64bit
Memory	16GB

Table 2: Abalone dataset

Name	Data Type	Description
Sex	nominal	M,F,and I(infant)
Length	continuous	Longest shell measurement
Diameter	continuous	perpendicular to length
Height	continuous	with meat in shell
Whole weight	continuous	whole abalone
Shucked weight	continuous	weight of meat
Viscera weight	continuous	gut weight (after bleeding)
Shell weight	continuous	after being dried
Rings	integer	the age in year

This data set is classified as tame data and wild data. First, training examples of tame data were randomly extracted from the data set. The number of tame data was $N_{\mathrm{T}} = \{40, 60, 80, 100\}$. Next, training examples remaining in the data set were taken as wild data. Some data were extracted randomly from the wild data. The labels of the extracted training examples were changed, and the relabeled sample were incorporated back into the wild data set. Here, we define training examples after their labels have been changed as intentionally modified label data. Intentionally modified label data accounted for $F_{\mathrm{r}} = \{20\%, 40\%, 60\%\}$ of the wild data. The proposed method was then applied to these data sets.

First, we examined whether the label changes in the proposed method worked effectively. Table 4 indicates the correct answer rate when the estimated label and the original label were compared. In Table 4, all patterns had accuracy rates exceeding 70%. Also, as the number of F_{r} increased, the accuracy rates of label estimation decreased. This finding was associated with the type of classifier used for label estimation. The classifier was learned from the training examples that were bootstrap-sampled from the wild data according to the BaggTaming process. The decrease in label estimation accuracy likely occurred because it becomes difficult to get the target concept according to increase F_{r}.

Next, we considered the validity of the wild data, the labels of which we had changed by using the proposed method. If the proposed method works effectively, we can expect BaggTaming using wild data after label correction to have a higher rate of detection of tame data than before. Therefore, we can verify the effectiveness of the proposed method by learning using wild data after application of the proposed method. Table 5 shows the rates of accuracy of differentiation of tame data from wild data before and after we applied the proposed method. Table 5 revealed that the proposed method improved the rate of accuracy of discrimination of tame data in many cases. The average execution time of each experiment is shown in the table 6.

Table 3: Binary dataset

Original labels	Number of data	New label
1∼9	2096	1
10∼29	2081	2

Table 4: Label estimate accuracy rate

		N_T			
		40	60	80	100
	20%	71.22	72.64	72.71	75.70
F_r	40%	70.93	72.47	72.44	74.43
	60%	70.56	72.06	71.13	71.64

Table 5: Experimental result of binary classification

N_T	F_r	Acc_T	Conventional method	Proposed method
40	20%	75.00	80.00	82.50
	40%	57.50	67.50	75.00
	60%	72.50	77.50	82.50
60	20%	76.67	81.67	81.67
	40%	61.67	63.33	68.33
	60%	70.00	75.00	81.67
80	20%	77.50	82.50	85.00
	40%	73.75	76.25	76.25
	60%	71.25	71.25	75.00
100	20%	62.00	70.00	74.00
	40%	66.00	72.00	76.00
	60%	70.00	73.00	78.00

Table 6: execution time of binary classification [sec]

N_T	F_r	Conventional method	Proposed method
40	20%	3.819	6.417
	40%	3.653	7.040
	60%	3.607	5.934
60	20%	5.292	8.374
	40%	5.498	8.188
	60%	5.541	7.202
80	20%	6.999	10.346
	40%	7.128	9.644
	60%	7.326	8.770
100	20%	9.301	12.722
	40%	9.402	11.870
	60%	9.478	10.632

Table 7: Multi-class dataset

Original labels	Number of data	New label
1∼8	1407	1
9∼10	1323	2
11∼29	1447	3

Table 8: Label estimate accuracy rate

		N_T		
		60	90	120
F_r	20%	57.16	60.09	61.08
	40%	56.87	58.41	59.31
	60%	54.69	55.60	55.91

4.2 Multi-class Classification

We then examined the effectiveness of the proposed method for multi-class classification. We classified the abalone data set mentioned above into multi-class values. We applied three types of labels to verify the effectiveness of the proposed method. The data set is shown in Table 7.

First, training examples of tame data were randomly extracted from the data set. The number of tame data was $N_\mathrm{T} = \{60, 90, 120\}$. Next, training examples remaining in the data set were taken as wild data. Some data were randomly extracted from these wild data; the labels of these extracted training examples were then changed, and the data were incorporated back into the wild data. The labels were completely randomly changed. Intentionally modified label data accounted for $F_\mathrm{r} = \{20\%, 40\%, 60\%\}$ of the wild data.

The proposed method was applied to these data sets. As in the binary classification, we then examined the label estimation accuracy of the proposed method. Table 8 shows the accuracy rate in a comparison of the estimated labels and the original labels. More than half of the accuracy rates are shown for all patterns. In this experiment, we divided into three class, and the rate of each label were same; thus the rate was higher than if the labels were estimated randomly. Also, as with the binary classification, the accuracy rate of label estimation decreased as F_r increased. As in the case of binary classification, label estimation accuracy likely decreased because as the number of F_r increased it became more difficult to get the same concept as for tame data.

Next, we considered the validity of the wild data, the labels of which had been changed by using the proposed method. As with binary classification, we expected that BaggTaming using wild data after label correction would have a higher accuracy rate for discrimination of tame data than before if the proposed method were to work effectively. Table 9 shows the rates of accurate discrimination of tame data when wild data before and after label estimation were used. In many cases, the rate of accurate discrimination of tame data improved. The average execution time of each experiment is shown in the table 10.

4.3 Discussion

We compared the effects of the proposed method in a binary classification problem and a multi-class classification problem. First, we discuss the accuracy of label estimation. In the binary classification, we could correctly estimate more than 70% of labels, but in the multi-class classification the accuracy rate dropped to about 50% of labels. The likely reason for the decrease in label estimation accuracy in the multi-class classification is that the result depended on the correct discrimination rate of the classifier adopted in BaggTaming.

Next, we consider the rate of correct discrimination of tame data when we learned by using wild data after label correction. In the binary classification problem, the rate of accurate discrimination of tame data improved by up to 7.5 points between before and after label change. On the other hand, in the multi-class classification problem, the improvement was only 4.5 points. This was likely due

Table 9: Experimental result of multi-class classification

N_{T}	F_{r}	Acc_{T}	Conventional method	Proposed method
60	20%	48.90	51.10	54.40
	40%	58.90	68.90	70.00
	60%	58.90	61.10	65.60
90	20%	60.00	64.40	66.70
	40%	71.10	77.80	77.80
	60%	60.00	62.20	64.40
120	20%	56.10	66.70	68.20
	40%	50.00	60.60	60.60
	60%	57.60	57.60	59.10

Table 10: execution time of multi-class classification [sec]

N_{T}	F_{r}	Conventional method	Proposed method
60	20%	5.172	7.203
	40%	5.446	8.351
	60%	5.596	7.626
90	20%	8.817	12.278
	40%	9.117	11.808
	60%	9.260	11.607
120	20%	12.344	15.712
	40%	12.916	15.348
	60%	11.871	13.021

to differences in label estimation accuracy: in the multi-class classification problem, the accuracy of label estimation was lower than that in the binary classification problem, so the classification accuracy of the proposed method was lower than in the binary classification problem.

Thus the final result of the proposed method also depends on the BaggTaming classifier. The rate of accurate detection of tame data using BaggTaming classifiers can be improved by optimizing the sampling and adjusting the criteria used to adopt classifiers. However, in many cases this optimization is a trade-off with generalization performance. Bagging is greatly influenced by ensemble sizethat is, the number of learning devices [8]. It is possible that both the proposed method and BaggTaming could not secure enough classifiers. This is because selectable classifiers are used selectively. In future work, we intend to design a method that takes into account generalization performance.

Finally, we consider the execution time of the conventional and proposed method. According to the conventional method, the execution time also increases as N_{T} increases. In the conventional method, since the sampling size is set to $N_{\mathrm{T}}/2$, the larger N_{T} is, the longer it takes time to calculate classifiers. There is no correlation between F_{r} and execution time. In the proposed method,The execution time also increases as the N_{T} increases. However, unlike the conventional method, the proposed method shortens the execution time with an increase in F_{r} and shortens it by 2 seconds at the maximum. Since the proposed method modified labels by classifiers adopted in the conventional method, the execution time gets faster as the number of classifiers adopted becomes small. Therefore, the smaller the number of wild data available in the conventional method, the shorter the execution time of the proposed method.

5 Related works

As with the Taming described here, there are a number of learning problems that deal with training example sets with different properties.

Semi-supervised learning [9] is a problem of learning using both labeled training example sets and unlabeled training example sets. Many methods of semi-supervised learning assume class labels based on models for unlabeled training examples. In many cases, class labels are determined by assuming the distribution of data. In semi-supervised learning, the point of dealing with unlabeled training examples clearly differs from that of Taming. In addition, self-training [10] a semi-supervised learning methodclassifies unlabeled training cases by using classifiers learned from labeled training example sets. Indices called confidence predictions are predicted for classified unlabeled training examples; unlabeled training examples with high confidence predictions are considered as labeled training examples, and learning is then conducted again. In terms of estimating labels, self-training involves learning from reliable, labeled training examples, whereas the proposed method learns from unreliable wild data. There is thus a difference.

Active learning [11] a method used to deal with situations in which there are few reliable data. Supervised data are selected in such a way as to exert the maximum effect when teacher data are created. Pool-based active learning [12], which is a typical release of active learning, pools an abundance of unlabeled data and selectively supervises the most effective data for learning in the current model, thereby reducing the cost of labeling. It is similar to Taming in that it selectively uses data useful for learning from among a large amount of unreliable data but uses unlabeled data.

As with Taming, transfer learning [13] is a method of dealing with multiple training example sets. In transfer learning, we deal with a training example set called the target domain and a training example set called the source domain. The problem presented by the target domain and that of the source domain are labeled by a process called induction transfer learning [14]. In induction transfer learning, the target domain is a problem we want to solve now, whereas the source domain is similar to the problem we want to solve, but it is not the same. In the source domain, sufficient data for solving the problem are secured, but enough data for solving the problem are not prepared in the case of the target domain. The goal of induction transfer learning is to transfer the knowledge acquired from the source domain and thus improve learning accuracy in the case of the target domain. The attachment of labels to each target domain and original domain in induction transfer learning is similar to the process used in Taming. However, in induction transfer learning, the target domain and source domain are given labels for different problems, whereas Taming shows the tame data, and the wild data are included as training examples representing problems to be solved. Therefore, inductive transfer learning treats similar distributions in the target domain and source domain, but essentially treats different distribution, target distribution of domesticated learning is the same for tame data and wild data. In addition, in induction transfer learning, the target domain and source domain are all given correct labels for each problem, whereas in Taming, wild data labels included as incorrect labels are decisively different.

6 Conclusion

Here, we proposed a method focusing on Taming that aimed to correct incorrect labels in wild data. Our experiments dealt with both binary classification problems and multilevel classification problems. To determine the validity of the proposed method, we experimented with a benchmark dataset and confirmed that the labels of wild data could be modified. Furthermore, we confirmed that, by using wild data, the labels of which were modified by using the proposed method, we were able to improve the accuracy of detection of tame data.

In future we intend to improve the accuracy of label estimation. Label estimation is performed by aggregating classifiers, as obtained in this paper by BaggTaming. Therefore, label estimation accuracy depends on the learning accuracy of BaggTaming. By improving the learning accuracy of BaggTaming, we expect to improve label estimation accuracy. As a method of improving the learning accuracy of BaggTaming, it may be possible to change the criteria of adoption of weak classifiers and to appropriately design the sampling size. Adjustment of label change criteria is also considered effective.

References

[1] Seide, F., Li, G., Yu D. Conversational speech transcription using context-dependent deep neural networks. In Twelfth Annual Conference of the International Speech Communication Association, 2011.

[2] Krizhevsky, A., Sutskever, I., Hinton, G. E. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097-1105, 2012

[3] Kamishima,T., Hamasaki, M., and Akaho S. BaggTaming Learning from Wild and Tame Data, ECMLPKDD2008 Workshop: Wikis, Blogs, Bookmarking Tools Mining the Web 2.0, 2008.

[4] Breiman, L. Bagging Predictors, Machine Learning, volume 24, pages 123-140, 1996

[5] Geman, S., Bienenstock, E., Doursat, R. Neural networks and the bias/variance dilemma. Neural computation, 4.1, pages 1-58, 1992

[6] Rosenstein, M. T., Marx, Z., Kaelbling, L. P., Dietterich, T. G. To transfer or not to transfer. In NIPS 2005 Workshop on Transfer Learning, volume 898, 2005

[7] "https://archive.ics.uci.edu/ml/datasets/abalone", 30. May. 2018 accessed.

[8] Zhou, Z, H. Ensemble Methods - Foundations and Algorithms, CRC Press, pages47-52.

[9] Xiaojin, Z. Semi-Supervised Learning Tutorial, ICML, 2007

[10] Yarowsky, D. Unsupervised word sense disambiguation rivaling supervised methods, Proceedings of the 33rd annual meeting on Association for Computational Linguistics ACL 95, pages 189-196, 1995.

[11] Settles, B. Active learning literature survey. University of Wisconsin, Madison, 52, 11, 2011

[12] McCallumzy, A. K., Nigamy, K. Employing EM and pool-based active learning for text classification. In Proc. International Conference on Machine Learning (ICML) pages 359-367, 1998.

[13] Caruana, R. Multitask Learning, Machine Learning, volume.28, pages 41-75,1997.

[14] S. J. Pan, Q. Yang. A survey on transfer learning, IEEE Transactions on knowledge and data engineering, volume. 22, pages 1345-1359,2010.