Machine Learning
https://doi.org/10.1007/s10994-022-06183-y

Machine truth serum: a surprisingly popular approach to improving ensemble methods

Tianyi Luo[1] · **Yang Liu**[1]

Received: 4 November 2021 / Revised: 11 February 2022 / Accepted: 9 May 2022

Abstract
Wisdom of the crowd (Surowiecki, 2005a) disclosed a striking fact that the majority voting answer from a crowd is usually more accurate than a few individual experts. The same story is observed in machine learning - ensemble methods (Dietterich, 2000) leverage this idea to exploit multiple machine learning algorithms in various settings e.g., supervised learning and semi-supervised learning to achieve better performance by aggregating the predictions of different algorithms than that obtained from any constituent algorithm alone. Nonetheless, the existing aggregating rule would fail when the majority answer of all the constituent algorithms is more likely to be wrong. In this paper, we extend the idea proposed in *Bayesian Truth Serum* (Prelec, 2004) that "a surprisingly more popular answer is more likely to be the true answer instead of the majority one" to supervised classification further improved by ensemble final predictions method and semi-supervised classification (e.g., MixMatch (Berthelot et al., 2019)) enhanced by ensemble data augmentations method. The challenge for us is to define or detect when an answer should be considered as being "surprising". We present two machine learning aided methods which can reveal the truth when the minority instead of majority has the true answer on both settings of supervised and semi-supervised classification problems. We name our proposed method the Machine Truth Serum. Our experiments on a set of classification tasks (image, text, etc.) show that the classification performance can be further improved by applying Machine Truth Serum in the ensemble final predictions step (supervised) and in the ensemble data augmentations step (semi-supervised).

Keywords Ensemble methods · Supervised classification · Semi-supervised classification · Data augmentation · Bayesian truth serum

Editors: Bo Han, Tongliang Liu, Quanming Yao, Mingming Gong, Gang Niu, Ivor W. Tsang, Masashi Sugiyama.

✉ Tianyi Luo
tluo6@ucsc.edu

Yang Liu
yangliu@ucsc.edu

[1] Computer Science and Engineering, UC Santa Cruz, Santa Cruz, USA

Published online: 12 July 2022

1 Introduction

Wisdom of the crowd reveals the power of aggregating opinion from a diverse groups rather than a few individuals. Although the idea was proposed for mainly aggregating human judgements, it has been successfully applied in the context of machine learning (ML). Ensemble methods was proposed to further improve the performance in various settings e.g., supervised learning (SL) and semi-supervised (SSL) by combining several component learning models trained from different categories to composite a system instead of utilizing a single one (Dietterich, 2000). More specifically, ensemble methods can be utilized to enhance the final prediction results in SL and applied to generate better pseudo labels based on data augmentations of unsupervised data in SSL (e.g., MixMatch (Berthelot et al., 2019)). Ensemble methods have achieved exceptionally satisfactory performance in some international ML competitions such as Kaggle, and KDD-Cups.

The most popular way of aggregating in ensemble methods is majority voting rule. One classical example is Random Forest (Ho, 1995), which outputs the majority answer from multiple trained decision trees. Inference methods have been applied to obtain better aggregation that aims to outperform the majority voting rule (Raykar et al., 2010; Zhang et al., 2014; Liu et al., 2012; Zhou et al., 2012, 2014). These inference methods usually perform joint inference under the homogeneous assumption of certain hidden models over a large amount of data points.

However, all the methods mentioned above are based on the same assumption that the majority answer is more likely to be correct. For more sophisticated inference models, the majority answer is mostly likely to initiate the inference when the algorithm has no prior information. While enjoying the assumption that the majority answer is tending to be correct, it is questionable in the settings where special knowledge is required to get the truth answer, but this kind of knowledge is only owned by few individual experts (when they are not widely shared) (Chen et al., 2004; Simmons et al., 2010; Prelec et al., 2017). Echoing to the above problem in the setting of aggregating human judgements, the similar challenge is faced when we need to aggregate the predictions of different learning methods in ML. For example, we have a state-of-the-art (SOTA) deep learning (Goodfellow et al., 2016) classification model which obtains the best performance among the learning methods utilized in the ensemble model. For some data point, the classification result of this SOTA deep learning model may be the correct minority. Apply the majority rule on this data point will lead to a wrong answer.

We aim to explore whether we can obtain better aggregation results than the majority voting rule even when the majority answer is wrong. We also target a method that can conduct the inference on each data point separately without having the homogeneous assumptions over a massive dataset.

The question sounds unlikely to resolve at a first look, but we are inspired by the seminal work *Bayesian Truth Serum* (BTS) (Prelec, 2004; Prelec et al., 2017) which approached this question in the setting of incentivizing and aggregating truthful human judgements. The core idea behind BTS is simple and elegant: the correctness of an answer cannot be guaranteed based on its popularity (having a higher posterior), but rather whether it is "surprisingly" popular or not. The answer having a higher posterior than its prior is taken as being "surprisingly" popular and should be considered as the true answer. Prelec et al. (2017) also argued that by eliciting a peer prediction information, which is defined as the fraction of "how many other people would agree with you", an informative prior can be constructed to compared with the majority-voted posterior. BTS can be operated on each

single question separately, without leveraging a certain homogeneity assumption through seeing a large number of similar tasks.

In this paper, we make a connection between these two seemingly irrelevant topics, and extend the key idea in *Bayesian Truth Serum* to further improve the performance of ensemble learning methods in the context of supervised and semi-supervised classification. The challenge is that we would not be able to elicit a belief from a classifier on "how many other classifiers would agree with themselves", which renders the task of computing the prior difficult. We proposed two ML aided algorithms to mimic the procedure of reporting the peer prediction information, which we jointly name as *Machine Truth Serum* (MTS). In Heuristic Machine Truth Serum (HMTS), we pair each baseline classifier (an agent) with a regressor model, which is trained to predict the peer prediction information using a processed training dataset. With the predictions from the regressors, we will be able to apply the idea of BTS to decide on whether adopting the minority as the answer via comparing the prior (computed using the regressor) and the posterior for each data point. In Discriminative Machine Truth Serum (DMTS), we directly train one classifier to predict whether adopting the minority as the answer or not. We applied our proposed MTS methods in both supervised and semi-supervised classification tasks. In supervised classification task, we adopted MTS methods in the ensemble final predictions step. For semi-supervised classification tasks, MixMatch (Berthelot et al., 2019) and MixText (Chen et al., 2020) are considered as the ensemble baseline methods and MTS are utilized to generate better pseudo labels for unsupervised data based on ensemble data augmentation method. As for the training complexity of our algorithm, the training time of HMTS is linear in the number of label classes because of the training of extra regressors. DMTS will only need to train one additional classifier and both the training and the running time are almost the same as the basic majority voting algorithm. Therefore our proposed methods are very practical to implement and run.

Our contributions summarize as follows: (1) We propose Heuristic Machine Truth Serum (HMTS) and Discriminative Machine Truth Serum (DMTS) to complement ensemble methods, which can detect when minority should be considered the true answer instead of the majority. (2) Our experiments over several real-world classification datasets reveal promising results of our approach in the settings of SL and SSL by applying MTS methods in the ensemble final predictions and ensemble data augmentation steps respectively. Our proposed methods also outperform popular ensemble algorithms. (3) To pair with our experimental results, we also provide analytical evidences for the correctness of our proposed approaches. (4) Our approaches can be generically applied in ensemble methods to replace the majority voting rules.

The rest of the paper is organized as follows. Section 2 introduces some related works. Section 3 reviews preliminaries and BTS. Section 4 introduces our Machine Truth Serum approaches. Section 5 presents our experimental results. Section 6 concludes our paper.

2 Related work

2.1 Ensemble methods

Wisdom of the crowd (Surowiecki, 2005b) often performs better than a few elite individuals in the applications such as decision making of public policy (Morgan, 2014), answering the questions on general world knowledge (Singh et al., 2002). The same idea has been also

successfully applied in ML - ensemble methods combine multiple learning algorithms and usually performs better than any single method (Dietterich, 2000). Ensemble methods are usually used where aggregating the predictions are needed such as ensemble final predictions in supervised learning and ensemble data augmentations in SSL. In this paper, we focus on classification problem which is one of the most fundamental problems in ML community (Dai & Le, 2015; Yang et al., 2015; Howard & Ruder, 2018; Clark et al., 2018; Yang et al., 2019; Sachan et al., 2019; Yao et al., 2020).

2.1.1 Ensemble final predictions for supervised classification

In this part, we focus on describing the ensemble methods aggregating the final predictions for supervised classification which is the most commonly used scenario of ensemble methods. Ensemble methods consist of a rich family of algorithms. Popular ensemble methods include Boosting (e.g., AdaBoost (Freund & Schapire, 1997)), Bootstrap aggregating (e.g., Random Forest (Ho, 1995)), and Stacking (Bishop, 2006).

2.1.2 Ensemble data augmentation for semi-supervised classification

Another important application of ensemble methods is to generate better pseudo labels for unsupervised data with the help of data augmentation in semi-supervised classification other than improving the performance of final predictions. There are a wide family of SSL algorithms (Chapelle et al., 2006; Oliver et al., 2018; Berthelot et al., 2019; Xie et al., 2019; Chen et al., 2020). In this paper, we mainly review the recent pseudo labeling based SSL methods (Lee et al., 2013; Rasmus et al., 2015; Gong et al., 2017; Liu et al., 2019; Iscen et al., 2019; Berthelot et al., 2019). Pseudo labeling based SSL methods benefit from the unlabeled dataset by providing the high-quanlity explicit pseudo labels after applying data augmentation and ensemble methods. Some recent SSL methods such as UDA (Xie et al., 2019) conducted the consistency regularization training with implicit pseudo-labels and cannot be considered as our ensemble baseline because they don't use ensemble data augmentation methods to generate the pseudo labels. In this paper, we utilized MixMatch (Berthelot et al., 2019) and MixText (Chen et al., 2020) as the ensemble baseline methods in the SSL setting.

2.2 Bayesian truth serum

As mentioned in above sections, typical algorithms for aggregating human judgements and classical ensemble methods for combining classifiers' predictions have the same assumption that the majority answer is likely to be correct. Most works in these two settings, except for (Prelec, 2004), would fail when the majority answer is instead likely to be wrong. But BTS only works in the setting of aggregating human judgements by collecting subjective judgment data. Inspired by the ideas proposed by Prelec (2004); Prelec et al. (2017), we proposed two ML aided algorithms to discover the correct answer when it is minority instead of majority in the setting of classification problem. As our proposed methods are ML algorithms, they can be trained and the predictions will be made automatically instead of collecting subjective judgment data as the case in (Prelec, 2004).

3 Preliminary

In this paper, we consider supervised and semi-supervised classification problems. Nonetheless, for simplicity of demonstration, our main presentation focuses on binary classification. A multi-class extension of our method is presented in Section 4.3.

3.1 Supervised classification tasks

Suppose that we have a training dataset $\mathcal{D}_L := \{(x_i, y_i)\}_{i=1}^{N_L}$ and a test dataset $\mathcal{T} := \{(x_t, y_t)\}_{t=1}^{T}$, where $x_i \in X \subseteq \mathbb{R}^d$ is a d-dimensional feature vector and y_i is its true class label. We have K baseline classifiers $\mathcal{F} := \{f_1, f_2, ..., f_K : X \to \{0, 1\}\}$ that map each feature vector to a binary classification outcome. Ensemble method such as boosting algorithms can combine $\{f_1, f_2, ..., f_K\}$ to get better prediction results than each single one. For instance, Random Forest first applies the bootstrap aggregating to train multiple different decision trees to correct overfitting problems of decision trees. After training, the majority rule will be applied to generate the prediction result. We define the binary cross-entropy (BCE) loss of supervised classification as $\ell(f_k(x_i), y_i) := -[y_i \cdot \ln(f_k(x_i)) + (1 - y_i) \cdot \ln(1 - f_k(x_i))]$ for the k-th classifier on each data point (x_i, y_i) in the training dataset. Therefore, the empirical risk of the supervised classifier for $f_k, k = 1, ..., K$ using true labels is as follows:

$$L_1(f_k, D_L) = \frac{1}{N_L} \sum_{i=1}^{N_L} \ell(f_k(x_i), y_i).$$

The above dependence on the majority voting rule is ubiquitous in ensemble methods. The key assumption of using the majority rule is that the majority is more likely to be correct than random guessing. Denoting as $\mathsf{Maj}(\{f_1(x), f_2(x), ..., f_K(x)\})$ the majority answer from the K classifiers, formally, most, if not all, methods require that

$$P(\mathsf{Maj}(\{f_1(x), f_2(x), ..., f_K(x)\}) \neq y) < 0.5$$

Our goal is still to construct a single aggregator $\mathcal{A}_L(\{f_1, f_2, ..., f_K\})$ that takes the classifiers' predictions on each supervised data point as inputs and generates an accurate aggregated prediction. But we aim to provide instruction to cases where it is possible that

$$P(\mathsf{Maj}(\{f_1(x), f_2(x), ..., f_K(x)\}) \neq y) > 0.5$$

The challenge is to detect when the minority population has the true answer.

3.2 Semi-supervised classification tasks

In the semi-supervised classification tasks, there is also an unlabeled dataset $\mathcal{D}_U := \{(x_{N_L+j}, \cdot)\}_{j=1}^{N_U}$, where the labels are missing or unobservable. Let $N := N_L + N_U$. We unify the whole data including both labeled and unlabeled as $\mathcal{D} := \{(x_n, y_n)\}_{n=1}^{N}$. $\{y_n\}_{n=1}^{N_L}$ are the true labels of supervised dataset and $\{y_n\}_{n=N_L+1}^{N}$ are the pseudo labels of unsupervised dataset. Compared with supervised classification tasks, the information of unsupervised should be leveraged to improve the performance. Recent SSL methods usually apply the consistency regularization methods to make use of unsupervised data, where the output of original inputs and their data augmented ones should be consistent (Lee et al., 2013; Rasmus et al., 2015; Tarvainen & Valpola, 2017; Miyato et al., 2018; Iscen et al.,

2019; Berthelot et al., 2019; Sohn et al., 2020; Chen et al., 2020). In this paper, we consider MixMatch (Berthelot et al., 2019) and MixText (Chen et al., 2020) as our ensemble baseline methods because they generated the high-quanlity explicit pseudo labels for unsupervised data using ensemble methods.

For each unlabeled data $x_{N_L+j}, j = 1, ..., N_U$, the pseudo label can be generated by ensemble the model predictions of its data augmentations. We set the number of data augmentations for each unlabeled data to M. The data augmentation is denoted by $x_{N_L+j,m} := f_{\text{augment}}(x_{N_L+j}), m = 1, ..., M; j = 1, ..., N_U$. The pseudo label y_{N_L+j} can be generated based on M model predictions of data augmentations as $y_{N_L+j} = f_{\text{sharpen}}\left(\frac{1}{M}\sum_{m=1}^{M} \bar{f}_m(x_{N_L+j,m})\right), m = 1, ..., M; j = 1, ..., N_U$, where $\{\bar{f}_1, \bar{f}_2, ..., \bar{f}_M\}$ are extra M classifiers which are only utilized to generate better pseudo labels of unsupervised data and ensemble methods are limited to applying on this pseudo labeling process (not used in final classification prediction). We denoted the classifier conducting the final classification prediction as $f(\cdot)$. The function $f_{\text{sharpen}}(\cdot)$ can reduce the entropy of pseudo labels, e.g., setting to one-hot encoding based on the probabilities of different class labels (Sohn et al., 2020). The empirical risk of the semi-supervised classifier for $f(\cdot)$ using pseudo labels is as follows:

$$L_2(f, D_L, D_U) = \frac{1}{N_L}\sum_{i=1}^{N_L} \ell(f(x_i), y_i) + \frac{1}{N_U}\sum_{j=1}^{N_U} \ell(f(x_{N_L+j}), y_{N_L+j}).$$

Similar to 3.1, our goal is to construct a single aggregator $\mathcal{A}_S(\{\bar{f}_1, \bar{f}_2, ..., \bar{f}_M\})$ that takes the model predictions of data augmentations on each unsupervised data point as inputs and generates a high-quality pseudo label even the majority of model predictions is wrong. The challenge is still to detect when the minority population has the true answer.

3.3 Bayesian truth serum

Prelec (2004) considers the following human judgement elicitation problem: There are a set of agents denoted by $\{a_i\}_{i=1}^K$. The designer aims to collect subjective judgement from each agent about an unknown event $y \in \{0, 1\}$ and aggregate accordingly. Each of the agent i needs to report his own predicted label $l_i \in \{0, 1\}$ for y, and the percentage of other agents he believes will agree with him $p_i \in [0, 1]$. We will also call this second belief information as the *peer prediction information*. Denote the i's local belief of $l_j, j \neq i$ as $l^b_{i,j}, j \neq i$. p_i is defined as follows:

$$p_i = \mathbb{E}_{l^b_{i,j} j \neq i}\left[\frac{\sum_{j \neq i} \mathbb{1}(l^b_{i,j} = l_i)}{K - 1}\right]$$

In above the expectation is w.r.t. $l^b_{i,j}, j \neq i$ - this definition rigorously sets up the formulation, since in BTS, each agent only observes his/her private signals but not others.

We, as the designer, obtain the prediction labels $\{l_i\}_{i=1}^K$ and the percentage information $\{p_i\}_{i=1}^K$ from all the agents. The posterior for each label is defined as the actual percentage of this label which can be easily calculated utilizing the prediction results: (for label 1)

$$\text{Posterior}(1) = \frac{\sum_i \mathbb{1}(l_i = 1)}{K} \tag{1}$$

In (Prelec, 2004; Prelec et al., 2017), Prelec et al. promote the idea of using the average predicted percentage of the responding label as the approximation of the priors: (for label 1).

$$\text{Prior}(1) = \frac{\sum_{i=1}^{K} p_i^{\mathbb{1}(l_i=1)} \cdot (1-p_i)^{1-\mathbb{1}(l_i=1)}}{K} \tag{2}$$

If $\text{Posterior}(1) > \text{Prior}(1)$, label 1 will be taken as the surprisingly more popular answer, which should be considered as the true answer $\hat{y}$, even though it might be in minority's hands. The same rule is applied to label 0. Formally, if we denote $\hat{y}$ as the aggregated answer:

$$\hat{y} = \begin{cases} 1 & \text{if Prior}(1) < \text{ Posterior}(1); \\ 0 & \text{if Prior}(1) > \text{Posterior}(1). \end{cases} \tag{3}$$

The rest of the paper will focus on generalizing the above idea to aggregate classifiers' predictions.

4 Machine truth serum

In this section, we introduce Machine Truth Serum (MTS). We aim to build a more robust ensemble method which can recover the true answer (in minority's hands) if the majority's answer is wrong. Suppose we have access to a set of basic classifiers. We'd like to build a BTS-ish ensemble method to further improve the model's robustness. The challenge is to compute the priors from the classifiers - machine-trained classifiers do not encode beliefs as human agents do, so we cannot elicit the peer prediction information from them directly. We propose two machine learning aided approaches to perform the generation of this peer prediction information. We first introduce two MTS approaches for binary classification in supervised learning. Then we extend these approaches to multiclass classification case in supervised learning. After describing our proposed methods in supervised learning, we show the MTS methods for binary classification in SSL. Finally, the theoretical analysis of our MTS methods are provided.

4.1 Heuristic machine truth serum

We first introduce Heuristic Machine Truth Serum (HMTS). The high-level idea is to train a regression model for each classifier to predict the percent of the agreement from other classifiers on the prediction of each particular data point. After getting the predicted labels and the predicted peer prediction information of the classifiers, we can again approximate the priors using the predicted peer prediction information for each classifier, compute the average and compare it to posterior. In this part, HMTS for binary classification in supervised learning is introduced firstly and its multiclass extension is stated in Sect.4.3.

Algorithm 1 Heuristic Machine Truth Serum (Binary classification)

Require:

Input:

$\mathcal{D}_L = \{(x_1, y_1), ..., (x_{N_L}, y_{N_L})\}$: training data

$\mathcal{T} = \{(x_1, y_1), ..., (x_T, y_T)\}$: testing data

$\mathcal{F} = \{f_1, ..., f_K\}$: classifiers

Ensure:

Train K classifiers ($\mathcal{F}$) on the training data

For $i, j = 1, ..., K$, compute $\bar{y}_i^j$ according to Eqn.(4).

Train machine belief regressors $g_{j,0}, g_{j,1}$ on training dataset $\mathcal{D}_j^H := \{(x_i, \bar{y}_i^j)\}_{i=1}^{N_L}$.

for $t = 1$ to T **do**

  Get $\mathsf{Prior}(x_t, l = 1)$ and $\mathsf{Posterior}(x_t, l = 1)$ according to Eqn.(5) and Eqn.(7).

  if $\mathsf{Prior}(x_t, l = 1) < \mathsf{Posterior}(x_t, l = 1)$ **then**

    Output "surprising" answer 1 as the final prediction.

  else

    if $\mathsf{Prior}(x_t, l = 1) > \mathsf{Posterior}(x_t, l = 1)$ **then**

      Output "surprising" answer 0 as the final prediction.

    end if

  end if

end for

Given the training data $D = \{(x_i, y_i)\}_{i=1}^{N_L}$ and multiple classifiers $\{f_k\}_{k=1}^K$, we first try to compute the k-th classifier's "belief" of the fraction of other classifiers that would "agree" with it. Denote this number as $\bar{y}_n^k$ for each training example (x_i, y_i). $\bar{y}_i^k$ can be computed as follows:

$$\bar{y}_i^k = \frac{\sum_{c \neq k} \mathbb{1}\left(f_c(x_i) = f_k(x_i)\right)}{K - 1} \tag{4}$$

By above, we have pre-processed the training data to obtain $D_{H|k} := \{(x_i, \bar{y}_i^k)\}_{i=1}^{N_L}$, $k = 1, ..., K$, which can serve as the training data to predict the peer prediction information of classifier k (again to recall, peer prediction information is the fraction of other classifiers that classifier k believes would agree with it). We then train peer prediction regression models $\{\bar{p}_k\}_{k=1}^K$ on $D_{H|k} := \{(x_i, \bar{y}_i^k)\}_{i=1}^{N_L}$, $k = 1, ..., K$ respectively to map x_i to $\bar{y}_i^k$. We consider different class labels[1] and will first train two regression models: p_k^- and p_k^+ are two belief regression models of classifier k and trained on the examples whose predicted labels are 0s ($D_{H|k}^- := \{(x_i, \bar{y}_i^k) : f_k(x_i) = 0\}_{i=1}^{N_L}$) and 1s ($D_{H|k}^+ := \{(x_i, \bar{y}_i^k) : f_k(x_i) = 1\}_{i=1}^{N_L}$) respectively.

Then we compute the following prior of label 1 for each $(x_t, y_t) \in \mathcal{T}$ in the testing dataset:

$$\bar{p}_k(x_t) = \begin{cases} p_k^+(x_t) & \text{if } f_k(x_t) = 1; \\ 1 - p_k^-(x_t) & \text{if } f_k(x_t) = 0. \end{cases} \tag{5}$$

[1] In BTS, an agent predicts how many other agents agree with it depending on its own prediction.

After obtaining these peer prediction regression results $\bar{p}_k(x_t)$ for all test data points, the prior and posterior of $(x_t, y_t) \in \mathcal{T}$ in the test dataset are then calculated by,

$$\begin{aligned} P(x_t, 1) &:= \frac{\sum_k \bar{p}_k(x_t)}{K}; \\ Q(x_t, 1) &:= \frac{\sum_k \mathbb{1}(f_k(x_t) = 1)}{K}. \end{aligned} \tag{6}$$

If $P(x_t, 1) < Q(x_t, 1)$, the "surprising" answer 1 will be considered as the true answer. The decision rule is similar for label 0. The procedure is illustrated in Algorithm 1.

To be noted, training the regressors to estimate the prior instead of directly using Eq.(4) is necessary. Because, if we don't train the regressors and estimate the prior directly using Eq.(4), prior will always be equal to posterior and we cannot use the decision rule mentioned above to obtain the "surprising" answer by comparing prior and postrior. For simplicity, the proof for binary classification (multiclass case is similar) is given as follows:

We set $K_1 = \sum_k \mathbb{1}(f_k(x_t) = 1)$ and $K_2 = \sum_k \mathbb{1}(f_k(x_t) = 0)$. Obviously, $K = K_1 + K_2$. Then we can get:

$$\begin{aligned} \bar{y}_t^k(1) &= \frac{\sum_{c_1 \neq k} \mathbb{1}\left(f_{c_1}(x_t) = f_k(x_t) = 1\right)}{K - 1} \\ \bar{y}_t^k(0) &= \frac{\sum_{c_2 \neq k} \mathbb{1}\left(f_{c_2}(x_t) = f_k(x_t) = 0\right)}{K - 1} \end{aligned}$$

The above two quantities further help us compute both the posterior and the "direct prior" as follows:

$$\begin{aligned} P_{direct}(x_t, 1) &:= \frac{\sum_k \left[\bar{y}_t^k(1) \cdot \mathbb{1}(f_k(x_t) = 1) + (1 - \bar{y}_t^k(0)) \cdot \mathbb{1}(f_k(x_t) = 0)\right]}{K}; \\ &= \frac{\sum_k \left[\bar{y}_t^k(1) \cdot \mathbb{1}(f_k(x_t) = 1)\right] + \sum_k \left[(1 - \bar{y}_t^k(0)) \cdot \mathbb{1}(f_k(x_t) = 0)\right]}{K}; \\ &= \frac{K_1 \cdot \frac{K_1 - 1}{K - 1} + K_2 \cdot (1 - \frac{K_2 - 1}{K - 1})}{K} = \frac{K_1}{K} = \frac{\sum_k \mathbb{1}(f_k(x_t) = 1)}{K}; \end{aligned} \tag{7}$$

$$Q(x_t, 1) := \frac{\sum_k \mathbb{1}(f_k(x_t) = 1)}{K}. \tag{8}$$

Therefore, the prior is equal to the postrior by comparing Eqs.(7) and (8). Based on this proof, learning the regressors to estimate the prior instead of directly using Eq.(4) is necessary.

4.2 Discriminative machine truth serum

The Heuristic Machine Truth Serum above relies on training models to predict the peer prediction information for each classifier (which will be used to compute the priors) and compare them to the posteriors, and then decide on whether to follow the minority opinion or not. HMTS closely mimicked the procedure of BTS method in the seed paper. But it is not the most efficient way due to the extra computational cost of regressors. Also, its performance is dependent on the quality of regression models. We notice the above task of

determining whether to follow the minority or not is also a binary classification question. This observation inspires us to utilize a classification model to directly predict for each data point whether the minority should be chosen as the answer or not.

We propose Discriminative Machine Truth Serum (DMTS). Again, DMTS for binary classification will be introduced firstly and its multiclass extension is stated in Section 4.3. With DMTS, a new training dataset $D_D := \{x_i, \hat{y}_i\}_{i=1}^{N_L}$ about whether considering the minority as the final answer or not is constructed. Each data $D_D := (x_i, \hat{y}_i)$, for $i = 1, ..., N_L$, in this new training dataset is calculated as follows: for each $(x_i, y_i) \in D_L$

$$\hat{y}_i = \begin{cases} 1 & \text{if majority of } \mathcal{F} \text{ on } x_i \neq \text{ the true label;} \\ 0 & \text{if majority of } \mathcal{F} \text{ on } x_i = \text{ the true label.} \end{cases} \tag{9}$$

Now with above preparation, predicting whether majority is correct or not becomes a standard classification problem on $D_D := \{x_i, \hat{y}_i\}_{i=1}^{N_L}$. This is readily solvable by applying standard techniques. In our experiments, we will mainly use a Multi-Layer Perceptron (MLP) (Goodfellow et al., 2016) denoted as $\overline{f}$. $\overline{f}$ is trained on this new training dataset and can directly predict whether we should adopt the minority as the answer or not. $\overline{f}$ does not restrict to MLP and can be other classifiers. We have tried several other methods, such as logistic regression and support vector machine, with similar conclusions obtained. The whole procedure of DMTS is illustrated in Algorithm 2.

Algorithm 2 Discriminative Machine Truth Serum (Binary classification)

Require:
Input:
$\mathcal{D}_L = \{(x_1, y_1), ..., (x_{N_L}, y_{N_L})\}$: training data
$\mathcal{T} = \{(x_1, y_1), ..., (x_T, y_T)\}$: testing data
Ensure:
for $i = 1$ to N_L **do**
    Compute $\hat{y}_i$ according to Eqn.(7).
end for
Train DMTS classifier $\overline{f}$ on the dataset $\{x_i, \hat{y}_i\}_{i=1}^{N_L}$
for $t = 1$ to T **do**
    Compute the classification result $\overline{y}_t := \overline{f}(x_t)$
    if $\overline{y}_t = 0$ **then**
        Stay with the majority answer.
    else
        if $\overline{y}_t = 1$ **then**
            Predict with the minority answer.
        end if
    end if
end for

4.3 Multiclass extension of HMTS and DMTS

HMTS and DMTS can be extended to multiclass classification problem with the same ideas by modifying them accordingly. In the multiclass case, $l \in \mathcal{Y} = \{0, ..., L\}$ is denoted as the class label of the dataset. Consider HMTS first. For each classifier k, we need to consider

different class labels of regression models $\{p_k^l\}$, where $l \in \mathcal{Y} = \{0, ..., L\}$. p_k^l is the belief regression model of classifier k and trained on the examples whose predicting labels are ls.

Again compute the following prior for each x_i

$$p_k(x_i, l) = \begin{cases} p_k^l(x_i) & \text{if } f_k(x_i) = l; \\ \left(1 - p_k^v(x_i)\right) \cdot r_l & \text{if } f_k(x_i) = v \neq l. \end{cases} \tag{10}$$

where $r_l = p_k^l(x_i)/(\sum_{c \in \mathcal{Y}: c \neq v} p_k^c(x_i))$ is defined as the ratio of the l's belief to the summation of all the other classes' beliefs except for class v. In the multi-class classification tasks, we cannot directly obtain the prior of class l — $p_k^l(x_i)$ as in the binary classification by using $(1 - p_k^v(x_i))$ if $f_k(x_i) = v \neq l$. Therefore, the prior regressors for other classes $\{p_k^c(x_i) \mid c \in \mathcal{Y} : c \neq v\}$ need to be utilized to calculate the prior of class l with a normalization parameter r_l.

In HMTS, Eq.(7) modify to the following:

$$\begin{aligned} P(x_i, l) &:= \frac{\sum_{k=1}^K p_k(x_i, l)}{K}, \\ Q(x_i, l) &:= \frac{\sum_{k=1}^K \mathbb{1}(f_k(x_i) = l)}{K} \end{aligned} \tag{11}$$

We then compute all the priors and posteriors of each class label based on Eq.(11). It is possible that there exist more than one class labels whose posterior is larger than its prior. We define the set containing all these label classes as $\mathcal{Y}_{sat} = \{l \mid P(x_i, l) < Q(x_i, l)\}$. We then predict the class label which has the biggest improvement from its prior to posterior:

$$\text{argmax}_{l \in \mathcal{Y}_{sat}} \{Q(x_i, l) - P(x_i, l)\}$$

In DMTS, firstly we need to train a model that decides whether to apply the minority as the final answer which are very similar to the binary case. The difference is that we will then choose the minority answer as the predicted answer instead of using majority if i) it has the most votes in the minority answers and ii) the prediction result of classifier obtained in the training phase is 1 (we should use minority).

How does MTS work? In Fig. 1, we show four sample images to demonstrate how HMTS correct the wrong majority predictions. We show for these four cases even with high prediction on the wrong class, we are able to correct the prediction by introducing MTS to check on the priors. For example, in the first sample, the wrong prediction (number 8) is provided if we only look at posterior (number 0: 0.400; number 8: 0.467) following the majority rule. But the "surprising popular" correct minority (number 0) will be recovered if we predict based on Posterior - Prior (number 0: +0.117; number 8: +0.054).

4.4 HMTS and DMTS for semi-supervised classification

In this section, we describe the HMTS and DMTS for SSL classification problem. For simplicity, we consider binary classification and its multiclass extension can be inferred accordingly.

As we focus on applying ensemble methods on the pseudo labels' generation based on data augmentations for each unsupervised data, we first need to compute the m-th data augmentation classifier's "belief" of the fraction of other data augmentation classifiers that would "agree" with it. We first train M data augmentation classifiers $\{\bar{f}_m\}_{m=1}^M$ on the

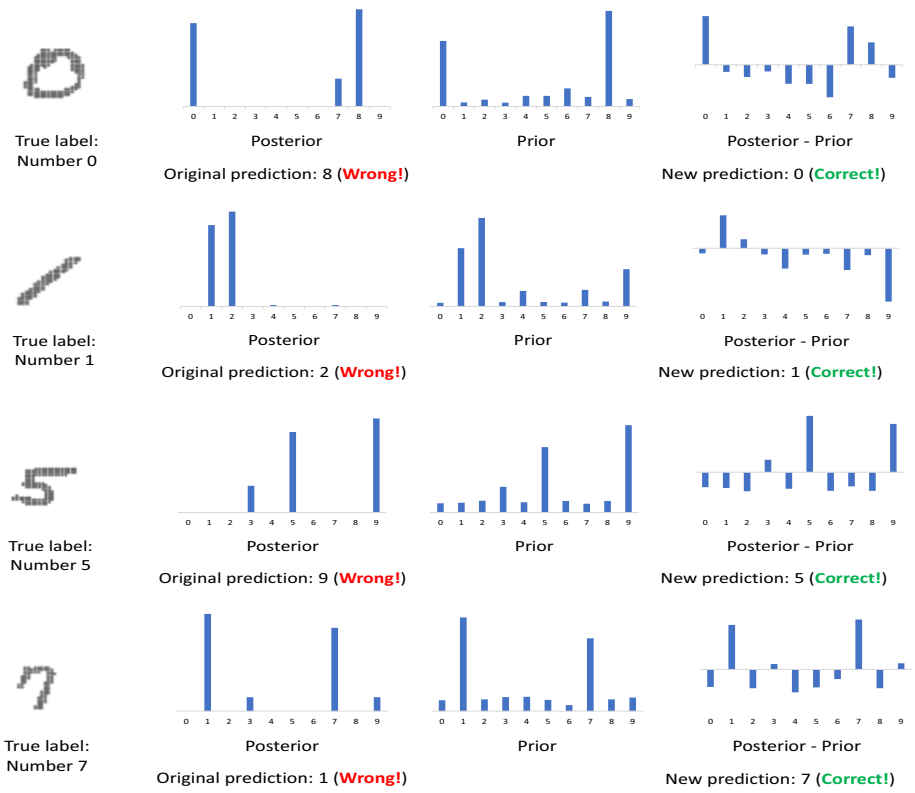

Fig. 1 Four sample images (number 0, 1, 5, and 7) where HMTS corrects the wrong majority predictions of the majority voting baseline on Pendigits dataset (10 classes) testing dataset. Their posterior, prior, and posterior-prior information are listed

supervised training dataset $D_L = \{(x_i, y_i)\}_{i=1}^{N_L}$. Then we can compute the classification predictions denoted as $\bar{f}_m(x_{i,m}), m = 1, ..., M; i = 1, ..., N_L$ for the data augmentations of supervised training dataset generated by $x_{i,m} := \text{Augmentation}(x_i), m = 1, ..., M; i = 1, ..., N_L$. Denote the m-th data augmentation classifier's "belief" (the fraction of other data augmentation classifiers that would "agree" with it) as $\hat{y}_i^m$ for the data augmentations of each supervised training example $(x_{i,m}, y_i)$. $\hat{y}_i^m$ can be computed as follows:

$$\hat{y}_i^m = \frac{\sum_{c \neq m} \mathbb{1}\left(f_c(x_{i,c}) = f_m(x_{i,m})\right)}{M - 1} \tag{12}$$

By above, we have pre-processed the supervised training data to obtain $D_{H|m}^L := \{(x_{i,m}, \hat{y}_i^m)\}_{i=1}^{N_L}$, $m = 1, ..., M$, which can serve as the training data to predict the peer prediction information of data augmentation classifier m. We then train peer prediction regression models $\{\hat{p}_m\}_{m=1}^M$ on $D_{H|m}^S := \{(x_{i,m}, \hat{y}_i^m)\}_{m=1}^M$, $m = 1, ..., M$ respectively to map $x_{i,m}$ to $\hat{y}_i^m$. We consider different class labels[2] and will first train two regression models: $\hat{p}_{k,m}^-$ and $\hat{p}_{k,m}^+$ are two belief regression models of data augmentation classifier m and trained on

[2] In BTS, an agent predicts how many other agents agree with it depending on its own prediction.

the examples whose predicted labels are 0s ($D^{-L}_{H|m} := \{(x_{i,m}, \hat{y}^m_i) : \hat{f}_m(x_{i,m}) = 0\}^{N_L}_{i=1}$) and 1s ($D^{+L}_{H|m} := \{(x_{i,m}, \hat{y}^m_i) : \hat{f}_m(x_{i,m}) = 1\}^{N_L}_{i=1}$) respectively.

Then compute the following prior of label 1 for the data augmentations of each x_{j+N_L} in the unsupervised dataset:

$$\hat{p}_m(x_{j+N_L,m}) = \begin{cases} \hat{p}^+_m(x_{j+N_L,m}) & \text{if } \hat{f}_m(x_{j+N_L,m}) = 1; \\ 1 - \hat{p}^-_m(x_{j+N_L,m}) & \text{if } \hat{f}_m(x_{j+N_L,m}) = 0. \end{cases} \tag{13}$$

After obtaining these peer prediction regression results $\hat{p}_m(x_{j+N_L,m}), j = 1, ..., N_U$ for all unsupervised data, the prior and posterior of $(x_{j+N_L,m}, y_{j+N_L,m}) \in D_U$ in the unsupervised dataset are then calculated by,

$$\begin{aligned} P(x_{j+N_L,m}, 1) &:= \frac{\sum_m \hat{p}_m(x_{j+N_L,m})}{M}; \\ Q(x_{j+N_L,m}, 1) &:= \frac{\sum_m \mathbb{1}(\hat{f}_m(x_{j+N_L,m}) = 1)}{M}. \end{aligned} \tag{14}$$

If $P(x_{j+N_U,m}, 1) < Q(x_{j+N_U,m}, 1)$, the "surprising" answer 1 will be considered as the true pseudo label in the semi-supervise classification. The decision rule is similar for answer 0.

As for the DMTS, $\{\hat{y}_i\}^{N_L}_{i=1}$ about whether considering the minority as the final pseudo label for each supervised training data x_i or not is constructed. Each data $D^L_D := (x_i, \hat{y}_i)$, for $i = 1, ..., N_L$, in this new training dataset is calculated as follows: for each $(x_i, y_i) \in D_L$

$$\hat{y}_i = \begin{cases} 1 & \text{if majority of predictions on } x_{i,m} (m = 1, ..., M) \neq \text{ the true label;} \\ 0 & \text{if majority of predictions on } x_{i,m} (m = 1, ..., M) = \text{ the true label.} \end{cases} \tag{15}$$

4.5 Theoretical analysis

We performed a formal analysis of the correctness of our proposed algorithms via proofs adapted from proofs for BTS (Prelec et al., 2017). Similar to BTS, with MTS, each classifier (i.e., an agent), depending on its own predicted label, will use a different regression model to predict how many other classifiers agree with it. For simplicity, we only present the theorems for binary classification. The proofs of multiclass are similar to the binary case. The details of proofs are reported in Appendix 1.

To set up for presenting the theorems, we restate our problem: we assume that each classifier $f_k(x)$ can take on any value in the discrete set $\{s_1, ..., s_S\}$ as its features for the simplicity of proof. In practice, conceptually each feature vector can be represented by an assigned (large-enough) categorical number. One can consider $s_i (i = 1, 2, , , S)$ as a code for each feature vector. Our proof builds on similar assumptions made in (Prelec et al., 2017):

Assumption 1 Conditional on each possible label l, $f_k(x), k = 1, 2, ..., K$ are independent from each other, and are identically distributed.

Assumption 2 The learner has access to the conditional distribution $\mathbb{P}(f_{-k}(x) \mid f_k(x))$, where $f_{-k}(x)$ denotes the prediction from a randomly selected classifier $j \neq k$.

We reproduce the following theorems:

Theorem 1 *The correct answer* (*majority or minority*) *cannot be deduced by any algorithm if only relying on posterior probabilities*, $Q(s_i, l), i = 1, ..., S; l = 0, 1$ *because considering either* 0 *or* 1 *as the correct label can generate the same posterior probabilities based on the training dataset.*

Theorem 1 implies that any existing ensemble algorithm based on the majority voting rule cannot always infer the true answer no matter either majority or minority is the final true answer. In other words, we cannot decide whether majority or minority is correct if we only know the information of the posterior probabilities Q over all the possible labels. The majority rule applied by the existing ensemble methods is a special case of Theorem 1.

Theorem 2 *For input* s_i, *the estimate of the prior prediction for the correct classification label denoted as* l^* *will be strictly underestimated if the prediction probability of the true label is less than* 1. *We can express this as*

$$P(s_i, l^*) < Q(s_i, l^*) \quad \text{if } \mathbb{P}\left(l^* \mid s_i\right) < 1.$$

We leave more details to the Appendix. Theorem 2 is applicable when the task is difficulty that the true label is only observed by a minority of the classifiers. A hidden assumption is that the minority but expert classifiers hold a stronger belief about the ground truth label than the majority classifiers who predicted wrongly. More formally we assume $\mathbb{P}(Y = l^* \mid f_k(s_i) = l^*) > \mathbb{P}(Y = l^* \mid f_k(s_i) = l)$ for all $l \neq l^*$. The high-level intuition is that the expert classifiers, though being minority, must retain a strong signal to classify a difficult task correctly. While for a non-expert one who predicted wrongly, would not "reason" specially how the hidden true label class. In Sect. 5.1, page 17-19, we have also provided an empirical observation and explanation.

Theorem 2 shows that having the prior information can help improve the robustness of models because the minority correct classification result can be recovered using the rule descried in the theorem when the minority is the true answer instead of the majority answer. In other words, having Theorem 2, the true minority answer can be revealed as correct if the prior probability is less than the posterior one. The existing ensemble methods always adopt the majority result as the final answer and cannot recover the minority correct answer.

As for the training complexity of our algorithm, the training time of HMTS is linear in the number of label classes because of the training of extra regressors. DMTS only needs to train one additional classifier and both the training and the running time are almost the same as the basic majority voting algorithm. Therefore our proposed methods are very practical to implement and run. Detailed discussions can be found in Appendix 2.

5 Experimental results

In this section, we present our experimental results. We test our proposed methods by applying in the ensemble final predictions step in supervised classification and in the ensemble data augmentations step in semi-supervised classification.

For supervised classification, we conducted the experiments on five binary and four multiclass real-world classification datasets. Experimental results show that consistently better classification accuracy can be obtained compared to always trusting the majority voting outcomes. As for the splitting of training and testing, the original setting are used when

training and testing files are provided. The remaining datasets only give one data file. We adopt 50/50 spliting for the testing results' statistical significance as more data is distributed to testing dataset.

As for semi-supervised classification, we adopt recent methods - MixMatch (Berthelot et al., 2019) and MixText (Chen et al., 2020) as our ensemble baselines. We also used UDA (Xie et al., 2019) as the baseline but it isn't the ensemble baseline because UDA doesn't use ensemble data augmentation methods to generate the pseudo labels. Both of the ensemble baselines (MixMatch and MixText) mixed labeled and unlabeled datasets utilizing MixUp (Zhang et al., 2018) by applying the recent data augmentations methods to generate low-entropy explicit pseudo labels for unlabeled examples. The difference is that Chen et al. (2020) applied MixUp in hidden space so that it is more suitable for text tasks. We used MixMatch to conduct the image classification experiments on CIFAR-10 and CIFAR-100 datasets (Krizhevsky et al., 2009). Both CIFAR-10 and CIFAR-100 consist of 50,000 training images and 10,000 testing images. The difference is that CIFAR-10 and CIFAR-100 have 10 and 100 classes respectively. MixText is used as the ensemble baseline model for text classification tasks, where Yahoo! Answers (Chang et al., 2008) and AG News (Zhang et al., 2015) datasets are performed. The experimental results show the effectiveness of our proposed methods by providing better pseudo labels for unsupervised data based on data augmentations than commly used ensemble method using the majority voting rule.

5.1 Experimental setup and results for supervised classification

In our binary classification experiments, we consider five commonly used binary classification algorithms which are Perceptron (Rosenblatt, 1958), Logistic Regression (LR) (Peng et al., 2002), Random Forest (RF), Support Vector Machine (SVM) (Chang & Lin, 2011), and MLP. In order to test the usefulness of our methods, we experiment with a noisy environment - we flipped the true class label with three noisy rates to construct three binary classifiers for each of the five methods which have mediocre performance on the test datasets. We wanted to diversify our classifiers by introducing different noisy rates (varying the data distribution). Our experiments used 0.06, 0.08, and 0.1 (probability of flipping the label) for each family of classifier. We also tried other values such as 0.1, 0.2, and 0.3, and we reached similar conclusions. In total, 15 different classifiers are obtained as the baseline classifiers.

In this subsection, we report the experimental results on five binary classification datasets and analyze when and why our proposed MTS methods perform better than majority voting.

Table 1 presents the experimental results of accuracy and the number of increased correct predictions for the three categories of cases, namely, Overall & "High disagreement (HA)" & "Low disagreement (LA)" cases', using methods of Uniformly-weighted Majority Voting, HMTS, and DMTS on five binary classification datasets. Specifically, "HA" cases are the tasks/instances when the ensemble is least certain about. "LA" ones are the relatively easier tasks/instances that the ensemble is more certain about, which is also the cases when the majority opinion is likely to be correct.

Because the accuracy improvement from using our proposed MTS mainly occurred for the HA cases, in Table 1, we report the accuracy of the majority voted baseline and our proposed methods (HMTS and DMTS) on HA cases, LA cases, and all cases separately.

Table 1 Accuracy and the number of increased correct predictions for the three categories of cases, namely, Overall & "High disagreement (HA)" & "Low disagreement (LA)" cases', using methods of Uniformly-weighted Majority Voting, HMTS, and DMTS on five binary classification datasets

Datasets	Breast cancer	Movie review	Spambase	Australian	German
Majority (ALL)	92.96% (264/284)	80.25%(321/400)	73.57%(1692/2300)	81.74%(282/345)	76.00%(380/500)
HMTS (ALL)	**95.42%** (264+7/284)	**82.00%** (321+7/400)	75.83% (1692+52/2300)	**84.06%** (282+8/345)	**77.20%** (380+6/500)
DMTS (ALL)	94.01% (264+4/284)	81.75% (321+6/400)	**76.30%** (1692+63/2300)	82.94% (282+4/345)	76.20% (380+1/500)
Majority (HA)	82.35% (42/51)	29.73% (11/37)	28.19% (42/149)	62.79% (27/43)	66.67% (30/45)
HMTS (HA)	**98.04%** (42+8/51)	**43.24%** (11+5/37)	60.40% (42+48/149)	**76.74%** (27+6/43)	**80.00%** (30+6/45)
DMTS (HA)	90.20% (42+4/51)	**43.24%** (11+5/37)	**75.17%** (42+70/149)	72.09% (27+4/43)	68.89% (30+1/45)
Majority (LA)	95.29% (222/233)	85.40% (310/363)	76.71% (1650/2151)	84.44% (255/302)	76.92% (350/455)
HMTS (LA)	94.85% (222-1/233)	**85.95%** (310+2/363)	**76.89%** (1650+4/2151)	**85.10%** (255+2/302)	**76.92%** (350+0/455)
DMTS (LA)	**95.29%** (222+0/233)	85.67% (310+1/363)	76.38% (1650-7/2151)	84.44% (255+0/302)	76.92% (350+0/455)

Best performance for each category is bolded

The "high disagreement" means that the difference between the number of predicting 0 and 1 is small. We have 15 classifiers and the instance will be considered as having "high disagreement" if the vote number of majority class is 8 or 9. For other conditions the instance will be considered as having "low disagreement". In MTS methods, the numbers of HA and LA instances we consider in five datasets are **51, 37, 149, 43, 45** and **233, 363, 2151, 302, 455** respectively. The numbers of five overall testing datasets are **284, 400, 2300, 345, 500**

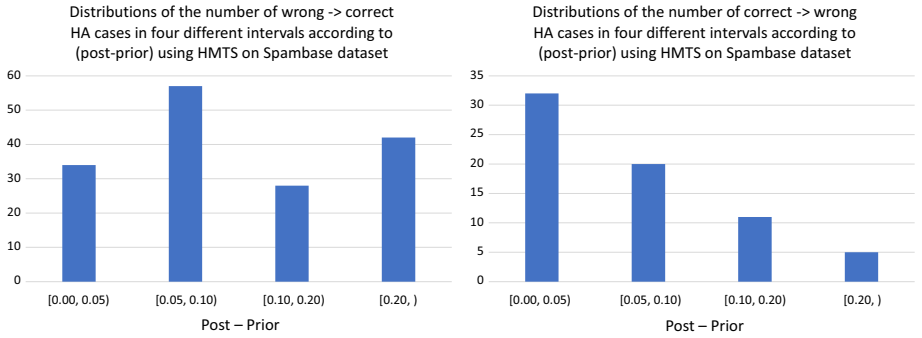

Fig. 2 Distributions of the number of wrong– >correct and correct– >wrong cases (two subsets of HA cases) in four different intervals according to the value of (posterior - prior) using HMTS on Spambase binary classification dataset, where a larger value means a bigger difference between prior and posterior probabilities. Our proposed MTS methods can obtain more correct answers when there is a significant difference between prior and posterior

From the results, we observe that our MTS methods significantly improve the performance on HA cases by 10--50%. It is reasonable because the "high disagreement" instances, compared with "low disagreement", are more difficult to classify. Hence, for HA cases, applying the majority voting rule leads to low accuracy and the majority answer is unreliable, when MTS is especially relevant because it was originally designed to address the issue of the majority being wrong. As such, our MTS methods can recover the correct minority answer when the majority is wrong, resulting in higher improvement in performance. For LA cases, that is, when the disagreement is low in the ensemble, accuracy achieved by trusting the majority labels is already high, as shown in the Majority (LA) row in Table 1. Such LA tasks leave us little room for our proposed methods to improve, as shown in the last three rows in Table 1 such that the accuracy is almost unchanged after applying our MTS methods as compared with the accuracy of the majority voting.

Another observation is that Heuristics Machine Truth Serum (HMTS) tends to have more robust and better performances than Discriminative Machine Truth Serum (DMTS) in most datasets, especially in the small-size datasets. These can be explained by the fact DMTS itself is a MLP classifier which needs a larger size of data to get good results. That HMTS can improve the classification accuracy in the small size of dataset is particularly useful in some fields such as healthcare in which collecting data is very time-consuming and expensive. As for the running time, DMTS is faster than HMTS as HMTS needs to compute the peer prediction results of all the 15 classifiers and DMTS only predicts once.

To further demonstrate the conditions under which MTS Methods are expected to be effective, we compare the distributions of the difference between prior and posterior probabilities in two subsets of HA cases from the Spambase dataset. The first subset consists of the cases where the correct classifications are successfully recovered by applying the MTS methods. The other subset is constituted by the cases where MTS ends up recovering wrong answers (i.e., the majority is correct in the first place, but rejected by MTS). In Fig. 2, for these two subsets of HA cases, we respectively present the distributions of the number of cases (wrong->correct and correct->wrong) in four different intervals according to the value of (posterior - prior), where a larger value means a bigger difference between prior and posterior probabilities. As Fig. 2 shows, the MTS methods obtain more correct answers when there is a significant difference between prior and posterior. In other words,

Table 2 Accuracy and the number of increased correct predictions for the three categories of cases, namely, Overall & "High disagreement (HA)" & "Low disagreement (LA)" cases', using methods of Uniformly-weighted Majority Voting, HMTS, and DMTS on four multi-class classification datasets

Datasets	Waveform	Statlog	Optical	Pen-Based
# of class	3	6	10	10
Majority (ALL)	85.04% (2126/2500)	86.70% (1734/2000)	97.50% (1752/1797)	95.08% (3326/3498)
HMTS (ALL)	85.60% (+14/2500)	**87.05%** (+7/2000)	**97.66%** (+3/1797)	95.48% (+14/3498)
DMTS (ALL)	**85.64%** (+15/2500)	86.75% (+1/2000)	97.61% (+2/1797)	**95.54%** (+16/3498)
Majority (HA)	42.59% (23/54)	23.08% (15/65)	40.00% (6/15)	57.32% (90/157)
HMTS (HA)	62.96% (23+11/54)	**53.33%** (15+8/65)	53.33% (6+2/15)	**68.15%** (90+17/157)
DMTS (HA)	**68.52%** (23+14/54)	24.62% (15+1/65)	**60.00%** (6+3/15)	66.88% (90+15/157)
Majority (LA)	85.98% (2103/2446)	88.84% (1719/1935)	97.98% (1746/1782)	96.86% (3236/3341)
HMTS (LA)	**86.10%** (2103+3/2446)	88.79% (1719-1/1935)	**98.04%** (1746+1/1782)	96.77% (3236-3/3341)
DMTS (LA)	86.02% (2103+1/2446)	**88.84%** (1719+0/1935)	97.92% (1746-1/1782)	**96.89%** (3236+1/3341)

Best performance for each category is bolded

We have 15 classifiers and the instance will be considered as having "high disagreement" if the vote number of majority class is less or equals to 6 for the 3-class dataset. The threshold number is 5 for 6-class and 3 for 10-class datasets. For other conditions the instance will be considered as having "low disagreement". In MTS methods, the numbers of HA and LA instances we consider in four datasets are **54, 65, 15, 157** and **2446, 1935, 1782, 3341** respectively. The numbers of four overall testing datasets are **2500, 2000, 1797, 3498**

we are more likely to recover the correct answer successfully if the difference between prior and posterior is large.

We also tested our extension to multi-class classification problems. Experimental results on four multi-class classification datasets are reported in Table 2. We observe that HMTS and DMTS obtained similarly good performance in the accuracy metric because the size of multi-class classification datasets is larger and the MLP of DMTS can perform better than the binary case. In Table 2, we also noted that the similar significant improvement on the HA cases and almost unchanged performance on the LA cases after applying our MTS methods for four multi-class classification datasets.

We also observe that, in both binary & multi-class classification tasks, DMTS performs much worse than HMTS for some datasets. We analyze this phenomenon below.

Analysis on why DMTS performs much worse than HMTS in some datasets In some datasets (e.g., German and Statlog), compared with HMTS, DMTS performs much worse. After examining the cases in those datasets, we observe that in the cases where HMTS recovers the correct minority answers, there is an imbalance in the distribution of labels. For example, most corrected cases in the Statlog dataset have the same label. It makes sense because HMTS is a heuristic method and can compute for each data point individually and doesn't have the constraints of balanced distribution on labels. For DMTS, however, we found that the labels of the cases using DMTS are balanced, which suggests that it seems to be subject to a constraint of label balance. This could be because we trained the model on the dataset with a balanced distribution of labels. As a result, it enforces the balanced distribution of the labels when applied in the testing datasets.

Finally, we compare between several popular ensemble algorithms and our proposed approaches. We list the testing accuracy for Adaboost with 15 decision tree base estimators,

Table 3 Comparison between popular ensemble and our proposed approaches

Methods	Adaboost	Random Forest	Weighted Majority	Stacking	HMTS	DMTS
Breast Cancer	94.37%	94.37%	94.01%	94.72%	**96.13%**	94.01%
Movie Review	75.10%	77.20%	**81.60%**	70.30%	80.85%	80.60%
Spambase	74.74%	74.65%	74.17%	75.91%	76.87%	**77.35%**
Australian	82.03%	84.06%	84.06%	**85.22%**	83.44%	82.94%
German	72.20%	74.80%	73.80%	**77.20%**	**77.20%**	76.20%
Waveform	81.80%	82.60%	85.36%	84.00%	85.48%	**85.60%**
Statlog	85.85%	86.15%	86.85%	82.70%	**87.10%**	86.75%
Optical	93.99%	94.88%	92.21%	95.83%	97.61%	**97.66%**
Pen-Based	94.97%	95.45%	90.59%	95.43%	**95.57%**	95.51%
# of best	0	0	1	1	**4**	**3**
# of significant wins	0	1	0	0	**3**	**3**

Best performance for each dataset is bolded

# of best means the number of datasets where the benchmark achieves the best performance. # of significant wins means winning number of comparisons between itself and other methods if they are significantly different (p-value<0.05) by doing paired t-test

Random Forest with 15 decision trees, Weighted Majority (Germain et al., 2015), Stacking with the same setting of 15 classifiers utilized in our two MTS algorithms and Logistic Regression or SVM as meta classifier, HMTS, and DMTS for all nine datasets in Table 3. As shown in the table, HMTS and DMTS outperform Adaboost, Random Forest, Weighted Majority, and Stacking in seven datasets and are very close to the best in two datasets. Compared to other weighted methods, we'd like to note that our aggregation operates on each single task separately - this means that our method will be more robust when the difficulty levels of tasks differ drastically in the dataset. None of the other weighted methods (with fixed and learned weights) has this feature. We also find that our method is robust to a smaller number of classifiers, in contrast to, say Adaboosting. We also conduct paired t-testing where all methods are compared to each other. If two methods are significantly different (p-value<0.05) and one method performs better, it means significant win or better. Random Forest is significantly better than Adaboost. HMTS and DMTS are significantly better than Adaboost, Random Forest, and Weighte Majority (almost for Stacking). Paired t-testing results show the effectiveness of our proposed approaches.

5.2 Experimental setup and results for semi-supervised classification

We adopt the recent SSL methods UDA (Xie et al., 2019), MixMatch (Berthelot et al., 2019), and MixText (Chen et al., 2020) as our baselines in SSL. Becuase UDA doesn't use ensemble data augmentation methods to generate the pseudo labels, we consider MixMatch and MixText as the ensemble baselines.

We applied UDA and MixMatch on image classification tasks (CIFAR-10 and CIFAR-100). In both CIFAR-10 and CIFAR-100 datasets, 14,000 data points are utilized as supervised dataset and the remaining as unsupervised dataset. For UDA, it performs worse than other methods because it doesn't use ensemble data augmentation methods to generate the pseudo labels. For MixMatch, we tried different data augmentation settings, where varying

Table 4 Classification accuracy (%) in UDA, MixMatch (2-AUG), MixMatch (5-AUG), HMTS, and DMTS settings on the CIFAR-10 and CIFAR-100 testing dataset using MixMatch method. 2-AUG means that two data augmentation samples are constructed for each unsupervised data. HMTS and DMTS are based on 5-AUG setting

Methods	CIFAR-10 (%)	CIFAR-100 (%)
UDA	88.70	75.23
MixMatch (2-AUG)	90.68	76.78
MixMatch (5-AUG)	91.59	78.20
HMTS	**92.62**	**80.75**
DMTS	91.90	79.52

Best performance for each dataset is bolded

Table 5 Classification accuracy (%) in UDA, MixText (2-AUG), MixText (3-AUG), HMTS, and DMTS settings on the Yahoo! Answers and AG News testing dataset using MixText method. 2-AUG means that two data augmentation samples are constructed for each unsupervised data. HMTS and DMTS are based on 3-AUG setting

Methods	Yahoo! Answers (%)	AG News (%)
UDA	65.6	86.8
MixText (2-AUG)	66.7	87.6
MixText (3-AUG)	67.1	88.3
HMTS	**67.8**	**89.5**
DMTS	67.3	88.9

Best performance for each dataset is bolded

number of data augmented samples are constructed for each unsupervised data. As shown in Table 4, 2-AUG and 5-AUG settings are conducted. We observe that constructing more data augmented samples can improve the classification accuracy. HMTS and DMTS in Table 4 are applied on 5-AUG settings. We change the pseudo labels on the "high disagreement" cases, which are the ones when the ensemble is least certain about. In the image classification tasks, instances are considered as having "high disagreement" if three give the same classification results and the remaining two provide another consistent prediction results. HMTS and DMTS further improve the better performance than 5-AUG ensemble setting.

MixText utilized Mixup in the hidden states so that it is more suitable for text tasks. UDA can also be used in the text tasks. Therefore, we conducted the experiments on two text classification datasets - Yahoo! Answers and AG News using UDA and MixText. 100 labeled data and 5,000 unlabeled data per class in both datasets are used to train the model. For UDA, similar to image classification tasks, it performs worse than other methods because it doesn't use ensemble data augmentation methods to generate the pseudo labels. For MixText, we also tried different data augmentation settings as in the MixMatch, where varying number of data augmented samples are constructed for each unsupervised data. In the 2-AUG setting, Russian and German machine translation models are utilized to generate data augmented samples for each unsupervised data. We add one more model - French machine translation model in the 3-AUG setting. We change the pseudo labels on the "high disagreement" cases which is defined in the above paragraph. In the text classification tasks, instances are considered as having "high disagreement" if two give the same

Table 6 Pseudo labels accuracy (%) in high disagreement (HA) cases for 2-AUG, 5-AUG, HMTS, and DMTS settings on CIFAR-10 dataset

Methods	HA accuracy (pseudo labels) in CIFAR-10	Improvement over 2-AUG (%)
2-AUG	86.34% (2308/2673)	-
5-AUG	89.45% (2391/2673)	+3.11%
HMTS	92.07% (2461/2673)	+5.73%
DMTS	90.35% (2415/2673)	+4.01%

2-AUG means that two data augmentation samples are constructed for each unsupervised data. HMTS and DMTS are based on the 5-AUG setting. The numbers of HA unsupervised cases and overall unsupervised cases are **2,673** and **36,000** respectively

classification results and the remaining one provide another prediction result. In Table 5, we observe the consistent improvement as the one in Table 4.

The reason that our MTS methods work in SSL is that better pseudo labels for unsupervised data are obtained. For better analyzing why our MTS methods are effective, we show that the accuracy improvement on the high disagreement (HA) cases' pseudo labels for unsupervised data since we only applied our MTS methods on HA cases. The number of HA cases in the 36,000 unsupervised cases in CIFAR-10 dataset is 2,673. Because we actually have true labels of unsupervised data in CIFAR-10, we can calculate the accuracy on HA cases' pseudo labels obtained by aggregating the predictions of data augmented cases for unsupervised data with ensemble methods. As shown in Table 6, our methods provide more correct pseduo labels and the improvement is significant. The similar improvements are observed on the experiments for other datasets (CIFAR-100, Yahoo! Answers, and AG News) in the SSL setting and the details are shown in Appendix 3.

6 Conclusion

In this paper, we proposed two ML aided methods HMTS and DMTS to detect when the minority should be the true answer instead of majority. Our experiments over a set of classification datasets show that better classification performance can be obtained by applying our MTS methods in the ensemble final prediction step in supervised classification and in the ensemble data augmentations step in SSL by generating better pseudo labels for unsupervised data. Our proposed methods also outperform popular ensemble algorithms and can be generically applied as a subroutine in ensemble methods to replace majority voting. For future work, we will apply our MTS methods on more real-world datasets.

Appendix 1: Proof of theorems in sect.4.4.

In this part, we provided the detailed proof of two theorems which are the analytical evidences for the correctness of our proposed approaches. For simplicity, we only show the proof details of binary classification. The proof of multi-class classification is similar to the

binary case. This proof is largely adapted from (Prelec et al., 2017). Nonetheless we reproduce the details for completeness.

Theorem 1 *The correct answer* (*majority or minority*) *cannot be deduced by any algorithm if only relying on posterior probabilities*, $Q(s_i, l), i = 1, ..., S; l = 0, 1$ *because considering either* 0 *or* 1 *as the correct label can generate the same posterior probabilities based on the training dataset*.

Proof In this proof, for any arbitrarily selected class label as the answer, we can generate the same posterior probabilities. Therefore, we cannot decide which label (majority or minority) is the true class label if only relying on posterior probabilities.

Denote by l^* as the true class label. Given the training dataset, $\mathbb{P}(s_i \mid l^*), i = 1, ..., S$ is known. Based on the description of theorem, the posterior probabilities $Q(s_i, l) = \mathbb{P}(l \mid s_i), i = 1, ..., S; l = 0, 1$ is also known.

But we don't know which class label is the truth label. We arbitrarily selected one class label l as the true label. We denote the corresponding model is $K(s_i, l)$. We will prove that $K(s_i, l)$ can generate the same $\mathbb{P}(s_i \mid l^*), i = 1, ..., S$ and $Q(s_i, l) = \mathbb{P}(l \mid s_i), i = 1, ..., S; l = 0, 1$ for any arbitrarily selected class label l.

Because the known parts don't constrain the prior over the feature vector s_i. In particular, we can set the prior of model $K(s_i, l)$ to:

$$\mathbb{K}(s_i) = \frac{\mathbb{P}(s_i \mid l^*)}{\mathbb{P}(l \mid s_i)} \left(\sum_r \frac{\mathbb{P}(s_r \mid l^*)}{\mathbb{P}(l \mid s_r)} \right)^{-1}$$

Because the posteriors in the corresponding model $K(s_i, l)$ must equal to the known posteriors, we have $\mathbb{K}(l \mid s_i) = \mathbb{P}(l \mid s_i)$, for $i = 1, ..., S; l = 0, 1$. So we can get the joint distribution of label l and the feature vector s_i:

$$\begin{aligned} \mathbb{K}(l, s_i) &= \mathbb{K}(l \mid s_i)\mathbb{K}(s_i) = \mathbb{P}(l \mid s_i)\mathbb{K}(s_i) \\ &= \mathbb{P}(s_i \mid l^*) \left(\sum_r \frac{\mathbb{P}(s_r \mid l^*)}{\mathbb{P}(l \mid s_r)} \right)^{-1} \end{aligned}$$

Then we can get the marginal distribution l by summing over i:

$$\mathbb{K}(l) = \sum_i \mathbb{P}(s_i \mid l^*) \left(\sum_r \frac{\mathbb{P}(s_r \mid l^*)}{\mathbb{P}(l \mid s_r)} \right)^{-1} = \left(\sum_r \frac{\mathbb{P}(s_r \mid l^*)}{\mathbb{P}(l \mid s_r)} \right)^{-1}$$

After getting the marginal distributions $\mathbb{K}(s_i), \mathbb{K}(l)$, and the posteriors, $\mathbb{K}(l \mid s_i)$, for $i = 1, ..., S$, the feature vector distribution s_i of the arbitrarily selected class label l, $\mathbb{K}(s_i \mid l)$ can be calculated by:

$$\mathbb{K}(s_i \mid l) = \frac{\mathbb{K}(l \mid s_i)\mathbb{K}(s_i)}{\mathbb{K}(l)} = \mathbb{P}(s_i \mid l^*)$$

Because l was arbitrarily chosen, this theorem is proved. □

Theorem 1 implies that any existing ensemble algorithm based on the majority voting rule cannot always infer the true answer no matter either majority or minority is the final true answer. In other words, we cannot decide whether majority or minority is correct if we

only know the information of the posterior probabilities Q over all the possible labels. The majority rule applied by the existing ensemble methods is a special case of Theorem 1.

In the following part, we are considering the extra information which is the estimation of other classifiers' prediction results. We use $\mathbb{P}(v_l \mid s_i), l \in \{0, 1\}$ to represent the how many percentage of basic classifiers will predict label l given s_i.

We also define two possible learnt final classification functions ω_0^i and ω_1^i which decide the final label for each s_i. ω_0^i is the function which finally predict s_i as 0 and ω_1^i is the function which finally predict s_i as 1. If the true label is 1, ω_1^i is defined as the actual final classifier and ω_0^i is counterfactural final classifier. For simplicity, we ignore the input index of $\omega_l^i, l \in \{0, 1\}$ for each s_i and write it as $\omega_l, l \in \{0, 1\}$ in the proof of Theorem 2.

Theorem 2 *For input s_i, the estimate of the prior prediction for the correct classification label denoted as l^* will be strictly underestimated if the prediction probability of the true label is less than* 1. *We can express this as*

$$P(s_i, l^*) < Q(s_i, l^*) \qquad \text{if } \mathbb{P}(l^* \mid s_i) < 1.$$

Proof For each s_i, we set l^* as the true label. We first prove that the actual percentage of predicted labels for the true label in the actual final classifier exceeds counterfactual classifier's percentage for the true label, $\mathbb{P}(v_{l^*} \mid w_{l^*}) > \mathbb{P}(v_{l^*} \mid w_l), l \neq l^*$.

Based on the description of ω_l and v_l mentioned above and a BTS's hidden assumption that the minority but expert classifiers hold a stronger belief about the ground truth label than the majority classifiers who predicted wrongly, for the true label l_*, the probability of ω_{l_*} being the actual final classifier for the expert classifiers predicting correctly is higher than the one for the non-expert classifiers predicting the other wrong label l. Therefore, we can get $\mathbb{P}(w_{l^*} \mid v_{l^*}) > \mathbb{P}(w_{l^*} \mid v_l)$. Then we have $\mathbb{P}(w_{l^*} \mid v_{l^*})\mathbb{P}(v_l) > \mathbb{P}(w_{l^*} \mid v_l)\mathbb{P}(v_l)$ by timing the same factor $P(v_l)$ on both sides. So we have:

$$\mathbb{P}(w_{l^*} \mid v_{l^*}) > \mathbb{P}(w_{l^*} \mid v_{l^*})\mathbb{P}(v_{l^*}) + \mathbb{P}(w_{l^*} \mid v_l)\mathbb{P}(v_l) = \mathbb{P}(w_{l^*}) \tag{A1}$$

According to Bayesian rule, we have the following deduction:

$$\frac{\mathbb{P}(v_{l^*} \mid w_{l^*})}{\mathbb{P}(v_{l^*} \mid w_l)} = \frac{\mathbb{P}(w_{l^*} \mid v_{l^*})\mathbb{P}(w_l)}{\mathbb{P}(w_l \mid v_{l^*})\mathbb{P}(w_{l^*})} = \frac{\mathbb{P}(w_{l^*} \mid v_{l^*})}{1 - \mathbb{P}(w_{l^*} \mid v_{l^*})} \frac{1 - \mathbb{P}(w_{l^*})}{\mathbb{P}(w_{l^*})} \tag{A2}$$

Based on (A1), (A2) is greater than one. So $\mathbb{P}(v_{l^*} \mid w_{l^*}) > \mathbb{P}(v_{l^*} \mid w_l), l \neq l^*$ is proved.

The estimate of classification prediction given the feature value s_i can be computed by marginalizing the actual and counterfactual final classifiers, $\mathbb{P}(v_{l^*} \mid s_i) = \mathbb{P}(v_{l^*} \mid w_{l^*})\mathbb{P}(w_{l^*} \mid s_i) + \mathbb{P}(v_{l^*} \mid w_l)\mathbb{P}(w_l \mid s_i)$. And we proved that $\mathbb{P}(v_{l^*} \mid w_{l^*}) > \mathbb{P}(v_{l^*} \mid w_l), l \neq l^*$. Therefore, $\mathbb{P}(v_{l^*} \mid s_i) \leq \mathbb{P}(v_{l^*} \mid w_{l^*})$. It will be the strict inequality unless $\mathbb{P}(w_{l^*} \mid s_i) = 1$. If the prediction probability is less than 1, the prior prediction for each s_i will be strictly underestimated. So we can get $P(s_i, l^*) < Q(s_i, l^*)$ if the prediction probability is less than 1. This theorem is proved. □

Theorem 2 shows that having the prior information can help improve the robustness of models because the minority correct classification result can be recovered using the rule descried in the theorem when the minority is the true answer instead of the majority answer. In other words, having Theorem 2, the true minority answer can be revealed as correct if the prior probability is less than the posterior one. The existing ensemble methods always adopt the majority result as the final answer and cannot recover the minority correct answer.

Appendix 2: Complexity analysis of HMTS and DMTS

For HMTS, for example in our experiments, another $K \cdot (L+1)$ (label classes $\{0, 1, ..., L\}$) simple regressors will be trained to predict others' beliefs based on K baseline classifiers. So the total training time is linear in the number of label classes.

After training the extra regressors, running the algorithm only requires taking $L+1$ averages (K of the $K \cdot (L+1)$ regressors each) and compare with average posterior. DMTS will only need to train one additional classifier based on K classifiers and both the training and the running time are almost the same as the basic majority voting algorithm. The above complexity analysis shows our methods are very practical.

Appendix 3: Pseudo labels accuracy (%) in high disagreement (HA) cases for other datasets

In this section, we show accuracy improvement on the high disagreement (HA) cases' pseudo labels for unsupervised data for CIFAR-100, Yahoo! Answers, and AG News datasets. As shown in Tables 7, 8, and 9 , we observe the consistent improvements after applying our proposed MTS methods.

Table 7 Pseudo labels accuracy (%) in high disagreement (HA) cases for 2-AUG, 5-AUG, HMTS, and DMTS settings on CIFAR-100 dataset

Methods	HA accuracy (pseudo labels) in CIFAR-100	Improvement over 2-AUG (%)
2-AUG	76.18% (2559/3359)	–
5-AUG	78.38% (2633/3359)	+2.20%
HMTS	81.30% (2731/3359)	+5.12%
DMTS	80.53% (2705/3359)	+4.35%

2-AUG means that two data augmentation samples are constructed for each unsupervised data. HMTS and DMTS are based on the 5-AUG setting. The numbers of HA unsupervised cases and overall unsupervised cases are **3,359** and **36,000** respectively

Table 8 Pseudo labels accuracy (%) in high disagreement (HA) cases for 2-AUG, 3-AUG, HMTS, and DMTS settings on Yahoo! Answers dataset

Methods	HA accuracy (pseudo labels) in Yahoo! Answers	Improvement over 2-AUG (%)
2-AUG	64.1% (8452/13186)	–
3-AUG	66.2% (8729/13186)	+2.1%
HMTS	67.4% (8887/13186)	+3.3%
DMTS	66.9% (8821/13186)	+2.8%

2-AUG means that two data augmentation samples are constructed for each unsupervised data. HMTS and DMTS are based on the 3-AUG setting. The numbers of HA unsupervised cases and overall unsupervised cases are **13,186** and **50,000** respectively

Table 9 Pseudo labels accuracy (%) in high disagreement (HA) cases for 2-AUG, 3-AUG, HMTS, and DMTS settings on AG News dataset

Methods	HA accuracy (pseudo labels) in AG News	Improvement over 2-AUG (%)
2-AUG	86.4% (3558/4118)	–
3-AUG	88.1% (3628/4118)	+1.7%
HMTS	90.7% (3735/4118)	+4.3%
DMTS	90.2% (3714/4118)	+3.8%

2-AUG means that two data augmentation samples are constructed for each unsupervised data. HMTS and DMTS are based on the 3-AUG setting. The numbers of HA unsupervised cases and overall unsupervised cases are **4,118** and **20,000** respectively

Author Contribution Yang Liu proposed applying the idea of Bayesian Truth Serum on aggregating human subjective judgements in the ensemble methods in machine learning. Tianyi Luo and Yang Liu designed the method details and wrote the paper. Tianyi Luo designed, implemented, and tuned the experiments.

Funding This work is partially supported by the National Science Foundation (NSF) under grant IIS-2007951 and UC Santa Cruz.

Data availability All data and materials support their published claims and comply with field standards.

Code availability All software application or custom code support their published claims and comply with field standards. The code will be publicly available.

Declarations

Conflict of interest The author declare that they have no conflict of interest.

Ethics approval Not Applicable.

Consent to participate Not Applicable.

Consent for publication Not Applicable.

References

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C.A. (2019). Mixmatch: A holistic approach to semi-supervised learning. Advances in neural information processing systems (Vol. 32). Curran Associates, Inc.

Bishop, C.M. (2006). Pattern recognition and machine learning. springer, New York

Chang, C.-C., & Lin, C.-J. (2011). Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST), 2*(3), 27.

Chang, M.-W., Ratinov, L.-A., Roth, D., & Srikumar, V. (2008). Importance of semantic representation: Dataless classification. *Aaai, 2*, 830–835.

Chapelle, O., Scholkopf, B., & Zien, A. (2006). *Semi-Supervised Learning*. US: MIT Press.

Chen, J., Yang, Z., Yang, D. (2020). Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. arXiv preprint arXiv:2004.12239.

Chen, K.-Y., Fine, L. R., & Huberman, B. A. (2004). Eliminating public knowledge biases in information-aggregation mechanisms. *Management Science, 50*(7), 983–994.

Clark, K., Luong, M.-T., Manning, C.D., Le, Q.V. (2018). Semi-supervised sequence modeling with cross-view training. arXiv preprint arXiv:1809.08370.

Dai, A. M., & Le, Q. V. (2015). Semi-supervised sequence learning. *Advances in Neural Information Processing Systems, 28*, 3079–3087.
Dietterich, T. G. (2000). *Ensemble methods in machine learning* (pp. 1–15). Springer, Berlin: International workshop on multiple classifier systems.
Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Sciences, 55*(1), 119–139.
Germain, P., Lacasse, A., Laviolette, F., Marchand, M., & Roy, J.-F. (2015). Risk bounds for the majority vote: From a pac-bayesian analysis to a learning algorithm. *The Journal of Machine Learning Research, 16*(1), 787–860.
Gong, C., Tao, D., Chang, X., & Yang, J. (2017). Ensemble teaching for hybrid label propagation. *IEEE Transactions on Cybernetics, 49*(2), 388–402.
Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. US: MIT press.
Ho, T.K. (1995). Random decision forests. Proceedings of 3rd international conference on document analysis and recognition (Vol. 1, pp. 278-282).
Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146.
Iscen, A., Tolias, G., Avrithis, Y., & Chum, O. (2019). Label propagation for deep semisupervised learning. Proceedings of the ieee/cvf conference on computer vision and pattern recognition (pp. 5070-5079).
Krizhevsky, A., & Hinton, G. (2009). *Learning multiple layers of features from tiny images* (Technical report). University of Toronto.
Lee, D.-H., et al. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. *Workshop on Challenges in Representation Learning ICML, 3*, 896.
Liu, Q., Peng, J., & Ihler, A. T. (2012). Variational inference for crowdsourcing. *Advances in Neural Information Processing Systems, 25*, 692–700.
Liu, X., Van De Weijer, J., & Bagdanov, A. D. (2019). Exploiting unlabeled data in cnns by self-supervised learning to rank. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 41*(8), 1862–1878.
Miyato, T., Maeda, S.-I., Koyama, M., & Ishii, S. (2018). Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 41*(8), 1979–1993.
Morgan, M. G. (2014). Use (and abuse) of expert elicitation in support of decision making for public policy. *Proceedings of the National Academy of Sciences, 111*(20), 7176–7184.
Oliver, A., Odena, A., Raffel, C., Cubuk, E.D., Goodfellow, I.J. (2018). Realistic evaluation of deep semi-supervised learning algorithms. arXiv preprint arXiv:1804.09170.
Peng, C.-Y.J., Lee, K. L., & Ingersoll, G. M. (2002). An introduction to logistic regression analysis and reporting. *The Journal of Educational Research, 96*(1), 3–14.
Prelec, D. (2004). A bayesian truth serum for subjective data. *Science, 306*(5695), 462–466.
Prelec, D., Seung, H. S., & McCoy, J. (2017). A solution to the single-question crowd wisdom problem. *Nature, 541*(7638), 532.
Rasmus, A., Valpola, H., Honkala, M., Berglund, M., Raiko, T. (2015). Semisupervised learning with ladder networks. arXiv preprint arXiv:1507.02672.
Raykar, V. C., Yu, S., Zhao, L. H., Valadez, G. H., Florin, C., Bogoni, L., & Moy, L. (2010). Learning from crowds. *Journal of Machine Learning Research, 11*, 1297–1322.
Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review, 65*(6), 386.
Sachan, D.S., Zaheer, M., Salakhutdinov, R. (2019). Revisiting lstm networks for semi-supervised text classification via mixed objective function. Proceedings of the aaai conference on artificial intelligence (Vol. 33, pp. 6940-6948).
Simmons, J. P., Nelson, L. D., Galak, J., & Frederick, S. (2010). Intuitive biases in choice versus estimation: Implications for the wisdom of crowds. *Journal of Consumer Research, 38*(1), 1–15.
Singh, P., Lin, T., Mueller, E.T., Lim, G., Perkins, T., Zhu, W.L. (2002). Open mind common sense: Knowledge acquisition from the general public. Otm confederated international conferences "on the move to meaningful internet systems" (pp. 1223-1237).
Sohn, K., Berthelot, D., Li, C.-L., Zhang, Z., Carlini, N., Cubuk, E.D., ... Raffel, C. (2020). Fixmatch: Simplifying semi-supervised learning with consistency and confidence. arXiv preprint arXiv:2001.07685.
Surowiecki, J. (2005). *The Wisdom of Crowds*. New York: Anchor.
Surowiecki, J. (2005b). The wisdom of crowds. Anchor.
Tarvainen, A., & Valpola, H. (2017). Mean teachers are better role models: Weightaveraged consistency targets improve semi-supervised deep learning results. arXiv preprint arXiv:1703.01780.

Xie, Q., Dai, Z., Hovy, E., Luong, M.-T., Le, Q.V. (2019). Unsupervised data augmentation for consistency training. arXiv preprint arXiv:1904.12848.

Yang, D., Chen, J., Yang, Z., Jurafsky, D., Hovy, E. (2019). Let's make your request more persuasive: Modeling persuasive strategies via semi-supervised neural nets on crowdfunding platforms. Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers) (pp. 3620-3630).

Yang, D.,Wen, M., Rose, C. (2015). Weakly supervised role identification in teamwork interactions. Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers) (pp. 1671-1680).

Yao, Y., Deng, J., Chen, X., Gong, C., Wu, J., Yang, J. (2020). Deep discriminative cnn with temporal ensembling for ambiguously-labeled image classification. Proceedings of the aaai conference on artificial intelligence (Vol. 34, pp. 12669- 12676).

Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D. (2018). mixup: Beyond empirical risk minimization. International conference on learning representations. Retrieved from https://openreview.net/forum?id=r1Ddp1-Rb

Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. *Advances in Neural Information Processing Systems, 28*, 649–657.

Zhang, Y., Chen, X., Zhou, D., & Jordan, M. I. (2014). Spectral methods meet em: A provably optimal algorithm for crowdsourcing. *Advances in Neural Information Processing Systems, 27*, 1260–1268.

Zhou, D., Basu, S., Mao, Y., & Platt, J. C. (2012). Learning from the wisdom of crowds by minimax entropy. *Advances in Neural Information Processing Systems, 25*, 2195–2203.

Zhou, D., Liu, Q., Platt, J., Meek, C. (2014). Aggregating ordinal labels from crowds by minimax conditional entropy. International conference on machine learning (pp. 262-270).