Heterogeneous Model Reuse via Optimizing Multiparty Multiclass Margin

Xi-Zhu Wu [1] **Song Liu** [2 3] **Zhi-Hua Zhou** [1]

Abstract

Nowadays, many problems require learning a model from data owned by different participants who are restricted to share their examples due to privacy concerns, which is referred to as *multiparty learning* in the literature. In conventional multiparty learning, a global model is usually trained from scratch via a communication protocol, ignoring the fact that each party may already have a local model trained on her own dataset. In this paper, we define a multiparty multiclass margin to measure the global behavior of a set of heterogeneous local models, and propose a general learning method called HMR (Heterogeneous Model Reuse) to optimize the margin. Our method reuses local models to approximate a global model, even when data are non-i.i.d distributed among parties, by exchanging few examples under predefined budget. Experiments on synthetic and real-world data covering different multiparty scenarios show the effectiveness of our proposal.

1. Introduction

In conventional machine learning problems, all the data are collected from a single user. However, in some medical, financial or biological scenarios, data are separately collected from different participants, who are unwilling to share their confidential datasets. Despite the concern on privacy, they want to cooperatively learn a global model from the union of all the datasets. The above problem is referred to as *multi-party learning problem* in the literature (Pathak et al., 2010).

Existing approaches for multi-party learning usually assume each party trains a homogeneous local model, that means they learn parameters of the same dimension, like neural network with the same structure (Shokri & Shmatikov, 2015). Therefore parties can communicate about the local model parameters under a designed protocol and compute a single global model. The homogeneous model assumption is reasonable if the local data distribution is identical for all parties. However, in reality the data are usually biasedly collected among parties due to temporal/spatial differences, such that using homogeneous local models is probably unrealistic.

Take the flu detection problem as an example. Doctors want to build a learning model to detect what type of virus one patient is affected based on her symptoms, for appropriate treatment. However, the types of influenza diverse geographically (Rejmanek et al., 2015), which means the distribution of patient records collected by a hospital in California may be different from those in Florida. In an extreme case, some types are unknown to the other hospital.

Assume there are 4 types of influenza in the United States. In California, 2 of 4 are commonly detected, while in Florida 3 of 4 types are often detected. We assume in the two states, doctors separately trained two models h_{CA} and h_{FL} which work locally well in California and Florida respectively. However, a direct ensemble of the two local models may not work well on all the patients. Let h_{US} denote the ideal global model trained on the combination of local datasets. When we input a patient record x, each model outputs its prediction as shown in Table 1.

Table 1: Example of flu detection on a patient x affected with type 2 flu. "$-$" means this model is not able to predict the corresponding class. Taking the maximal score as prediction, h_{FL} is consistent with h_{US}, but the combination of two local models $h_{\{\text{CA,FL}\}}$ is not since $3/4 > 4/7$.

Type	1	2	3	4
$h_{\text{US}}(x)$	$2/10$	$\mathbf{4/10}$	$1/10$	$3/10$
$h_{\text{CA}}(x)$	$-$	$-$	$1/4$	$\mathbf{3/4}$
$h_{\text{FL}}(x)$	$2/7$	$\mathbf{4/7}$	$1/7$	$-$
$h_{\{\text{CA,FL}\}}(x)$	$2/7$	$4/7$	$1/4$	$\mathbf{3/4}$

In this example, the output spaces among the two local models ($h_{\text{CA}}, h_{\text{FL}}$) and the global model ($h_{\text{US}}$) are all different.

[1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China [2]University of Bristol, Bristol, United Kingdom [3]The Alan Turing Institute, London, United Kingdom. Correspondence to: Zhi-Hua Zhou <zhouzh@lamda.nju.edu.cn>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

It is not straightforward to use a protocol on homogeneous local models to address this problem. Besides, if we directly take the maximum score out of two local models as $h_{\{\text{CA,FL}\}}$, the predicted class will be inconsistent with the ideal global model h_{US}.

Inspired by margin-based multiclass methods, we propose *multiparty multiclass margin* (MPMC-margin) to measure the correctness of an ensemble of local models. Instead of learning from scratch, we trust what a local model already learned on its own dataset, and try to slightly modify it to match the hidden global groundtruth, i.e., the unavailable combination of local datasets.

We design an iterative method called HMR (short for Heterogeneous Model Reuse) compatible with heterogeneous local models to optimize MPMC-margin, when we are permitted to share few examples under agreement. As illustrated in Figure 1, we reuse local models to make a rough global prediction, then check the MPMC-margin to find the defects of local models. Due to the nice property of MPMC-margin, the optimization can be decomposed to localized calibration operations. Since the output scores of each local model sums to 1, we also design to add a virtual "reserved" class, which catches the remaining mass, to support the calibration operation.

Our contributions are twofold. First, the proposed MPMC-margin views the multiparty learning problem from a novel perspective, serving a measurement on an ensemble of local models. Second, we design a new algorithm HMR to show the possiblity on optimizing MPMC-margin privacy friendly, which keeps most of the local data safe.

The remainder of paper is organized as follows. Section 2 formally defines our problem setting. Section 3 describes our HMR method, together with theoretical justifications. Section 4 discusses on some related topics. Section 5 reports experimental results. Finally, we conclude the paper in Section 6.

2. Preliminaries

2.1. Notations

Suppose the global learning problem is defined on a dataset $S = (X, Y) = \{(x, y) \in \mathcal{X} \times \mathcal{Y}\}$, where $\mathcal{Y}$ is a finite set of classes, i.e., $\mathcal{Y} = \{1, 2, \cdots, k\}$. The underlying global dataset S cannot be observed directly.

There are n participants and each observes a part of the global data as her local dataset $S_i = (X_i, Y_i) = \{(x, y) \in \mathcal{X} \times \mathcal{Y}_i\} \subseteq S$, where the instances are from the same input space $\mathcal{X}$ and the labels are in $\mathcal{Y}_i \subseteq \mathcal{Y}$ representing a potentially *biased* class prior comparing to the full set $\mathcal{Y}$.

Each participant is equiped with a local algorithm $\mathcal{A}_i$ to learn a local classifier $f_i : \mathcal{X} \to \mathcal{Y}_i$ on her local data S_i. f_i is based on a scoring function $h_i : \mathcal{X} \times \mathcal{Y}_i \to \mathbb{R}$. The label assigned to x is the one gives the highest score $h_i(x, y)$ where $h_i = \mathcal{A}_i(S_i)$:

$$f_i(x) = \arg\max_{y \in \mathcal{Y}_i} h_i(x, y). \tag{1}$$

Since the classifier f_i can be easily induced from the predictor h_i, we will mainly use h_i afterwards. In our previous flu example, with a slight abuse of the notation, given a specific patient x, we can say $h_{\text{CA}}(x) = (1/4, 3/4)$ or $h_{\text{CA}}(x, 3) = 1/4$ and $h_{\text{CA}}(x, 4) = 3/4$.

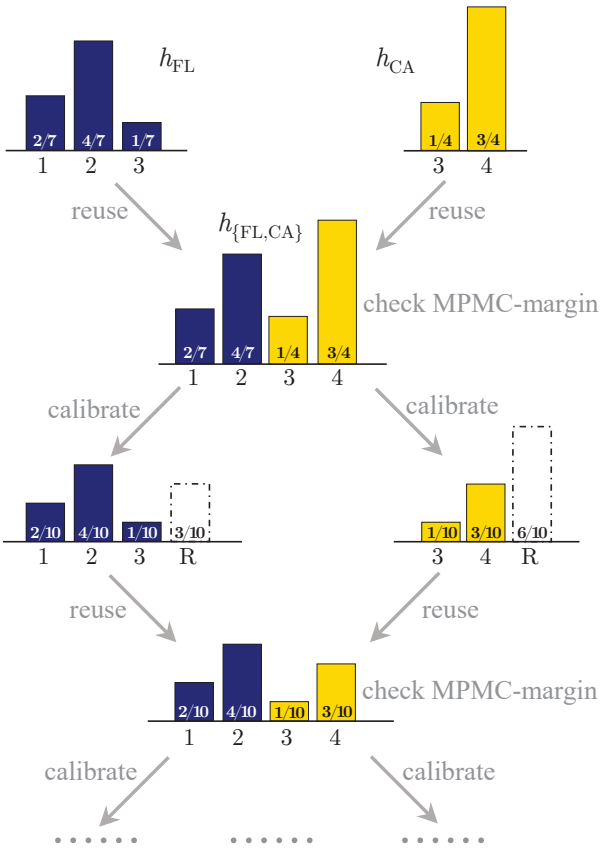


Figure 1: The illustration of running our method on the flu example. The dashed bars denote the virtual reserved classes, which are discarded at reuse step.

2.2. Assumptions

Being consistent with previous multiparty learning literature (Bellet et al., 2018), we use the word "local" to denote resources that is owned by a specific party i and cannot be accessed by other parties without the permission from i. In our setting, each party owns a local dataset, a local algorithm, and a local model. If one party receives local models from others, she can only use them as black-box predictors and is not permitted to access the model parameters or to

modify the model.

Handling distribution change is an important independent research line in domain adaptation (Sugiyama et al., 2007). Instead of the i.i.d assumption on data, we assume the distribution of local data is not identical to the underlying global data. In an extreme case, examples relevant to some classes may be nonexistent in one's dataset. This setting is more reasonable in multiparty learning because data are usually biasedly collected among participants due to temporal/spatial differences.

We assume parties are honest-but-curious: they want to know the examples in others' datasets but they strictly follow any pre-defined protocol. The local models are considered safe to be shared with other parties in a black-box manner. In this work, we tackle the global classification problem by reusing black-box heterogeneous local models, with restrictions on exchanging examples. For instance, in the flu detection problem, we can ask a small number of patients for sharing permission if they are willing to contribute to the global interest.

2.3. The ensemble of local models

It seems straightforward to combine trained local models by merging their confidence scores on each predictable class and taking the maximum class out, like the following max-model predictor:

Definition 1 (Max-Model Predictor)**.** Given a set of multiclass predictors $H = \{h_1, \cdots, h_n\}$, with $h_i : \mathcal{X} \times \mathcal{Y}_i \to \mathbb{R}$, the max-model predictor h_H is defined as

$$h_H(x, y) = \max_{y \in \mathcal{Y}_i, h_i \in H} h_i(x, y). \tag{2}$$

However, as we have already seen in the example described in Table 1, the max-model predictor $h_{\{\text{CA,FL}\}}$ may fail. Although $h_{\text{FL}}(x)$ scores $4/7$ on type 2, the merged prediction is type 4 because $h_{\text{CA}}(x, 4)$ scores highest among all classes. It looks like h_{CA} misleads the ensembled final prediction.

Why such a misleading behavior happens? Intuitively, we can think one local model has never seen a specific type of examples, and outputs very high scores on them. This is a bigger issue when the correct class label is out of model's predictable label space (type 2 flu is out of h_{CA}'s label space). Claim 1 formally state this phenomenon in the next section.

3. The HMR Method

In this section, we first propose the multiparty multiclass margin, and then describe our method in detail. We finally conclude this section with theoretical justifications proved in simple linear cases.

3.1. Multiclass margin: from single to multiple parties

Recall that in (single-party) multi-class problems, we usually define the *margin* $\rho_h(x, y)$ of the function h at a labeled example (x, y) as (Mohri et al., 2012):

$$\rho_h(x, y) = h(x, y) - \max_{y' \neq y} h(x, y'). \tag{3}$$

Thus, h misclassifies (x, y) if and only if $\rho_h(x, y) \leq 0$. The *empirical margin loss* of a hypothesis h on dataset S is

$$R_S(h) = \frac{1}{|S|} \sum_{(x,y) \in S} \ell(\rho_h(x, y)), \tag{4}$$

where ℓ is the 0-1 margin loss function:

$$\ell(\rho) = \begin{cases} 1 & \text{if } \rho \leq 0, \\ 0 & \text{if } \rho > 0. \end{cases}$$

Based on the definition of multiclass margin and margin loss, we can now formally state the failure of max-model predictor defined in Definition 1.

Claim 1 (Max-model predictor may fail)**.** *Suppose we have a set of optimal local predictors $H = \{h_1, \cdots, h_n\}$. Each of them gets zero empirical margin loss, which means $\forall i \in [n], R_{S_i}(h_i) = 0$. The max-model predictor h_H based on this set is not ensured to have zero margin loss ($R_S(h_H) = 0$) on the combined dataset $S = \bigcup_{i=1}^{n} S_i$.*

Proof. It is sufficient to prove Claim 1 with a counterexample. Suppose there are two parties A and B, each with optimal local models $h_A^\star$ and $h_B^\star$. The following positive margin conditions ensure zero empirical margin loss $R_{S_A}(h_A^\star) = R_{S_B}(h_B^\star) = 0$ on local datasets:

$$\begin{gathered} \forall i \in \{A,B\}, \forall (x, y) \in S_i, \forall y' \in \mathcal{Y}_i \backslash \{y\}, \\ h_i^\star(x, y) > h_i^\star(x, y'). \end{gathered} \tag{5}$$

But a specific $(x_a, y_a) \in S_A$ can be incorrectly predicted by $h_{\{A,B\}}$ if there exists $y' \in \mathcal{Y}_B \backslash \{y_a\}$,

$$h_B^\star(x_a, y') > h_A^\star(x_a, y_a). \tag{6}$$

Because (6) does not violate (5), the max-model predictor is likely not optimal on $S_A \cup S_B$. □

Claim 1 shows local optimal margin does not ensure the performance of a combination of local models using the maximum operator. We need a new metric to measure the behavior of the ensembled predictor. Therefore we define the multiparty multiclass margin:

Definition 2 (Multiparty Multiclass Margin)**.** The multiparty multiclass margin (MPMC-margin) on the local model

function set $H = \{h_1, \cdots, h_n\}$ at a labeled example (x, y) is defined as:

$$\rho_H(x, y) = \max_i h_i(x, y) - \max_{j,y'} h_j(x, y'), \quad \text{where } y \in \mathcal{Y}_i, y' \in \mathcal{Y}_j \backslash \{y\}. \tag{7}$$

Accordingly, the empirical MPMC-margin loss of H on $S = \bigcup_{i=1}^n S_i$ is:

$$R_S(H) = \frac{1}{|S|} \sum_{(x,y)\in S} \ell(\rho_H(x, y))$$

Evidently, when there is only one party ($n = 1$), the MPMC-margin degenerates to standard multiclass margin defined in (3).

To the best of our knowledge, it is the first time the performance of a multiparty learning problem over a set of local models is measured by margin. MPMC-margin serves a generalization of multiclass margin in multiparty scenario, and the empirical MPMC-margin loss measures the fraction of misclassified examples.

In the single-party setting, we are able to compute the multiclass margin and minimize (4) on all the training examples. However, considering that examples and local models are separately stored in the multiparty scenario, the minimization of MPMC-margin loss is nontrivial. We now present an optimization algorithm.

3.2. The procedure

Algorithm 1 shows the brief procedure of our method. Initially, we require each party to broadcast her local model to others while receiving others' model for computing MPMC-margin on her own data.

Our method is iteartive. At each iteration, we first choose a party i with probability $|S_i|/\sum_{i=1}^n |S_i|$. Then party i randomly selects one local example $(x, y) \in S_i$, and compute the MPMC-margin $\rho_H(x, y)$. Meanwhile, party i selects i^+, i^- via the selection criterion:

$$\begin{aligned} i^+ &= \arg\max_i h_i(x, y), \text{ where } y \in \mathcal{Y}_i, \\ (i^-, y^-) &= \arg\max_{j,y'} h_j(x, y'), \text{ where } y' \in \mathcal{Y}_j \backslash \{y\}. \end{aligned} \tag{8}$$

Therefore, the MPMC-margin is now represented as

$$\rho_H(x, y) = h_{i^+}(x, y) - h_{i^-}(x, y^-) \tag{9}$$

To minimize the empirical MPMC-margin loss, it is reasonable to increase the margin if it is small or negative. As (9) indicates, we can enlarge it by increasing the first term and/or decreasing the second term.

Algorithm 1 HMR

input:
Parties $1, 2, \cdots, n$, each owns a local dataset S_i and a local model h_i. Example communication budget N.
output:
Calibrated local models $h_1, \cdots, h_n$.
procedure:
1: Each party broadcasts its local model to others.
2: Inner iteration counter $T = 0$
3: **while** $T < N$ **do**
4: Sample a party i according to $|S_i|/\sum_{i=1}^n |S_i|$.
5: Party i randomly selects an example $(x, y) \in S_i$.
6: Party i computes MPMC-margin $\rho_H(x, y)$ according to (7). Records the party i^+, i^- and maximum incorrect class y^- as in (8).
7: **if** $\rho_H(x, y) \leq 0$ **then**
8: Party i sends (x, y, y^-) to i^+ and i^-.
9: Party i^+ *calibrates* h_{i^+} with (x, y, y^-).
10: Party i^- *calibrates* h_{i^-} with (x, y, y^-).
11: Party i^+ and i^- broadcast their updated model.
12: **if** $i^+ \neq i$ or $i^- \neq i$ **then**
13: $T = T + 1$.
14: **end if**
15: **end if**
16: **end while**

The first problem is, h_{i^+} and h_{i^-} may be black-boxes trained by other parties, so party i itself who owns the example (x, y) and also computes the margin cannot modify these functions. Thus we let party i dispatch the example to the model owners i^+ and i^- then leave the modification work to them. Line 9 and 10 describe the modification work which is called *calibration operation*. We will talk about it in detail later. After the calibration operation, MPMC-margin is supposed to be enlarged (a simple justification on linear predictors is presented in Theorem 1). At the end of the iteration, party i^+ and i^- broadcast their updated models to other parties, ready to be used for next iteration.

To protect local examples, the number of example communications is restricted by a total budget N. Notice that MPMC-margin can be safely evaluated as many times as we need, and only the calibration operation may consume one of the budget. It is possible that $i^+ = i^- = i$, which means the model owner and the data owner are the same, where no example is sent. Our algorithm stops if the budget runs out. For simplification, we use a global budget here. Local communication budgets can be easily applied with slight modification on algorithm. We will see in the experiments that a small budget (less than 1% of the global data) is sufficient to reach satisfactory results. Early stopping before using up the budget is also possible in practice by checking the local convergence.

3.3. The calibration operation

In Algorithm 1 Line 9-10, we use the word *calibrate* to describe the operation that one party slightly modifies her local model for global benefits. Specifically, we hope to increase $h_{i^+}(x, y)$ and decrease $h_{i^-}(x, y^-)$ to enlarge the MPMC-margin, when a non-positive margin violation is found.

The calibration operation differs on different local models. For a linear model or other models that supports online update, the operation can be done by feeding the received example into an update step. If the local models cannot be updated online, we can augment the local dataset with the newly received example and retrain the model on it. As a classification problem, we assume each party uses any surrogate loss of 0-1 classification loss, provided that the loss decreases when the predicted score on correct class increases, and vice versa. Common loss functions like logit loss and cross entropy loss satisfy this mild condition.

Notice that $y \in \mathcal{Y}_{i^+}$, party i^+ can add (x, y) to her local dataset, and retrain h^+ on the augmented dataset. The local loss minimizer tends to increase the first term $h_{i^+}(x, y)$ in (9). However, for party i^- the calibration operation is more complicated. There is no off-the-shelf way to reduce $h_{i^-}(x, y^-)$ by augmenting data.

Here we assume the outputs of h_{i^-} sums up to 1. When $y \in \mathcal{Y}_{i^-}$, it is clear that adding (x, y) into S_{i^-} will increase $h_{i^-}(x, y)$ but will decrease $h_{i^-}(x, y^-)$. When $y \notin \mathcal{Y}_{i^-}$, we propose to handle this issue by adding a virtual *reserved class* into h_{i^-}. The received example is marked as (x, R) if $y \notin \mathcal{Y}_{i^-}$, where R represents the reversed class. Party i^- then learns h_{i^-} on the augmented local dataset with expanded label space. Although the reserved class will be ignored in max-model prediction and computing MPMC-margin, adding (x, R) will decrease $h_{i^-}(x, y^-)$ as h_{i^-} tends to label x as R.

3.4. Properties

In this subsection, we present some theoretical insights about HMR on simple linear local models, in order to explain the mathematical rationality behind our design.

We formally define the multiclass linear predictor first. Let $\mathcal{Y} = \{1, 2, \cdots, k\}$ and let $\mathcal{X} \in \mathbb{R}^m$. We define the class-sensitive feature mapping $\Psi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$, where $d = mk$, as follows

$$\Psi(x, y) = [\underbrace{0, \cdots, 0}_{\in \mathbb{R}^{(y-1)m}}, \underbrace{x_1, \cdots, x_m}_{\in \mathbb{R}^m}, \underbrace{0, \cdots, 0}_{\in \mathbb{R}^{(k-y)m}}] \tag{10}$$

Then a linear predictor h is defined by $w \in \mathbb{R}^d$ as $h(x, y) = \langle w, \Psi(x, y) \rangle$. We have the following claim to show learning a combination of local models is no worse than learning a single global model in terms of hypothesis class complexity:

Claim 2. *In terms of the complexity of hypothesis class, the complexity of a set of linear predictors is no less than that of a single linear predictor with the same parameter dimension.*

Proof. It is straightforward to show the argument. For each linear predictor $h_w(x, y) = \langle w, \Psi(x, y) \rangle$, a max-model predictor based on a set of same linear predictors $H = \{h_w, h_w, \cdots\}$ performs the same as h_w. Therefore, in any measurement of hypothesis class complexity, e.g., fat-shattering dimension (Bartlett et al., 1994), the max-model predictor has at least the same level of richness. □

Next we show our method enlarges MPMC-margin in the setting of two parties with linear predictors being aware of the full label space. The results on more parties are similar.

Suppose there are two parties A and B equipped with linear predictors defined by w_A and w_B. Assume $\mathcal{Y}_A = \mathcal{Y}_B = \mathcal{Y}$, the calibration operation of Algorithm 1 in Line 9-10 is:

$$\begin{aligned} w_{i^+}^{(t+1)} &= w_{i^+}^{(t)} + \eta \Psi(x, y), \\ w_{i^-}^{(t+1)} &= w_{i^-}^{(t)} - \eta \Psi(x, y^-), \end{aligned} \tag{11}$$

where $\eta > 0$ controls the step size.

Theorem 1. *Assume $\|x\| = 1$, then the calibration operation described in (11) on linear predictors $\{h_A, h_B\}$ defined by $\{w_A, w_B\}$ will increase the MPMC-margin on sent example (x, y) by at least η.*

Proof Sketch. Due to the space limit, we present the detailed proof in Appendix A. The main idea of this proof is to enumerate possible cases when computing MPMC-margin and analyze the effect of the calibration operation on $\rho_H(x, y)$. The non-positive margin violation condition, and properties of $\Psi(x, y)$ are also important in analysis. □

Theorem 1 shows the MPMC-margin increases after calibration operation on multiclass linear predictors, which is a desirable property for improving the performance on the global problem. Whether such a nice property holds for other models depends on specfic calibration operations. At least, we can conduct the calibration operation by augmenting local data and retrain the model as described in Section 3.3. It is also possible to propose other algorithms as long as the MPMC-margin can be increased.

4. Related Work

In this section, we discuss the relationships and differences with related work.

Learning with rejection (Cortes et al., 2016): The goal of this learning paradigm is to learn a function to produce

$\{0, 1, \text{Reject}\}$ predictions from data with $\{0, 1\}$ labels, minimizing the misclassification rate (Chow, 1970; Fumera et al., 2000) or rejection loss (Bartlett & Wegkamp, 2008). In Perelló-Nieto et al. (2016), the reject option is used to make classifers more reliable and versatile, which is highly related to our technique. Nevertheless, "reject" class in the literature usually used to describe a virtual class that no datum in this class is observed, while in our work, one party will mark the data point out of her label space to be "reserved". The term "reserved" indicates that a example is likely not in one's label space, and stores the predicted mass to lower the confidence on other classes.

Private multi-party learning: In secure multi-party computation (SMC) (Lindell & Pinkas, 2008), cryptographic techniques are used to compute a function on multi-party data, to ensure none of the parties learn anything about others besides what may be inferred from the final result of the computation. Unfortunately, in machine learning, a learned model itself as the computed final result leaks private information about the training data. To ease the problem, differential privacy (Dwork, 2011) techniques are proposed. Existing approaches for differential private multi-party learning (Pathak et al., 2010; Rajkumar & Agarwal, 2012; Konecný et al., 2016) usually assume homogeneous local models, which limits the usage in biased collected multi-party data. Bellet et al. (2018) studied a problem similar to ours where local data distribution differs, but they require a prior similarity measure between parties. Hamm et al. (2016) and Bassily et al. (2018) proposed model agnostic methods like ours, which requires pre-prepared additional public data, while our method shares a limited existing local data guided by evaluating the MPMC-margin on the fly.

Dynamic classifier selection (Cruz et al., 2018): Dynamic classifier selection (DCS) techniques usually estimate the competence level of base models on a new test example, and select the most competent one to predict. DCS methods often take advantage of heterogeneous models to improve performance. However, most DCS strategies require accessing (Ko et al., 2008) or manipulating (Soares et al., 2006) all the data, which cannot be directly used in our multi-party setting. Furthermore, in contrast to explicitly competence estimation (Woods et al., 1997; Zhu et al., 2004), HMR can be seen as a way to implicitly establish a competition mechanism and does not need extra computations to compete.

Model Reuse (Zhou, 2016): Model reuse methods aim at reusing pre-trained models to help related learning tasks. In the literature, it is also named as hypothesis transfer learning (Kuzborskij & Orabona, 2013), or learning from auxiliary classifiers (Duan et al., 2009). Besides the well-known technique – finetuning pre-trained neural networks, many model reuse methods like biased regularization (Tommasi et al., 2014; Ye et al., 2018) and refining random forests (Segev et al., 2017) are proposed. Theoretical study of model reuse also attracted attentions in recent years (Kuzborskij & Orabona, 2017; Du et al., 2017; Zhao et al., 2018). Our work fits in the model reuse paradigm for solving multiparty learning problems.

5. Experiments

In this section, we validate our method on toy example, benchmark data and real-world multi-lingual handwriting data. The toy example serves as a visualization of our algorithm. Experiments on benchmark data demonstrate our method on various biased data distribution scenarios. Finally, experiment on multi-lingual data shows that our method can get satisfying performance on real problems, within limited budget. Heterogeneous local models are implemented in these experiments.

5.1. Toy example for visualization

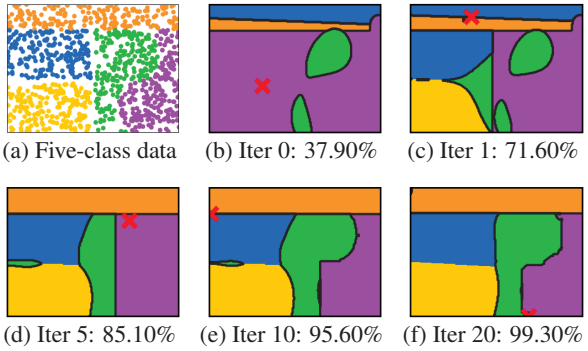
Here we create a 2D toy example with five classes as shown in Figure 2a, including 1000/1000 points for train/test. Assume there are three parties equipped with different local models. Party P_{LR} uses logistic regression as her model, party P_{SVM} uses Gaussian kernel SVM, and P_{GBDT} uses gradient boosting decision tree. These three models may be the most commonly used learning models besides deep neural nets. Implementations in scikit-learn (Pedregosa et al., 2011) with default parameters are used for easy reproduction.

We denote the 5 classes by the color of points. Each party sees partial data points, as their local datasets. P_{LR}: {blue, yellow}, P_{SVM}: {green, purple} and P_{GBDT}: {purple, orange}. We set the example budget to 50, but we found HMR converges and stops after 24 inner iterations, without using up the budget.

We present the result at inner iteration (triggered when margin violation occurs) $0/1/5/10/20$, together with the test accuracy and decision boundaries in Figure 2b-2f. The accuracy increased from 37.90% to 99.30% after sending 20 examples, and the decision boundaries are clearly moving closer to the groundtruth. Notice that the selected examples marked by red crosses are often located in wrong areas or on incorrect boundaries, supporting our intuition that non-positive MPMC-margin examples are helpful for optimizing margin loss.

5.2. Benchmark data for understanding

Recall that our method can handle the multiparty learning problem when the local data distribution is not idential to the combined global data dsitribution. In order to show how HMR works on various biased distribution scenarios, and



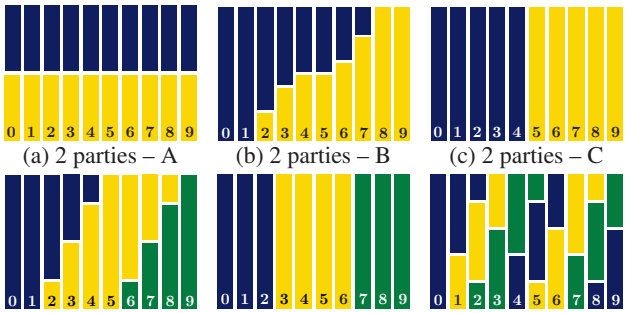
(a) Five-class data (b) Iter 0: 37.90% (c) Iter 1: 71.60%

(d) Iter 5: 85.10% (e) Iter 10: 95.60% (f) Iter 20: 99.30%

Figure 2: (2a): The data distribution of five-class points. (2b)-(2f): The decision boundaries and accuracy at plotted iteration. Red cross shows the selected example to send with non-positive MPMC-margin.

understand more about the strength and weakness of our method, we conducted experiments on Fashion-MNIST[1] (Xiao et al., 2017), which is a widely used benchmarking dataset.

Fashion-MNIST contains 70,000 28×28 grayscale fashion product images, each associated with a label from 10 classes. 10,000 out of 70,000 are used for testing. To simulate the multiparty setting, we seperate the training data into different parties according to Figure 3. For example, Figure 3d represents the class label distribution of 3 parties, where the first party sees all the data from class 0 and 1, 80%/50%/20% data from class 2/3/4. From the 2 parties setting to 7 parties setting, the data distirbution becomes more skewed, some parties even see no data from most classes.



(a) 2 parties – A (b) 2 parties – B (c) 2 parties – C

(d) 3 parties – A (e) 3 parties – B (f) 7 parties

Figure 3: Six data distribution setting between parties. Bars grouped together in the same color denote the amount of each party's local data. The proportion of the colored bar in a column precisely denotes the ratio of per class data, from class 0 to class 9.

Each party is equipped with a simple neural network with 3 conv-layers as the same structure in Google Colab [2]. To add the reserved class output on these models, we create a new neuron at the last layer, and use the average weights of other neurons at the same layer to initialize the weights of this new neuron. At calibration operation, each local model will be retrained with augmented data for one epoch. We set the example communication budget to 200, and plot the test performance of HMR. To ease the randomness, we run 10 times for each setting and show the variance band around.

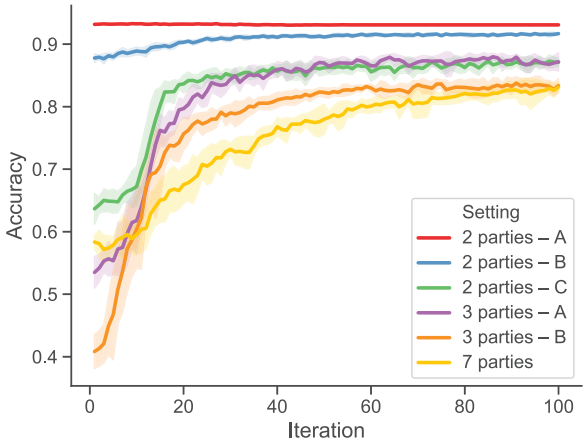


Figure 4: The test accuracy curves over iterations.

Figure 4 reports the results. When each party has enough local data with identical distribution to combined global data (2 parties – A), the direct combination of local models are fairly well. In that case, our algorithm can hardly detect any violation of MPMC-margins, and keep the local models intact. When one party is totally unaware of many classes (3 parties – B), our method can boost the overall performance a lot.

It is shown that as the number of parties increases and each party sees fewer classes, the initial performance (at iteration 0) of the max-model predictor is lower. This observation is consistent to our intuition that these local models are more severely uncalibrated because they are unaware of example of less-seen or unseen classes. The lower final accuracy on more skewed setting is reasonable as these local models lacks data to learn a good representation over all classes, which sets an upper limit of the proposed method.

5.3. Multi-lingual data for real-world problem

In this section, we demonstrate our method on multi-lingual handwriting data. Researchers using different languages often collect the handwriting data of their own scripts, and build local recognition models on them. Now if they want to cooperate to build a global model that can recognize a

[1] https://github.com/zalandoresearch/fashion-mnist

[2] https://colab.research.google.com/github/tensorflow/tpu/blob/master/tools/colab/fashion_mnist.ipynb

character from any script, but do not want to share too much data with other researchers, our HMR method will be an ideal choice.

Table 2 shows the information about the handwriting character datasets of 6 scripts. We collect them from different sources. The first three are different scripts about Janpanese[3]. Devanagari is collected from native Hindi speakers (Bharath & Madhvanath, 2009). Hangul is the script of Korean (Kim & Xie, 2015), we use 30% of the original data because of memory restriction. Letter containing English alphabet is from EMNIST (Cohen et al., 2017). We rescale all the data into 64×64 grayscale images for convenience. Example images of each script are shown in Figure 5.

Table 2: Script datasets, with local model accuracies.

Script name	#instances	#classes	Accuracy
Hiragana	9600	75	95.50
Katakana	6528	48	95.83
Kanji	112384	878	99.30
Devanagari	18357	111	92.96
Hangul	156000	520	96.58
Letter	124800	26	95.04

Figure 5: Handwriting images of each script.

We seperately build a model on each script by convolutional neural networks with different structures (implementation details are in Appendix B.3), and these models' accuracies on own local test data are reported in the last column of Table 2. Then we run HMR on the set of 6 local models. The calibration operation is implemented by training one additional epoch on augmented local data. To compare with HMR, we also train a single neural network over all the collected data, serving an upper bound of the prediction accuracy. The best score that a DCS method can get assumes an oracle to select corresponding local model when predicting a test instance. The results are reported in Table 3.

Although all the local models perform well locally (above 90% as in Table 2), we can see from Table 3 that the direct ensemble of these models using max-model predictor can only reach about 72% before running our method. After selectively sending 300 examples for calibration operation,

[3] http://etlcdb.db.aist.go.jp/

Table 3: Results of HMR after 0/50/100/300 iterations. Followed with results of comparison methods.

Method	Iterations			
	0	50	100	300
HMR	72.16	76.40	84.51	94.32
Single model	96.38			
DCS with oracle	95.86			

the global performance significantly increased to 94.32%. Notice that a single model trained over all local datasets are not permitted in multiparty setting, and an ideal DCS local model selector also requires a large amount of data, thus they are not applicable under our limited budget setting. Our method reaches a comparable result by exchanging merely 0.07% of the entire combined data, while keeping most of our locally stored data unexposed.

Furthermore, comparing with the benchmark experiment, accuracy of the multi-lingual recognition task is higher, even when the class distribution is skewed among parties. We conjecture this result is due to the common underlying feature representing of handwriting strokes, which makes it easier to learn a new script when a local model has already known one (Lake et al., 2015).

6. Conclusion

In this paper, we revisit the multiparty learning problem from the perspective of margin theory and propose the MPMC-margin to measure the inconsistency between the ensemble prediction of local models and the global groundtruth. We also design a novel method HMR to optimize MPMC-margin, which accepts heterogeneous local models and handles biased data distributions. Experiments reveal the difficulties and the potential of reusing local models for multiparty learning, which further demonstrate the usefulness of our method.

Our method still sends a few examples across local parties, which may be undesirable in security-sensitive applications. In the future, it is important to design algorithms which has better privacy guarantees when optimizing MPMC-margin.

Acknowledgements

This research was supported by the National Key R&D Program of China (2018YFB1004300), NSFC (61751306), Collaborative Innovation Center of Novel Software Technology and Industrialization, and The Alan Turing Institute under the EPSRC grant EP/N510129/1.

Authors want to thank Peng Zhao, Ming Pang and Bo-Jian Hou for insightful discussions.

References

Bartlett, P. L. and Wegkamp, M. H. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9:1823–1840, 2008.

Bartlett, P. L., Long, P. M., and Williamson, R. C. Fat-shattering and the learnability of real-valued functions. In *COLT*, pp. 299–310, 1994.

Bassily, R., Thakurta, A. G., and Thakkar, O. D. Model-agnostic private learning. In *NeurIPS*, pp. 7102–7112, 2018.

Bellet, A., Guerraoui, R., Taziki, M., and Tommasi, M. Personalized and private peer-to-peer machine learning. In *AISTATS*, pp. 473–481, 2018.

Bharath, A. and Madhvanath, S. Online handwriting recognition for indic scripts. In *Guide to OCR for Indic scripts*, pp. 209–234. Springer, 2009.

Chow, C. K. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16 (1):41–46, 1970.

Cohen, G., Afshar, S., Tapson, J., and van Schaik, A. EMNIST: an extension of MNIST to handwritten letters. *CoRR*, abs/1702.05373, 2017.

Cortes, C., DeSalvo, G., and Mohri, M. Learning with rejection. In *ALT*, pp. 67–82, 2016.

Cruz, R. M. O., Sabourin, R., and Cavalcanti, G. D. C. Dynamic classifier selection: Recent advances and perspectives. *Information Fusion*, 41:195–216, 2018.

Du, S. S., Koushik, J., Singh, A., and Póczos, B. Hypothesis transfer learning via transformation functions. In *NIPS*, pp. 574–584, 2017.

Duan, L., Tsang, I. W., Xu, D., and Chua, T. Domain adaptation from multiple sources via auxiliary classifiers. In *ICML*, pp. 289–296, 2009.

Dwork, C. Differential privacy. In *Encyclopedia of Cryptography and Security*, pp. 338–340. Springer, 2011.

Fumera, G., Roli, F., and Giacinto, G. Reject option with multiple thresholds. *Pattern Recognition*, 33(12):2099–2101, 2000.

Hamm, J., Cao, Y., and Belkin, M. Learning privately from multiparty data. In *ICML*, pp. 555–563, 2016.

Kim, I. and Xie, X. Handwritten hangul recognition using deep convolutional neural networks. *International Journal on Document Analysis and Recognition*, 18(1):1–13, 2015.

Ko, A. H., Sabourin, R., and de Souza Britto Jr., A. From dynamic classifier selection to dynamic ensemble selection. *Pattern Recognition*, 41(5):1718–1731, 2008.

Konecný, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., and Bacon, D. Federated learning: Strategies for improving communication efficiency. *CoRR*, abs/1610.05492, 2016.

Kuzborskij, I. and Orabona, F. Stability and hypothesis transfer learning. In *ICML*, pp. 942–950, 2013.

Kuzborskij, I. and Orabona, F. Fast rates by transferring from auxiliary hypotheses. *Machine Learning*, 106(2): 171–195, 2017.

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Lindell, Y. and Pinkas, B. Secure multiparty computation for privacy-preserving data mining. *IACR Cryptology ePrint Archive*, 2008:197, 2008.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. Foundations of machine learning. chapter 8, pp. 183–190. MIT Press, 2012.

Pathak, M. A., Rane, S., and Raj, B. Multiparty differential privacy via aggregation of locally trained classifiers. In *NeurIPS*, pp. 1876–1884, 2010.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., VanderPlas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Perelló-Nieto, M., Filho, T. M. S., Kull, M., and Flach, P. A. Background check: A general technique to build more reliable and versatile classifiers. In *ICDM*, pp. 1143–1148, 2016.

Rajkumar, A. and Agarwal, S. A differentially private stochastic gradient descent algorithm for multiparty classification. In *AISTATS*, pp. 933–941, 2012.

Rejmanek, D., Hosseini, P. R., Mazet, J. A., Daszak, P., and Goldstein, T. Evolutionary dynamics and global diversity of influenza a virus. *Journal of Virology*, 89(21):10993–11001, 2015.

Segev, N., Harel, M., Mannor, S., Crammer, K., and El-Yaniv, R. Learn on source, refine on target: A model transfer learning framework with random forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1811–1824, 2017.

Shokri, R. and Shmatikov, V. Privacy-preserving deep learning. In *ACM SIGSAC*, pp. 1310–1321, 2015.

Soares, R. G. F., Santana, A., Canuto, A. M. P., and de Souto, M. C. P. Using accuracy and diversity to select classifiers to build ensembles. In *IJCNN*, pp. 1310–1316, 2006.

Sugiyama, M., Krauledat, M., and Müller, K. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8:985–1005, 2007.

Tommasi, T., Orabona, F., and Caputo, B. Learning categories from few examples with multi model knowledge transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(5):928–941, 2014.

Woods, K. S., Kegelmeyer, W. P., and Bowyer, K. W. Combination of multiple classifiers using local accuracy estimates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):405–410, 1997.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Ye, H., Zhan, D., Jiang, Y., and Zhou, Z. Rectify heterogeneous models with semantic mapping. In *ICML*, pp. 1904–1913, 2018.

Zhao, P., Cai, L., and Zhou, Z. Handling concept drift via model reuse. *CoRR*, abs/1809.02804, 2018.

Zhou, Z. Learnware: on the future of machine learning. *Frontiers of Computer Science*, 10(4):589–590, 2016.

Zhu, X., Wu, X., and Yang, Y. Dynamic classifier selection for effective mining from noisy data streams. In *ICDM*, pp. 305–312, 2004.