Distributed Classification by Divide and Conquer Approach

1st Max Y. Chen
Illinois Mathematics and Science Academy
1500 Sullivan Rd
Aurora, IL 60506, USA
mchen2@imsa.edu

2nd Qiang Wu
Department of Mathematical Sciences
Middle Tennessee State University
Murfreesboro, TN 37132, USA
qwu@mtsu.edu

2024 IEEE 3rd International Conference on Computing and Machine Intelligence (ICMI) | 979-8-3503-7297-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICMI60790.2024.10586091

***Abstract*—In this paper, we investigate the efficacy of the divide and conquer approach for implementing distributed logistic regression and distributed support vector machine (SVM) algorithms for classification of large-scale datasets. This approach is designed to handle datasets that exceed the capacity of a single processor, necessitating the partitioning of data into multiple subsets. Logistic regression or SVM is then applied to each subset, yielding individual local classifiers. Subsequently, a global classifier is derived by aggregating these local classifiers to make the final decision. We propose three strategies for the aggregation stage: voting based on predicted labels, averaging of real-valued predictions, and averaging of posterior probabilities. Our analysis reveals that for distributed logistic regression, probability averaging is the most robust approach and is therefore recommended. Conversely, in the context of distributed SVM, probability averaging requires additional modeling but has a minimal impact on the performance. Therefore, functional averaging is recommended instead.**

***Index Terms*—distributed machine learning, logistic regression, support vector machine, divide and conquer**

I. Introduction

Due to the rapid advancements in information technology, data collection has become significantly more accessible, leading to the ubiquity of large-scale datasets. To address the challenges posed by big data processing, various technologies and approaches have been proposed, including the utilization of GPUs or computer clusters, quantum computing, and parallel and distributed systems. Among these approaches, distributed machine learning via the divide and conquer approach has emerged as a simple yet highly effective approach for predictive analytics. Moreover, it provides an added benefit of safeguarding data privacy and confidentiality.

In the context of distributed machine learning, the divide and conquer approach is executed as follows: when confronted with a big dataset that exceeds the processing capability of a single machine, the data is initially partitioned into multiple subsets, each suitable for analysis by a single machine. This step may be omitted if the data has already been collected by different entities, with each subset naturally residing in distinct locations, or if merging them would be impermissible due to privacy or confidentiality concerns. Subsequently, each subset is independently modeled using a pre-specified machine learning method. Finally, the outcomes from all subsets are aggregated. Averaging is the most commonly employed strategy for combining the local outcomes, especially in parameter estimation and regression analysis scenarios. This method, though simple, has been proven empirically effective and theoretically optimal for a variety of learning tasks; see for example the M-estimation [1], kernel ridge regression [2], [3], kernel spectral regression [4], bias corrected regularization kernel network [5], [6] and minimum error entropy [7], [8].

In this paper, we focus on the divide and conquer approach for distributed binary classification. Unlike the parameter estimation and regression problems, where each local model's output directly represents the quantity of interest or the prediction of the target value, leading to a natural global estimation through averaging local outputs, classification models typically produce real values that serve as a basis for inferring class labels. Although averaging local outputs remains a valid means of combining local models, alternative methods have been proposed. In a prior study [9], a voting strategy was proposed and compared with averaging. The results there showed that these two strategies are comparable for most applications while averaging could be more robust in some scenarios.

Many classification algorithms can predict the posterior probabilities. For instance, the functional output of logistic regression and the posterior probability are linked via one-to-one logit mapping. This allows us to propose a new strategy for distributed logistic regression that averages the posterior probabilities. One goal of this paper is to study its effectiveness and compare it with other strategies.

Support vector machine is another effective and widely used classification algorithm. It was motivated by maximizing the margin between classes. Therefore, its functional output has natural geometric interpretations, allowing the distributed support vector machine via voting and functional output averaging to be well defined. It is not directly related to the posterior probabilities, preventing the direct use of the posterior probability averaging. Thanks to the Platt's approach [10] that fits a post-training model to transform the functional output to posterior probability, we are able to define distributed support vector machine via the probability averaging strategy using the transformed model. Exploring the effectiveness of distributed support vector machine for large scale classification is the second goal of this paper.

II. Distributed logistic regression

Logistic regression is one of the most popular binary classification approaches developed in the classical statistics context. It is motivated from maximum likelihood estimation (MLE). Given the paired data $D = \{(x_i, y_i), i = 1, \ldots, n\}$, where $x_i \in \mathbb{R}^p$ represents a p-dimensional feature vector and $y_i \in \{0, 1\}$ is the binary response. Assume y_i following a Bernoulli distribution with $\pi_i = \Pr(y_i = 1|x_i)$ satisfying a generalized linear model:

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = x_i^\top \beta, \quad \text{or} \quad \pi_i = \frac{e^{x_i^\top \beta}}{1 + e^{x_i^\top \beta}}.$$

The log-likelihood is given by

$$\ell(\beta) = \sum_{i=1}^{n} \left(y_i x_i^\top \beta - \log(1 + e^{x_i^\top \beta}) \right)$$

and the MLE estimator $\hat{\beta}$ can be obtained by maximizing ℓ using the Newton-Raphson method, the iterative reweighted least square algorithm, or their variants [11], [12]. The prediction of the label for a new data point x is given as $\hat{y} = 1$ if $x^\top \hat{\beta} > 0$ or equivalently $\hat{\pi} > 0.5$ and $\hat{y} = 0$ otherwise.

Assume there exists a true model parameter β^*. It can be proved that the MLE estimator $\hat{\beta}$ is asymptotically normal with mean β^* and covariance $I^{-1}(\beta) = O(\frac{1}{n})$, where $I(\beta)$ is the Fisher information

$$I(\beta) = -\mathbb{E}\left[\frac{\partial^2 \ell}{\partial \beta^2}\right] = O\left(n\right).$$

This implies that $\hat{\beta}$ converges to β^* in probability.

When D is big, we can use the divide and conquer approach to implement distributed logistic regression. First, we partition D randomly into m distinct subsets $D = \bigcup_{j=1}^{m} D_j$. The optimal performance is usually achieved when all subsets are of equal size $n_j = \frac{n}{m}$. Next, we apply logistic regression to each subset D_j to obtain a local estimator $\hat{\beta}_j$. Finally, we combine these estimators together to produce a final model. There are several strategies for this purpose. In [9] voting and parameter averaging methods were proposed. For each x to be classified, the voting strategy first predicts a label $\hat{y}_j(x)$ using each local estimator $\hat{\beta}_j$ and then $\hat{y}$ is defined as

$$\hat{y} = \begin{cases} 1 & \text{if } \displaystyle\sum_{j:\hat{y}_j(x)=1} n_j > \sum_{j:y_j(x)=0} n_j; \\ 0 & \text{if } \displaystyle\sum_{j:\hat{y}_j(x)=1} n_j < \sum_{j:y_j(x)=0} n_j. \end{cases}$$

In case all subsets are of equal size, this is simply the majority voting, that is, $\hat{y} = 1$ if more than $\frac{m}{2}$ local estimators predict the output as 1 and $\hat{y} = 0$ otherwise.

The parameter averaging strategy produces a final model

$$\bar{\beta} = \sum_{j=1}^{n} \frac{n_j}{n} \hat{\beta}_j$$

and the classification is determined by $\hat{y}(x) = 1$ if $x^\top \bar{\beta} > 0$ and $\hat{y}(x) = 0$ otherwise. Note that

$$x^\top \bar{\beta} = \sum_{j=1}^{n} \frac{n_j}{n} \left(x^\top \hat{\beta}_j \right).$$

So the parameter averaging is equivalent to functional output averaging. If m increases slower than n as $n \to \infty$ so that $n_j \to \infty$ is guaranteed and therefore each local estimator $\hat{\beta}_j$ is asymptotically normal, then $\bar{\beta}$ is also asymptotically normal and the covariance is

$$\sum_{j=1}^{n} \frac{n_j^2}{n^2} I_j^{-1}(\beta) = O\left(\sum_{j=1}^{n} \frac{n_j}{n^2} \right) = O\left(\frac{1}{n}\right).$$

In this paper we propose another strategy by averaging the posterior probabilities. This approach first predicts $\hat{\pi}_j(x)$ by each local estimator $\hat{\beta}_j$ and then predicts the new data point with $\hat{y}(x) = 1$ if

$$\bar{\pi}(x) = \sum_{j=1}^{n} \frac{n_j}{n} \hat{\pi}_j(x) > 0.5.$$

Since each $\hat{\beta}_j$ is consistent, by the Delta method [13], [14], every estimation $\hat{\pi}_j(x)$ converges to the true value of $\pi(x)$. Therefore, $\bar{\pi}(x) \to \pi(x)$ and the probability averaging strategy is consistent.

Although all three strategies are theoretically consistent, we expect that probability averaging to be more robust than voting and parameter averaging. In the voting strategy, there exists a vulnerability to the amplification of the impact of weak local models, particularly those yielding less confident decisions with predicted probabilities $\hat{\pi}_j$ close to 0.5. On the other hand, in parameter averaging, it is important to note that the decision is based on the sign of the weighted average of $x^\top \hat{\beta}_j$. This means that local models giving abnormally large values of $x^\top \hat{\beta}_j$ may dominate the final model decision. It's worth highlighting that when each subset of data is relatively small in size, linear separability becomes more probable. It is a well-known fact that logistic regression can encounter convergence issues when dealing with nearly separable data, leading to abnormally large values of $x^\top \hat{\beta}_j$. Intuitively the probability averaging strategy is resistant to both abnormalities.

Note further that the decision boundary generated by the parameter averaging strategy is still a linear hyperplane in the feature space while voting and posterior probability averaging may lead to nonlinear decision boundaries although each local classifier is linear. This might be beneficial if linear classification is not optimal for the data under investigation.

III. Distributed support vector classification

Support Vector Machine (SVM) is inspired by the concept of large margin classifiers. In the context of binary classification, the hard margin SVM assumes that the two classes are separable and seeks to maximize the geometric distance from the two classes to the decision boundary. However, a more versatile alternative to the hard margin SVM is the soft margin

SVM. This variant allows for a trade-off between separability and model complexity, making it a more effective and popular choice in practice.

The soft margin SVM is a kernel based regularization approach with the hinge loss. Let the labels be represented as $y_i = 1$ or $y_i = -1$. The hinge loss takes the form

$$L(y_i, f(x_i)) = (1 - y_i f(x_i))_+ = \max(0, 1 - y_i f(x_i)).$$

Given a reproducing kernel K and the associated reproducing kernel Hilbert space $\mathcal{H}_K$ equipped with the norm $\|\cdot\|_K$ [15], the SVM for binary classification solve the problem

$$\hat{f} = \arg\min_{f \in \mathcal{H}_K} \frac{1}{2}\|f\|_K^2 + C \sum_{i=1}^{n} L(y_i, f(x_i)),$$

where $C > 0$ is the tuning parameter. The labels for new data can be predicted according to the sign of $\hat{f}(x)$, i.e. $\hat{y} = \text{sign}(f(x))$. SVM has shown great success in a variety of applications and its consistency has theoretical guarantees; see e.g. [16]–[21] and references therein. In particular, let

$$\mathcal{R}(\mathcal{C}) = \Pr\left[y \neq \mathcal{C}(x)\right]$$

be the classification risk of a classifier $\mathcal{C}(x)$ and

$$\mathcal{E}(f) = \mathbb{E}\left[L(y, f(x))\right]$$

be the population hinge loss. It has been proved in [22], [23] that the optimal classifier $\mathcal{C}^*$ defined by

$$\mathcal{C}^*(x) = \begin{cases} 1 & \text{if } \Pr(y = 1|x) > 0.5; \\ -1 & \text{if } \Pr(y = -1|x) > 0.5 \end{cases}$$

is a minimizer of both $\mathcal{R}$ and $\mathcal{E}$. Although the minimizer of $\mathcal{E}$ is not unique, all minimizers must have the same sign as $\mathcal{C}^*$ and for all real-valued functions

$$\mathcal{R}(\text{sign}(f)) - \mathcal{R}(\mathcal{C}^*) \leq \mathcal{E}(f) - \mathcal{E}(\mathcal{C}^*).$$

Since SVM is consistent, we have $\mathcal{E}(\hat{f}) \to \mathcal{E}(\mathcal{C}^*)$ and $\mathcal{R}(\text{sign}(\hat{f})) \to \mathcal{R}(\mathcal{C}^*)$ as $n \to \infty$, which guarantees $\text{sign}(\hat{f}(x)) \to \mathcal{C}^*(x)$ for all x in the non-degenerate domain.

In distributed SVM, once each subset D_j is trained and produces a local decision function $\hat{f}_j$, voting strategy is naturally defined according to the locally predicted labels $\text{sign}(\hat{f}_j(x))$ and the global prediction is consistent. The function averaging strategy uses

$$\bar{f}(x) = \sum_{j=1}^{m} \frac{n_j}{n} \hat{f}_j(x)$$

for the global decision.

The SVM output function $\hat{f}_j(x)$ represents the geometric distance or margin from x to the decision boundary. The lack of probabilistic interpretation prevents probability averaging from being naturally defined to implement distributed support vector machine. To overcome this problem, we adopt the Platt's approach [10] to approximate probabilistic outputs by fitting a post-training one-dimensional model

$$\Pr(y_i = 1|x_i) = \frac{1}{1 + e^{A_j \hat{f}_j(x_i) + B_j}}$$

with the predicted values $\hat{f}_j(x_i)$ for x_i in subset D_j. After the parameters A_j and B_j are estimated, $\Pr(y = 1|x)$ can then be approximated locally using each triple $(\hat{f}_j, A_j, B_j)$. Then averaging these approximated probabilities defines the posterior probability averaging model for distributed support vector machines.

TABLE I
Description of Data Sets and Classification Tasks

Classification Task	Abbreviation	n	p
Magic Gamma Telescope	MGT	19,020	10
Wireless Localization {1,2} vs {3,4}	WL	2,000	7
Student Evaluation {1,2} vs {3,4,5}	SE	5,046	32
Wilt	Wilt	4,889	5
Spambase	Spam	4,601	57
Default of Credit Card Clients	DCCC	30,000	23
APS Failure at Scania Trucks	APS	60,000	170
Epileptic Seizure Recognition	ESR	9,200	178
MNIST 5 vs 8	MNIST	12,017	786

IV. Experiments

In this section, we test the effectiveness of our distributed classification strategies against a variety of data sets and compare the results. For our study, we utilized eight different data sets from the UC Irvine Machine Learning Repository (https://archive.ics.uci.edu/ml/index) and the MNIST handwritten digits recognition data (http://yann.lecun.com/exdb/mnist/). These data sets were chosen for their diverse applications in various fields, ensuring the consistency of our results.

1) The Magic Gamma Telescope dataset contains 10 image parameters, produced by a Monte Carlo program. The purpose is to differentiate between gamma (signal) and hadron (background) in a Cherenkov gamma telescope.
2) The Wireless Indoor Localization dataset consists of seven Wi-Fi signal strengths from smartphones in an indoor space to experimentally determine if the signal strength could determine the indoor location. With four classes present, we perform binary classification by combining classes: {1, 2} vs {3, 4}.
3) The Turkiye Student Evaluation dataset has student evaluation scores provided by students from Gazi University in Ankara. It includes 28 course-specific questions and 4 additional attributes. The task is to predict the difficulty level of the course based on these factors. We categorize difficulty scores of {1, 2} as below-average and {3, 4, 5} as above-average for binary classification. Only the 28 course-specific questions were considered due to inconsistency in additional attributes.
4) The Wilt dataset, stemming from pansharpening Quickbird imagery, aims to predict whether a tree is diseased. This data set is notably imbalanced with 74 instances of the 'diseased trees' class and 4265 instances of the 'other land cover' class.
5) The Spam dataset aims to determine if an email is spam or not based upon a variety of attributes of the email, including word and character frequency.

6) The Default of Credit Card Clients data set captures customer default payments in Taiwan. The objective is to determine the credibility of customers based on 23 attributes like gender, education, payment history, etc.
7) The APS (Air Pressure System) Failure at Scania Trucks dataset consists of truck failures due to a specific component of the APS system, with other data points being failures unrelated to the APS. The goal is to diagnose whether an issue was caused due to the APS based upon 170 other factors. This data set is heavily imbalanced and has numerous missing values.
8) The Epileptic Seizure Recognition dataset contains EEG readings of one-second brain activities. The purpose is to differentiate individuals with epileptic seizures from healthy individuals.
9) The MNIST data set contains images of handwritten digits with each image consisting of $28 \times 28 = 784$ gray scale pixels. To perform binary classification, we only considered digits 5 and 8.

A summary of these nine datasets and their binary classification tasks is presented in Table I. Each task is assigned an abbreviation for the purpose of easy presentation of our experiment results below.

In our study, we partitioned each dataset into 60% training and 40% testing. The training data was further subdivided into 11 batches for distributed classification. For logistic regression, we used the `LogisticRegression` function from Python's `sklearn` library, excluding the default ℓ_2 penalty. For SVM, we used `sklearn`'s `SVC` function. We used the Gaussian kernel with bandwidth chosen via cross-validation. Before classification, data were normalized using the `StandardScalar` function from `sklearn`. Each experiment was conducted 50 times, with results measured in terms of classification accuracy and the area under the ROC curve (AUC).

Results for distributed logistic regression with all three strategies are presented in Table II and Table III. As a reference, we also included the classification accuracy and AUC of logistic regression trained from the training data without using distributed strategies. We see that the classification accuracy of all three strategies on eight tasks either have no essential differences or the differences are not statistically significant. Functional averaging showed significantly lower accuracy in spam detection. Comparing the AUC, we see that functional averaging showed surprisingly higher AUC than voting in four tasks (the MGT signal detection, student evaluation, Wilt diseased tree classification, and the default detection of credit card clients), although they have similar classification accuracy in these applications. This is probably caused by the amplified values for misclassified data near the boundaries. In three applications (the spam email detection, APS failure detection, and the ESR diagnosis), functional average gives significantly lower AUCs with large standard errors. Deeper exploration shows that the local models face rank deficiency problems that either prevents logistic regression from converging or produce extremely large values. This drives the AUC lower when the misclassified data are predicted with large and unstable function values. The probability averaging showed to be the most robust in terms of both classification accuracy and AUC.

TABLE II
CLASSIFICATION ACCURACY (IN PERCENTAGE) OF DISTRIBUTED LOGISTIC REGRESSION

Task	Distributed Logistic Regression			LR
	Voting	Func. Ave.	Prob. Ave.	
MGT	79.12 (0.38)	79.11 (0.37)	79.11 (0.37)	79.10 (0.38)
WL	92.19 (0.78)	91.31 (1.31)	92.27 (0.69)	92.32 (0.68)
SE	63.90 (0.84)	64.07 (0.83)	64.09 (0.86)	64.31 (0.86)
Wilt	96.95 (0.31)	96.82 (0.39)	97.03 (0.32)	96.78 (0.32)
Spam	92.36 (0.58)	87.77 (2.17)	92.36 (0.57)	92.41 (0.49)
DCCC	81.00 (0.29)	80.99 (0.28)	80.99 (0.28)	81.04 (0.28)
APS	98.87 (0.09)	98.65 (0.13)	98.88 (0.09)	99.05 (0.09)
ESR	83.50 (0.43)	83.12 (0.92)	83.52 (0.43)	82.26 (0.41)
MNIST	96.22 (0.27)	95.96 (0.27)	96.23 (0.28)	94.00 (0.50)

TABLE III
AUC (IN PERCENTAGE) OF DISTRIBUTED LOGISTIC REGRESSION

Task	Distributed Logistic Regression			LR
	Voting	Func. Ave.	Prob. Ave.	
MGT	77.76 (0.46)	83.86 (0.37)	83.88 (0.37)	83.87 (0.37)
WL	96.89 (0.49)	97.60 (0.66)	98.02 (0.26)	98.06 (0.24)
SE	56.22 (0.91)	60.24 (0.86)	59.88 (0.95)	60.12 (0.98)
Wilt	91.08 (2.27)	97.74 (0.53)	97.77 (0.51)	97.68 (0.54)
Spam	97.16 (0.36)	93.13 (1.79)	97.30 (0.33)	97.09 (0.27)
DCCC	66.91 (0.35)	72.52 (0.39)	72.63 (0.39)	72.24 (0.38)
APS	98.36 (0.65)	85.04 (4.93)	99.11 (0.39)	96.11 (1.26)
ESR	89.38 (0.70)	51.02 (1.45)	90.01 (0.71)	52.42 (1.06)
MNIST	99.18 (0.10)	98.58 (0.29)	99.33 (0.07)	97.67 (0.30)

Results for distributed SVM are presented in Table IV and Table V. Similarly, we included the performance of non-distributed SVM as references. All three strategies showed similar performance in terms of classification accuracy. Functional averaging and probability averaging have similar AUCs which are usually higher than or comparable to those for voting except for the 5 vs 8 classification in MNIST data where probability averaging showed lower AUC. A plausible explanation is that the SVM tries to approximate the optimal classifier $\mathcal{C}^*(x)$ and is unlikely to produce extremely large values. Therefore, transforming function values to probabilities does not help robustify the averaging decision.

V. CONCLUSIONS AND DISCUSSIONS

In this paper we studied the divide and conquer approach in distributed binary classification. Three strategies for combining local classifiers are proposed, each demonstrating effectiveness with subtle variations in performance. When logistic regression is used as the base classifier on each local subset, functional averaging performs better than voting in most situations, albeit not universally. Notably, probability averaging is found most robust and consistently comparable to or better than the other two strategies in almost all situations. On the other hand, when SVM is used as the base classifier, the difference between the three strategies seems negligible. Probability averaging does not exhibit the same degree of

TABLE IV
CLASSIFICATION ACCURACY (IN PERCENTAGE) OF DISTRIBUTED SVM

Task	Distributed SVC			SVC
	Voting	Func. Ave.	Prob. Ave.	
MGT	86.40 (0.29)	86.53 (0.29)	86.60 (0.28)	87.21 (0.28)
WL	97.33 (0.51)	97.45 (0.44)	97.34 (0.43)	98.00 (0.44)
SE	64.93 (0.78)	64.92 (0.77)	64.92 (0.80)	64.88 (0.70)
Wilt	97.59 (0.42)	97.47 (0.38)	96.96 (0.45)	98.66 (0.23)
Spam	91.13 (1.32)	91.56 (1.12)	91.54 (1.10)	93.01 (0.60)
DCCC	81.43 (0.28)	81.55 (0.26)	81.22 (0.26)	81.83 (0.26)
APS	98.67 (0.11)	98.89 (0.11)	98.74 (0.11)	99.30 (0.12)
ESR	95.96 (0.34)	96.05 (0.26)	95.77 (0.27)	97.60 (0.29)
MNIST	96.63 (0.58)	96.88 (0.58)	96.79 (0.75)	98.52 (0.58)

TABLE V
AUC (IN PERCENTAGE) OF DISTRIBUTED SVM

Task	Distributed SVM			SVM
	Voting	Func. Ave.	Prob. Ave.	
MGT	88.52 (0.39)	91.40 (0.32)	91.40 (0.32)	92.31 (0.29)
WL	99.37 (0.24)	99.69 (0.09)	99.68 (0.08)	99.79 (0.10)
SE	58.03 (1.03)	57.05 (1.17)	57.29 (1.33)	57.13 (1.62)
Wilt	94.53 (1.87)	99.17 (0.32)	99.32 (0.25)	99.14 (1.13)
Spam	95.21 (0.59)	96.59 (0.39)	96.43 (0.42)	97.28 (0.38)
DCCC	67.99 (0.61)	73.34 (0.48)	73.52 (0.52)	72.30 (0.48)
APS	94.58 (0.92)	99.10 (0.23)	99.10 (0.13)	98.76 (0.49)
ESR	97.96 (0.64)	99.29 (0.07)	99.27 (0.07)	99.52 (0.07)
MNIST	99.17 (0.16)	99.73 (0.06)	96.72 (0.05)	99.92 (0.11)

TABLE VI
CLASSIFICATION ACCURACY (IN PERCENTAGE) OF WEIGHTED DISTRIBUTED LOGISTIC REGRESSION

Task	Weighted Distributed Logistic Regression			LR
	Voting	Func. Ave.	Prob. Ave.	
MGT	79.12 (0.38)	79.11 (0.37)	79.11 (0.37)	79.10 (0.38)
WL	92.19 (0.78)	91.30 (1.31)	92.25 (0.69)	92.32 (0.68)
SE	63.90 (0.84)	64.09 (0.83)	64.07 (0.84)	64.31 (0.86)
Wilt	96.95 (0.31)	96.81 (0.40)	97.03 (0.32)	96.78 (0.32)
Spam	92.36 (0.58)	87.76 (2.19)	92.36 (0.56)	92.41 (0.49)
DCCC	81.00 (0.29)	80.99 (0.28)	80.99 (0.28)	81.04 (0.28)
APS	98.87 (0.09)	98.66 (0.13)	98.88 (0.09)	99.05 (0.09)
ESR	83.50 (0.43)	83.10 (0.93)	83.52 (0.43)	82.26 (0.41)
MNIST	96.22 (0.27)	95.96 (0.27)	96.23 (0.28)	94.00 (0.50)

robustification observed in logistic regression. Considering these findings, along with the additional modeling requirements imposed by probability averaging in distributed SVM, we recommend probability averaging for distributed logistic regression and functional averaging for distributed SVM.

It is worth considering the possibility of encountering unrepresentative subsets when data is partitioned, which can result in weak local classifiers. To address this concern, we explored a solution involving the assignment of importance weights to local classifiers based on their cross-validation accuracy. We subsequently defined weighted averaging and voting strategies for distributed classification. However, our analysis indicated that such weighted approaches did not yield significant improvements, as evidenced by a comparison between the results in Table VI and Table II. This suggests that distributed learning with random partition is stable and unrepresentative subsets are uncommon in practice.

Future work may include theoretical investigations of optimal partition strategies for global convergence and more applications of distributed approaches in real-world problems.

ACKNOWLEDGEMENT

This work is partially supported by NSF (DMS-2110826).

REFERENCES

[1] J. D. Rosenblatt and B. Nadler, "On the optimality of averaging in distributed statistical learning," *Information and Inference: A Journal of the IMA*, vol. 5, no. 4, pp. 379–404, 2016.

[2] Y. Zhang, J. C. Duchi, and M. J. Wainwright, "Divide and conquer kernel ridge regression: a distributed algorithm with minimax optimal rates." *Journal of Machine Learning Research*, vol. 16, pp. 3299–3340, 2015.

[3] S.-B. Lin, X. Guo, and D.-X. Zhou, "Distributed learning with regularized least squares," *Journal of Machine Learning Research*, vol. 18, no. 92, pp. 1–31, 2017.

[4] Z.-C. Guo, S.-B. Lin, and D.-X. Zhou, "Learning theory of distributed spectral algorithms," *Inverse Problems*, vol. 33, no. 7, p. 074009 (29 pages), 2017.

[5] Z.-C. Guo, L. Shi, and Q. Wu, "Learning theory of distributed regression with bias corrected regularization kernel network," *Journal of Machine Learning Research*, vol. 18, no. 118, pp. 1–25, 2017.

[6] H. Sun and Q. Wu, "Optimal rates of distributed regression with imperfect kernels," *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 7732–7765, 2021.

[7] T. Hu, Q. Wu, and D.-X. Zhou, "Distributed kernel gradient descent algorithm for minimum error entropy principle," *Applied and Computational Harmonic Analysis*, vol. 49, no. 1, pp. 229–256, 2020.

[8] X. Guo, T. Hu, and Q. Wu, "Distributed minimum error entropy algorithms," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 4968–4998, 2020.

[9] D. Wang, H. Xu, and Q. Wu, "Averaging versus voting: A comparative study of strategies for distributed classification," *Mathematical Foundations of Computing*, vol. 3, no. 3, pp. 185–193, 2020.

[10] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*, A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, Eds. MIT Press, 2000.

[11] G. H. Givens and J. A. Hoeting, *Computational Statistics*. John Wiley & Sons, Inc., 2013.

[12] P. McCullagh and J. Nelder, *Generalized Linear Models*. Chapman & Hall, 1989.

[13] J. L. Doob, "The limiting distributions of certain statistics," *The Annals of Mathematical Statistics*, vol. 6, no. 3, pp. 160–169, 1935.

[14] S. A. Klugman, H. H. Panjer, and G. E. Willmot, *Loss Models: From Data to Decisions*. John Wiley & Sons, 2012.

[15] N. Aronszajn, "Theory of reproducing kernels," *Trans. Amer. Math. Soc.*, vol. 68, pp. 337–404, 1950.

[16] V. Vapnik, *Statistical Learning Theory*. Wiley, New York, 1998.

[17] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning*. Springer series in statistics New York, 2001.

[18] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT press, 2002.

[19] F. Cucker and D. X. Zhou, *Learning Theory: an Approximation Theory Viewpoint*. Cambridge University Press, 2007.

[20] I. Steinwart and A. Christmann, *Support Vector Machines*. Springer, 2008.

[21] Q. Wu, *Classification and Regularization in Learning Theory*. VDM Verlag, 2009.

[22] T. Zhang, "Statistical behavior and consistency of classification methods based on convex risk minimization," *Annals of Statistics*, vol. 32, no. 1, pp. 56–85, 2004.

[23] Q. Wu and D.-X. Zhou, "Analysis of support vector machine classification." *Journal of Computational Analysis & Applications*, vol. 8, no. 2, pp. 99–119, 2006.