JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS
2022, VOL. 00, NO. 0, 1–10
https://doi.org/10.1080/10618600.2022.2126484

Meta Clustering for Collaborative Learning

Chenglong Ye[a], Reza Ghanadan[b], and Jie Ding[c]

[a]Dr. Bing Zhang Department of Statistics, University of Kentucky, Lexington, KY; [b]Google Research, Mountain View, CA; [c]School of Statistics, University of Minnesota, Minneapolis, MN

ABSTRACT

In collaborative learning, learners coordinate to enhance each of their learning performances. From the perspective of any learner, a critical challenge is to filter out unqualified collaborators. We propose a framework named meta clustering to address the challenge. Unlike the classical problem of clustering data points, meta clustering categorizes learners. Assuming each learner performs a supervised regression on a standalone local dataset, we propose a Select-Exchange-Cluster (SEC) method to classify the learners by their underlying supervised functions. We theoretically show that the SEC can cluster learners into accurate collaboration sets. Empirical studies corroborate the theoretical analysis and demonstrate that SEC can be computationally efficient, robust against learner heterogeneity, and effective in enhancing single-learner performance. Also, we show how the proposed approach may be used to enhance data fairness. Supplementary materials for this article are available online.

ARTICLE HISTORY
Received March 2021
Accepted September 2022

KEYWORDS
Data integration; Distributed computing; Fairness; Meta clustering; Regression

1. Introduction

Collaborative learning has been an increasingly important area that aims to build a higher-level, simpler, and more accurate global model by combining various sources. The data from each source can be regarded as a sub-dataset of an overarching dataset. These sub-datasets are usually heterogeneous and stored in decentralized locations for various reasons. For example, each sub-dataset is from a unique research activity with domain-specific features, data are too large to be stored in one location, or the data privacy concern entails separate access to sub-datasets. Suppose each sub-dataset is handled by a *learner*. A natural way to improve the modeling performance is to integrate these learners to leverage the distributed computing resources and enlarged sample size.

The general question of "how to collaborate" has led to several recent research on collaborative learning, which we will elaborate in Section 1.1. This article aims to answer the following question: Whom to collaborate with? Selecting collaborators is crucial when not all learners are qualified, such as learners with incapable models or irrelevant sub-datasets. In particular, we suppose each sub-dataset is of a supervised nature, consisting of predictor-response pairs $(\boldsymbol{X}, Y)$. A learner tends to collaborate with those whose data exhibit the same or similar underlying $\boldsymbol{X}$–Y relationship. To that end, we propose to study the problem of *clustering for supervised relationships*. The idea is that sub-datasets exhibiting similar function relationships (between $\boldsymbol{X}$ and Y) should fall into the same cluster. An alternative view of such clustering is categorizing sub-datasets into fewer meta-datasets, offering better learning quality without inducing many estimation biases. As such, we name the problem "meta clustering." Unlike the classical learning problem of data-level clustering, our goal here is to cluster datasets instead of single data points. In this framework, learners should collaborate with those in the same cluster. We focus on the regression scenario, where each sub-dataset can be modeled by $f(\boldsymbol{X}) = E(Y|\boldsymbol{X})$ for some function f, and sub-datasets in the same cluster share the same (latent) function f. We propose a computationally efficient algorithm for meta clustering, consisting of three steps: select, exchange, and cluster. Figure 1 illustrates the main idea of the proposed method. In summary, we first train local models for each learner and select the best model. Then, each pair of learners exchange their already-learned best models. We then calculate the similarity between each pair of two learners by evaluating one's model on the other's dataset. Finally, spectral clustering is performed based on the similarity matrix.

The contribution of our work is 3-fold. First, we propose to study the problem of clustering for datasets based on the underlying supervision relationships. The problem of meta clustering naturally fits the emerging need for robust collaborations in adversarial learning scenarios. We propose a general approach named Select-Exchange-Cluster (SEC). Second, the proposed SEC method is both computationally efficient and theoretically guaranteed. We show that when the sample size of each sub-dataset is sufficiently large, the sub-datasets with the same generating function can be accurately categorized into the same cluster. Moreover, the number of clusters does not need to be specified in advance, and it can be appropriately identified in a data-driven manner. Third, we can use the proposed method in general supervised regression tasks that involve nonlinear and nonparametric learning models. It can be used for various learning tasks even if learners are not sure about the existence

CONTACT Jie Ding *dingj@umn.edu* School of Statistics, University of Minnesota, Minneapolis, MN.
Supplementary materials for this article are available online. Please go to *www.tandfonline.com/r/JCGS*.

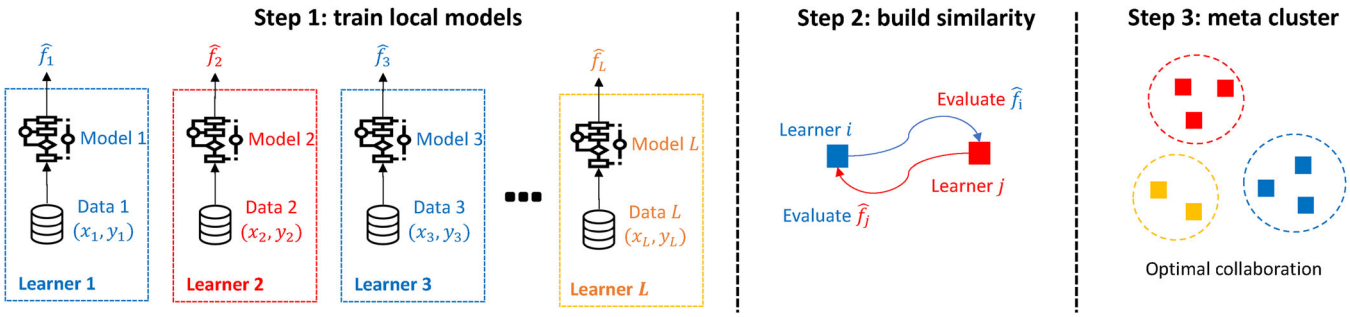


Figure 1. Illustration of meta clustering for learners/datasets, based on supervised relationships.

of latent functions. For example, we show its use to significantly enhance the prediction performance under data fairness constraints, where a reduction of approximated 50% prediction error is achieved without using any sensitive variable.

1.1. Related Work

We briefly describe the connection between meta clustering and existing research.

1.1.1. Collaborative Learning

When data are stored across distributed clients such as edge devices, directly sharing local datasets compromises data privacy. Federated learning (Konecny et al. 2016; McMahan et al. 2017; Ding et al. 2022) is a popular collaborative learning framework that aims to train a global model on distributed datasets without sharing local data. The main idea is to exchange model parameters updated from local data and iteratively update the globally trained model (assuming the same model for all clients). More general federated learning frameworks beyond exchanging parameters have been recently developed (Diao, Ding, and Tarokh 2021b, 2021c). Our proposed meta clustering framework may serve as a preliminary analysis tool for selecting "qualified" collaborators before applying any federated learning algorithm. Assisted learning (Xian et al. 2020; Diao, Ding, and Tarokh 2021a; Diao, Tarokh, and Ding 2022) is another recently developed collaborative learning framework for decentralized organizations, where any organization being assisted or assisting others does not share its local data, model, or learning objective. In assisted learning, data variables held by participants are often distinct and assumed to be linked by a non-private identifier. In contrast, our article focuses on the scenario where participants have the same variables, but the supervised relationships are possibly heterogeneous.

1.1.2. Data Integration

Data integration aims to improve statistical performance by sharing model parameters or combining datasets. Many methods have been proposed in this research direction. For example, Tang and Song (2016) developed a fused lasso approach to learn parameter heterogeneity in linear models on different datasets. Li and Li (2018) proposed an integrative method of linear discriminant analysis (LDA) for multi-type data, which was shown to improve classification accuracy over the performance on a single dataset. Jensen, Chen, and Stoeckert Jr (2007) proposed a Bayesian hierarchical model in a variable selection framework that integrates three types of data in gene regulatory networks: gene expression, ChIP binding, and promoter sequence. Yang, Yan, and Huang (2019) studied the problem of integrating regression data from different sources by pooling data for centralized learning. They proposed an objective function that estimates regression coefficients by penalizing pairwise differences between coefficients of the same covariate to identify heterogeneous and homogeneous coefficients automatically. Hector and Song (2020b) proposed a method for joint integrative analysis of multiple data sources with correlated vector outcomes under a distributed quadratic inference function framework. They assume the clustering of data sources is known. In that regard, our approach may be used as a preliminary step before applying their method when the underlying clustering structure is unknown.

In comparison with most data integration methods where statistical models are specified in each sub-dataset, our proposed meta clustering framework is model-free in the sense that it allows each learner to use different local models without sharing the form of those models. For example, one learner can use a linear model to fit a sub-dataset, while another can use a random forest. The proposed SEC algorithm only exchanges the predicted values for clustering without exchanging the parameters or the models. It is worth noting that with our meta clustering, a learner considers a binary decision whether to collaborate with another learner or not. A similar setup was also considered by Zhou et al. (2021), where the authors proposed the notion of model linkage selection for learners who share parameters of common interest. Alternatively, a learner may use a soft decision-based collaboration with others. In that direction, Shen, Liu, and Xie (2020) developed an approach that summarizes inference results from other learners as confidence density functions and then combines them using a weighting scheme. Tan, Chang, and Tang (2021) proposed a tree-based ensemble approach that integrates the prediction results from other learners as feature variables.

1.1.3. Divide-and-Conquer

Divide-and-conquer in the context of distributed learning often refers to the procedure that partitions a large dataset into sub-datasets and then combines results (e.g., p-values, coefficients) obtained from each sub-dataset. For example, Zhang, Duchi, and Wainwright (2015) proposed a method that randomly partitions the dataset into sub-datasets and fits a kernel ridge regression estimator in each sub-dataset. A simple average of local predictors is used as the global estimator, achieving minimax opti-

mal convergence rates. Mackey, Talwalkar, and Jordan (2015) proposed the Divide-Factor-Combine (DFC) framework for noisy matrix factorization, which improves the scalability and enjoys estimation guarantees. Fan et al. (2019) proposed a distributed Principle Component Analysis (PCA) algorithm for data stored across multiple locations, which performs similarly to the PCA estimator based on the whole dataset. Different assumptions of the distributed sub-datasets were also investigated, such as independent cross-sectional data (Xie, Singh, and Strawderman 2011), independent sources/studies (Claggett, Xie, and Tian 2014; Battey et al. 2015), network meta-analysis (Yang et al. 2014), high-dimensional correlated data (Hector and Song 2020a), and multi-measurements data from different experiments (Gao and Carroll 2017).

The primary goal of divide-and-conquer is to reduce computational costs via parallel computing across sub-datasets. One learner may or may not have access to all the sub-datasets. In our framework, each learner can only access its local sub-dataset. Also, divide-and-conquer methods assume the underlying relationship between the response and the predictors for each sub-dataset is the same, so combining results from all the sub-datasets is reasonable. However, the datasets in distributed storage may be heterogeneous in distributions. Identifying the potential clustering of the subs-datasets is important for bias reduction and robust modeling. For divide-and-conquer methods, meta clustering can be applied to analyze whether there are potential cluster structures on the whole dataset. If there exist cluster structures, a random splitting in divide-and-conquer may lead to a modeling bias.

The remainder of the article is outlined below. We describe the meta clustering problem in Section 2 and propose our method, together with its theoretical properties, in Section 3. In Section 4, we demonstrated a potential use of the method in fairness learning scenarios. In Sections 5 and 6, we show the performance of our method through more experimental studies. The proofs are included in the supplementary materials.

2. Problem

Suppose the dataset $\mathcal{D} := \{\mathcal{D}_i\}_{i=1}^L$ is the union of L sub-datasets. For example, $\mathcal{D}_i$ can represent the sub-dataset stored in the ith location/server, the sub-dataset from the ith study in a meta-analysis, or the sub-dataset from the ith patient in the same research project. We assume each sub-dataset $\mathcal{D}_i$ is handled by a learner l_i who considers a set of available methods $\mathcal{M}_i = \mathcal{M}_i^p \cup \mathcal{M}_i^{non}$ for data analysis. Here $\mathcal{M}_i^p$ ($\mathcal{M}_i^{non}$) denotes the parametric (nonparametric) models in $\mathcal{M}_i$. Briefly speaking, we assume a parametric model (e.g., a linear regression model) has a better convergence rate than a nonparametric one (e.g., a decision tree), and the latter is consistent in estimation. More detailed assumptions are included in the supplementary materials. The notions of parametric and nonparametric are made only for technical convenience. It is practically hard to distinguish them with finite samples, even in linear models. We refer to (Ding, Tarokh, and Yang 2018) for more discussions on this. Parallel computing can be regarded as a particular case when all sub-datasets use the same learner. Throughout the article, we will use lowercase letters (e.g., x, $\boldsymbol{x}$, a_n) to denote observed data or constants, uppercase letters to denote random variables or vectors (e.g., X, $\boldsymbol{X}$), typewriter uppercase letters (e.g., $\mathtt{A}$) to represent matrices, and calligraphy uppercase letters (e.g., $\mathcal{A}$) to represent sets.

Suppose the sub-dataset $\mathcal{D}_i$ consists of n_i independent data points, denoted by $\mathcal{D}_i = \{(y_{i,j}, \boldsymbol{x}_{i,j}) : y_{i,j} \in \mathbb{R}, \boldsymbol{x}_{i,j} \in \mathbb{R}^p\}_{j=1}^{n_i}$, from the underlying model $Y_i = f_i(\boldsymbol{X}_i) + \varepsilon_i$, where $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_L$ are independent p-dimensional random variables with a distribution function $P_{\boldsymbol{X}}(\cdot)$, and the noise $\varepsilon_i \sim \mathcal{N}(0, \sigma_i^2)$ is independent of $\boldsymbol{X}_i$. Moreover, for any $i_1, i_2 \in \{1, \ldots, L\}$, ε_{i_1} is independent of $\boldsymbol{X}_{i_2}$. We suppose the L sub-datasets consist of the same p predictors.

Let $n := n_1 + \cdots + n_L$ denote the overall sample size. Throughout the article, we assume there are K (fixed but unknown) data generating functions, namely $f_i \in \mathrm{F} = \{f^{(1)}, \ldots, f^{(K)}\}$ for $i = 1, \ldots, L$. Let $||\cdot||$ denote the Euclidean norm. Define the L_2 norm $||f||_2 = \sqrt{\int f(\boldsymbol{x})^2 P_{\boldsymbol{X}}(d\boldsymbol{x})}$ and the L_∞ norm $||f||_\infty = \operatorname{ess\,sup} |f| = \inf\{c \geq 0 : |f(\boldsymbol{X})| \leq c \text{ a.s.}\}$. We say two underlying models $f^{(i)}$ and $f^{(j)}$ are different if $||f^{(i)} - f^{(j)}||_\infty > 0$.

Our goal is to accurately cluster the L sub-datasets into K clusters, where the underlying regression functions corresponding to the sub-datasets in the same cluster are similar.

3. Method

The intuition of our method is that if two sub-datasets are from the same or similar data generating function, a modeling procedure should produce similar results on the two sub-datasets. We propose the following three-step method named Select-Exchange-Cluster (SEC), where learners communicate with their estimated regression functions.

Step 1 [Select]: Each learner uses its own sub-dataset to learn a model from a set of candidate methods $\mathcal{M}_i$. Suppose each learner conducts the half-half cross-validation to perform model selection. In particular, learner l_i splits the data D_i into two parts $\mathcal{D}_{i,1}$ and $\mathcal{D}_{i,2}$ of equal size $n_{i,1} = n_{i,2} = n_i/2$ (assuming an even n_i for simplicity). The learner applies each candidate method $\delta \in \mathcal{M}_i$ to the training set $\mathcal{D}_{i,1}$ and obtains the corresponding estimator $\hat{\delta}_{n_{i,1}}$. For learner l_i, denote the best method δ_i as the one that minimizes the mean squared error (MSE) on the test set $\mathcal{D}_{i,2}$, namely

$$\delta_i = \operatorname{argmin}_{\delta \in \mathcal{M}_i} \sum_{(y,\boldsymbol{x}) \in \mathcal{D}_{i,2}} \left(y - \hat{\delta}_{n_{i,1}}(\boldsymbol{x})\right)^2 / n_{i,2}. \tag{1}$$

The "best" method δ_i is then applied to the whole data $\mathcal{D}_i$ to estimate the underlying function f_i. Denote the resulting estimated function as $\hat{f}_i$ and its fitted mean squared error as $\hat{e}_i := \sum_{(y,\boldsymbol{x}) \in \mathcal{D}_i} \left(y - \hat{f}_i(\boldsymbol{x})\right)^2 / n_i$. To summarize, for each learner l_i, we have the non-shared data $\mathcal{D}_i$ and the shareable information $\{\delta_i, \hat{f}_i, \hat{e}_i\}$.

Step 2 [Exchange]: For any two learners, they exchange the sharable information $\{\delta_i, \hat{f}_i, \hat{e}_i\}$. In particular, denote v_{ij} as the dissimilarity between any two learners (l_i, l_j), $i \neq j$. We apply the ith learner's best estimator $\hat{f}_i$ to the jth learner's dataset $\mathcal{D}_j$ and obtain its prediction loss $\hat{e}_{i \to j} := n_j^{-1} \sum_{(y,\boldsymbol{x}) \in \mathcal{D}_j} \left(y - \hat{f}_i(\boldsymbol{x})\right)^2$,

Algorithm 1 Pseudocode for the Step 3 of SEC algorithm

Input: Number of learners L, learners/datasets $\{\mathcal{D}_i\}_{i=1}^L$, the number of clusters K (optional).
Output: The number of clusters K (if not given), and the cluster labels $c_i \in \{1,\dots,K\}, i = 1,\dots,L$.

1. Calculate the similarity matrix $\mathtt{S} \in \mathbb{R}_+^{L\times L}$, where each $\mathtt{S}_{ij} = \exp(-a v_{ij})$ and v_{ij} is given by (2).
2. If K is given, conduct the spectral clustering:
   (a) Calculate the Laplacian $\mathtt{L}_\mathtt{S}$ of $\mathtt{S}$: $\mathtt{L}_\mathtt{S} = \mathtt{D}^{-1/2}\mathtt{S}\mathtt{D}^{-1/2}$, where $\mathtt{D} := \mathrm{diag}(\sum_{j=1}^L \mathtt{S}_{1j},\dots,\sum_{j=1}^L \mathtt{S}_{Lj})$.
   (b) Compute the K largest eigenvectors of $\mathtt{L}_\mathtt{S}$: $\boldsymbol{u}_1,\dots,\boldsymbol{u}_K$. Denote $\mathtt{U} = [\boldsymbol{u}_1,\dots,\boldsymbol{u}_K] \in \mathbb{R}^{L\times K}$.
   (c) Standardize each row of $\mathtt{U}$ to have unit ℓ_2 norm. Denote the standardized matrix as $\mathtt{U}_*$.
   (d) Apply k-means clustering to the rows of $\mathtt{U}_*$ into K clusters, and record the labels $c_i, i = 1,\dots,L$.
3. If K is not given:
   (a) Sort the eigenvalues of $\mathtt{S}$ from small to large and determine K (Remark 2).
   (b) Go back to Step 2.

where the subscript $i \to j$ denotes the information flow from l_i to l_j. Similarly, we apply $\hat{f}_j$ to the dataset $\mathcal{D}_i$ and obtain the prediction loss $\hat{e}_{j\to i}$. The dissimilarity v_{ij} is then defined as the difference between their best estimators:

$$v_{ij} = |\hat{e}_{i\to j} - \hat{e}_j| + |\hat{e}_{j\to i} - \hat{e}_i|, \tag{2}$$

where $v_{ij} = v_{ji}$ for any $i \neq j$. When $i = j$, the self-dissimilarity of a learner l_i is $v_{ii} := 0$.

Step 3 [Cluster]: Based on the dissimilarity v_{ij}, a similarity matrix is constructed, which is used to cluster the L learners. In particular, we calculate a symmetric matrix $\mathtt{S}$ whose (i,j)th component is $\mathtt{S}_{ij} := \exp(-a v_{ij})$. Here, a is a tuning parameter for computational convenience. For example, when $\min_{i,j} v_{ij}$ is large and $a = 1$, $\mathtt{S}_{ij}$'s can be negligibly small for all (i,j) and thus become not distinguishable by the computer (due to its limited precision). Let $\mathcal{P} = \{1,\dots,L\}$ denote the set of labels of the L learners. For a given K, we will find a collection of sets $\{\mathcal{S}_i\}_{i=1}^K$ that forms a partition of $\mathcal{P}$. The partition is obtained by applying a spectral clustering algorithm to the matrix $\mathtt{S}$ and dividing the L learners into K groups. For completeness, we summarize the clustering step (Step 3) in Algorithm 1.

Remark 1 (Spectral clustering). There are different variants of spectral clustering in the literature. Due to technical convenience, we build on the work of (Ng, Jordan, and Weiss 2002). We will show that the spectral clustering algorithm based on the constructed similarity matrix can guarantee desirable performance.

Remark 2 (Selection of K). When K is unknown, we may add a penalty term $K \cdot \lambda_n$ in to the k-means clustering in Step 2(d) of Algorithm 1 to minimize

$$\sum_{t=1}^K \sum_{i,j\in\mathcal{S}_t} \frac{1}{2|\mathcal{S}_t|} ||\mathbf{u}_{(i)} - \mathbf{u}_{(j)}||^2 + K \cdot \lambda_n \tag{3}$$

over all possible partitions of $\mathcal{P}$ and a grid of values of K. Here, $\mathbf{u}_{(i)}$ denotes the ith row of $\mathtt{U}_*$ defined in Algorithm 1. The minimization problem (3) is equivalent to comparing the within-cluster distance over a grid of K values. We suggest $\lambda_n = O(\max(n^{-1}, u_n^4))$, where u_n is an upper bound of the convergence rates of non-parametric estimators (elaborated in the supplementary document). In practice, picking an appropriate penalty term may be complex because of the known convergence rates of nonparametric methods in Step 1. An alternative approach we suggest is using the gap statistics (Tibshirani, Walther, and Hastie 2001) that searches for the so-called "elbow point" in the curve of the sum of within-cluster mean-squared errors (namely the first term in (3)) against different K's. We will also show in the supplementary document that an adequately chosen penalty can select the correct K with a high probability.

Remark 3 (Future prediction). The clustering results may also be used for downstream collaborative learning methods, where a learner only interacts with others in the same cluster. Though prediction is not the main focus of this article, we discuss two use cases to perform prediction based on the clustering results from SEC. For any particular learner l_i, suppose it belongs to the cluster $\mathcal{S}_t$. In the first case, the sub-datasets cannot be pooled due to communication bandwidths or privacy regulations. To collaborate, other learners in the same cluster may transmit their learned models $\hat{f}_{n_j}$ $(j \in \mathcal{S}_t, j \neq i)$ to the learner l_i. Then, to predict for a future observation $\boldsymbol{x}$, the learner l_i uses the weighted average of the fitted models from learners in the same cluster, for example,

$$\sum_{i\in\mathcal{S}_t} \frac{n_i}{\sum_{j\in\mathcal{S}_t} n_j} \hat{f}_i(\boldsymbol{x}), \tag{4}$$

where the weights are proportional to the sample size. In this way, the above case does not require direct data-sharing among learners. It is worth mentioning that the weights in (4) may not be optimal for a statistical gain of prediction accuracy. We include further discussions on the statistical gain in the supplementary material. The second use case is when the sub-datasets are allowed to be pooled. Then, the learner l_i pools all the sub-datasets in a cluster $\mathcal{S}_t$ and fits one model to make future predictions. In this case, the learner i directly obtains a larger sample and thus tends to learn a better model. Nevertheless, this case requires the learners to share data, which may violate the purpose of collaborative learning.

The following theorem shows that the SEC can accurately identify the correct clusters when the overall sample size goes to infinity. Its proof is included in the supplementary materials.

Theorem 4. Under some assumptions (elaborated in the supplementary document), the labels $c_1,\dots,c_L$ produced by SEC satisfy $c_i = c_j$ if and only if $f_i = f_j$, for any i,j, with probability going to one as $n \to \infty$.

Remark 5 (Data independence). The clustering accuracy in the theorem may no longer hold if the independence of y_{ij}'s breaks

down. For longitudinal settings, for example, we may assume additional conditions on y_{ij} (e.g., a α-mixing sequence) for the proof to hold. We leave the more sophisticated analysis for dependent data as future work.

4. Application to Data Fairness

One promising application of the proposed method is to enhance data fairness. Biases inherent in data collection and techniques based on these data will not address (sometimes even worsen) the inequity for disadvantaged groups. In recent years, there have been many works to define fairness, discover unfairness, and apply algorithms to promote fairness. For example, based on the maximum likelihood principle, Kamishima, Akaho, and Sakuma (2011) proposed a prejudice remover regularizer (based on the mutual information between response and sensitive variables) for classification models. Hardt, Price, and Srebro (2016) proposed a criterion called equal opportunity (or equalized odds) for a particular sensitive variable and demonstrated how to adjust a predictor to alleviate discrimination. Zafar et al. (2017) devised a notion called positive rate disparity and proposed a method to reduce disparities in mistreatment and treatment. Verma and Rubin (2018) compared the differences among 20 fairness definitions for classification problems.

We consider a linear regression setting where the sensitive variable is independent of other variables. In particular, we generate a dataset $\mathcal{D}$ that consists of 50 sub-datasets $\{\mathcal{D}_i\}_{i=1}^{50}$, each with size $n_i = 50$ from the linear model: $Y_i = X_{1i} + 2X_{2i} - 2X_{3i} + 2X_{4i} + bR_i + \epsilon_i$, where $(X_{1i}, X_{2i}, X_{3i}, X_{4i}) \sim \mathcal{N}(0, \mathbb{I}_4)$ are the nonsensitive variables, $\epsilon_i \sim \mathcal{N}(0,1)$ is the random noise, and R_i is the sensitive variable that may induce unfairness if it were known. We consider different scales of the coefficient of the sensitive variable, $b \in \{0.01, 0.5, 1, 2, 3, 4, 5, 6, 20\}$. The sensitive variable R_i is generated from a standard normal $\mathcal{N}(0,1)$ distribution and is set to be fixed for each given i. Using a fixed value as a sensitive variable is reasonable for data fairness problems where multiple measurements exist for the same subject. For example, if $\mathcal{D}$ represents longitudinal data and each sub-dataset represents a person, then the subject-specific sensitive variable (e.g., gender, race, age, home location) is fixed for each person. We set R_i as a continuous variable in this example. We split the dataset into a training set of 30 sub-datasets (e.g., $\{\mathcal{D}_i\}_{i=1}^{30}$) and a test set of 20 sub-datasets (e.g., $\{\mathcal{D}_i\}_{i=31}^{50}$). For each learner in the test set, we further split it into two parts of the same size (e.g., $\mathcal{D}_i = \mathcal{D}_i^1 \cup \mathcal{D}_i^2$). The splitting is because we need extra data points to cluster the learners in the test set. Then, the dataset $\mathcal{D}$ is reorganized into the following three sets: the training set $\{\mathcal{D}_i\}_{i=1}^{30}$, the test set $\{\mathcal{D}_i^1\}_{i=31}^{50}$, and the validation set $\{\mathcal{D}_i^2\}_{i=31}^{50}$.The random data splitting is repeated 100 times.

For the training set, in the existence of a sensitive variable, we consider three methods of building a model: *Oracle*, *Fairness*, *SEC-Fairness*. The Oracle method directly builds a linear regression model from the training set using the sensitive variable (namely without considering fairness constraints), which is expected to have the best predictive performance. The Fairness method builds a linear regression model on the training set without using the sensitive variable since using the sensitive variable is not allowed or even available in the modeling procedure. The SEC-Fairness method finds potential groupings among the sub-datasets in the training set before building models without the sensitive variable. In particular, it first uses the SEC algorithm on the training set to cluster these 30 learners $\{\mathcal{D}_i\}_{i=1}^{30}$ into groups. Then, it uses the similarity between a learner and a cluster to identify which cluster (identified from the training set) each of these 20 learners in the test set belongs to. To measure the similarity between a learner l_i and a cluster, we use the sum of the similarities between l_i with each learner in that cluster. Then, the learner l_i belongs to a cluster if its similarity to the cluster is larger than any other cluster. In the SEC algorithm, for simplicity, each learner l_i considers two candidate modeling methods: Random Forest (Breiman 2001) (RF) and linear regression (LR), namely $\mathcal{M}_i = \{\text{RF}, \text{LR}\}$ in the "select" step. For the validation set, we evaluate the predictive performances of the models by the mean square error (MSE), which are presented in Table 1. As shown in the table, when the importance (coefficient b) of the sensitive variable is high, the SEC-Fairness reduces the MSE of Fairness by about 50% overall. One possible reason is as follows. The linear relationship between y and the variables $\{X_1, \ldots, X_4\}$ only differs in the intercept per learner. The similarity between two learners, as in the SEC algorithm, will be small if the difference between their sensitive variables $|R_i - R_j|$ is large. It is then more likely that SEC divides those with similar values of the sensitive variable into the same cluster. It is worth mentioning that the SEC-fairness method satisfies the fairness constraint since it does not use the sensitive variable at all, and the nonsensitive variables used for clustering are independent of the sensitive variable.

Table 1. Predictive performances of the three methods for the data fairness example.

	SEC-Fairness		Fairness	Oracle
b	MSE	$\hat{K}$	MSE	MSE
0.01	1.03(0.005)	1 (0)	1.03(0.005)	1.03(0.005)
0.5	1.12(0.007)	2.25(0.59)	1.20(0.007)	1.03(0.005)
1	1.41(0.02)	2.65(0.59)	2.04(0.03)	0.92(0.005)
2	1.96(0.04)	2.64(0.48)	5.19(0.09)	1.06(0.005)
3	4.31(0.22)	2.77(0.44)	12.55(0.27)	0.96(0.005)
4	4.79(0.28)	2.86(0.37)	15.16(0.36)	1.00(0.005)
5	4.21(0.12)	2.97(0.17)	18.89(0.36)	0.99(0.005)
6	12.03(0.60)	2.81(0.39)	50.58(1.04)	1.06(0.004)
20	80.47(4.90)	2.91(0.29)	438.64(9.38)	1.07(0.006)

NOTE: The values in the parentheses are the standard error of the averaged MSE and the standard deviation of the estimated number of clusters $\hat{K}$, respectively, over 100 replications.

The Oracle method, as expected, is very stable in MSE (around 1) over different values of b. When b is large, SEC-Fairness is comparable to the oracle method, though it performs better than the Fairness method. One reason is that the estimated number of clusters $\hat{K}$ is in the interval $[2, 3]$. In this example, we select $\hat{K}$ by the gap statistic. We note that there exists no "true" value of K since every learner/sub-dataset has a unique sensitive value. In the case $b = 20$, we have $\hat{K} = 2.91$. But if we force $\hat{K} = 10$ in the SEC algorithm, the MSE performance of SEC-Fairness is much improved. One reason that the gap statistic selects a small $\hat{K}$ is that it chooses the value of K that most reduces the gap compared with $K - 1$ instead of selecting K that achieves the global minimum. Consequently, the gap statistic tends to select $\hat{K}$

as 3 or 4 in this example. An alternative way to estimate K is cross-validation. Specifically, we can split each sub-dataset into a training set and a test set. Then, on the collection of the test sets, we can compare the MSE performance based on a list of K's and select the most appropriate K. The cross-validation splitting ratio for each sub-dataset will likely affect the selection of K. We recommend using half-half splitting for each sub-dataset. Because the sample size n_i, the modeling methods $\mathcal{M}_i$, and the existence of data heterogeneity are different across learners, it will be nontrivial and interesting to study how to decide the splitting ratios of cross-validation. We leave that as future work and refer interested readers to (Ding, Tarokh, and Yang 2018; Zhang, Ding, and Yang 2022) for related discussions on cross-validation.

5. Simulated Data Experiments

In this section, we present two simulation settings. Each example is repeated 100 times. From a theoretical view, no standardization of the data is required since only the function relationship between Y and $\boldsymbol{X}$ matters. So one cluster may contain two sub-datasets/learners whose responses or predictors are not on the same scale. However, the nonparametric method usually requires compact support, which may cause some computational issues. In the experiments, we standardize $\boldsymbol{x}$ and y in each sub-dataset/learner.

5.1. Simulation 1: Clustering Accuracy

This example is to demonstrate the clustering accuracy of our method. A clustering result is accurate if the number of clusters is accurately identified, and each learner's label matches the underlying truth (up to a permutation). Suppose there are 20 learners, $\{l_i\}_{i=1}^{20}$, each with a sub-dataset D_i containing $n_i = 50$ observations and $p = 5, 10, 20$ predictors. The data of the first 10 learners are generated from the underlying model $Y = f_1(\boldsymbol{X}) + \varepsilon_1 = \boldsymbol{\beta}_1^{\mathrm{T}}\boldsymbol{X} + \varepsilon_1$, where $\boldsymbol{X} \sim \mathcal{N}(0, \mathtt{I}_p)$, $\varepsilon_1 \sim \mathcal{N}(0, \sigma^2)$, and $\boldsymbol{\beta}_1 \in \mathbb{R}^p$. The data of the remaining 10 learners are generated from $Y = f_2(\boldsymbol{X}) + \varepsilon_2 = \boldsymbol{\beta}_2^{\mathrm{T}}\boldsymbol{X} + \varepsilon_2$, with $\varepsilon_2 \sim \mathcal{N}(0, \sigma^2)$, and $\boldsymbol{\beta}_2 \in \mathbb{R}^p$. We randomly generate $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ from the standard Gaussian distribution (both $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ are set as fixed in each replicated experiment such that $\boldsymbol{\beta}_1 \neq \boldsymbol{\beta}_2$). The signal-to-noise ratio (SNR) is defined by $\mathbb{E}(\|\boldsymbol{\beta}\|^2)/\mathbb{E}(\varepsilon^2)$, which reduces to p^2/σ^2 in this case. We set the SNR level to be one of the following: $2^0, \ldots, 2^7$, and the corresponding noise level $\sigma^2 = p^2/\mathrm{SNR}$ falls into the range of 25/128 to 400. In the SEC algorithm, let each learner consider two candidate methods: LASSO (Tibshirani 1996), with built-in half-half cross-validation to select the tuning parameter, and Random Forest, with 50 trees and depth 3. We apply the SEC algorithm to cluster the 20 sub-datasets. The averaged clustering accuracy over 100 replications is presented in Figure 2. We can see that the clustering accuracy increases as the SNR increases. Also, for a fixed SNR, a smaller p tends to lead to better clustering accuracy. It is mainly because a less parsimonious model suffers from more estimation variance given the same amount of data. We also see that for a fixed p, the accuracy curve tends to be flat when SNR is larger than 2^5, showing the SEC algorithm's robustness against high noise levels. In Figure 3, we also present

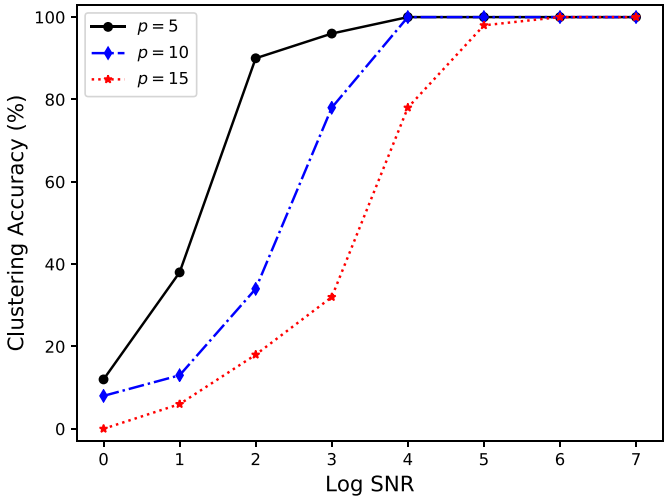


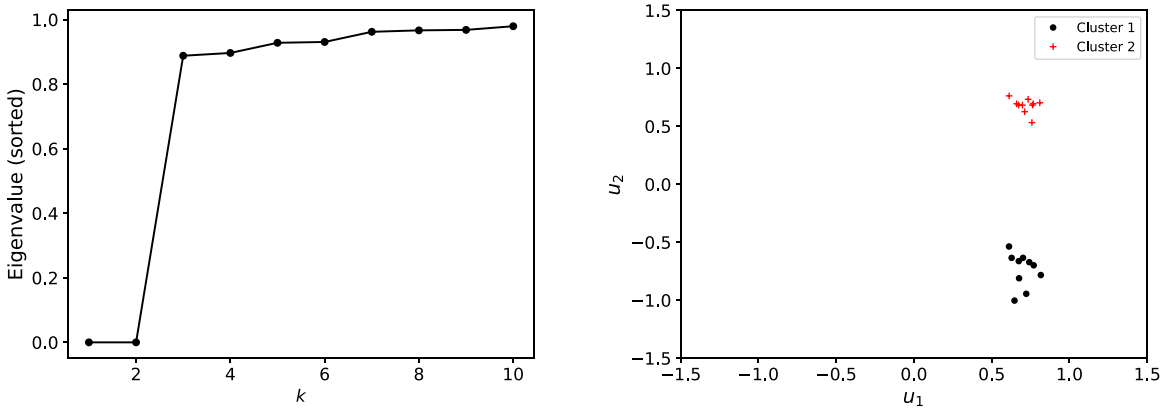
Figure 2. Clustering accuracy of the SEC algorithm for Simulation 1.

the result of a replication of the simulation with $p = 5$ and SNR$= 2^4$, with clustering accuracy near 100%. The eigenvalues used to apply the gap statistic are plotted in Figure 3(a). The eigenvectors in the spectral clustering algorithm are shown in Figure 3(b).

5.2. Simulation 2: Robustness against Candidate Models

In this example, we demonstrate that our method is robust against candidate models in the cross-validation part of the "select" step. Suppose there are 20 learners, $\{l_i\}_{i=1}^{20}$, each with a sub-dataset D_i containing $n_i = 100$ observations and $p = 500$ predictors. We use the following two benchmark datasets described in (Friedman 1991; Breiman 1996). The sub-datasets of the first 10 learners are generated from $Y = f_1(\boldsymbol{X}) + \varepsilon_1 = \sqrt{X_1^2 + (X_2X_3 - 1/(X_2X_4))^2} + \varepsilon_1$, and the sub-datasets of the remaining 10 learners are generated from $Y = f_2(\boldsymbol{X}) + \varepsilon_2 = \arctan(X_2X_3 - 1/(X_2X_4)/X_1 + \varepsilon_2$, where $X_1 \sim U(0, 100)$, $X_2 \sim U(40\pi, 560\pi)$, $X_3 \sim U(0, 1)$, $X_4 \sim U(1, 11)$, and $\varepsilon_1, \varepsilon_2 \sim \mathcal{N}(0, 0.01)$ are independent. The remaining 496 predictors $\{X_5, \ldots, X_{500}\}$ follow a standard multivariate gaussian distribution $\mathcal{N}(0, \mathtt{I}_{496})$.

For each learner, we consider the candidate methods: Random Forest (RF), k-nearest neighbors (KNN), Support Vector Regression (SVR) (Drucker et al. 1997), Neural Network (NN), Gradient Boosting (Friedman 2001) (GB), LASSO, Least Angler Regression (Efron et al. 2004) (LARS), Elastic Net (Zou and Hastie 2005) (EN), Ridge Regression (Ridge). To show the robustness of our procedure against the number of candidate models and against the types of candidate models, we consider four different choices of $\mathcal{M}_i$: {RF, KNN, SVR, NN, GB, LASSO, LARS, EN, Ridge}, {RF, KNN, SVR, GB, LASSO, LARS, EN}, {RF, KNN, GB, LASSO, LARS}, {RF, GB, LASSO}, and {GB}. The results are presented in Table 2. The clustering accuracy is stable over different choices of $\mathcal{M}_i$. We can see the robustness of our method against both the number of candidate models and the type of candidate models.



(a) Eigenvalues used to determine the gap statistic and number of clusters.

(b) Eigenvectors of the learners, indicated by the underlying true labels.

Figure 3. An illustration of the clustering results for Simulation 1, based on a realization with $p = 5$, SNR=2^4.

Table 2. Prediction performance with collaboration and without collaboration, under various sets of candidate methods (rows).

$\lvert\mathcal{M}_i\rvert$	Proportion being selected (GB, RF, LASSO)	Accuracy	$\hat{K}$	Collaboration MSE	No collaboration MSE
1	(1, 0, 0)	66.0	2	0.100(0.0056)	0.133(0.0038)
3	(0.56, 0.11, 0.33)	74.0	2	0.095(0.0053)	0.131(0.0042)
5	(0.56, 0.10, 0.34)	58.0	2	0.087(0.0049)	0.125(0.0044)
7	(0.57, 0.11,0.32)	70.0	2	0.096(0.0056)	0.134(0.0051)
9	(0.57, 0.10, 0.33)	64.0	2	0.060(0.0018)	0.112(0.0034)

NOTE: The column "Proportion being selected" is the proportion of each method being selected as the best method. The column "Accuracy" is the clustering accuracy of the SEC algorithm. The standard error of the averaged MSE over 100 replications is reported in parentheses. The $\hat{K}$ denotes the estimated number of clusters.

Without loss of generality, we focus on the first learner l_1 to evaluate whether the SEC algorithm improves prediction accuracy. We generate a test set $\mathcal{D}_{\text{test}} = \{(y_i^{\text{test}}, \boldsymbol{x}_i^{\text{test}})\}_{i=1}^{100}$ generated from the model $Y = f_1(\boldsymbol{X}) + \varepsilon_1$. We consider two modeling methods: No collaboration and Collaboration. The "Collaboration" method first applies the SEC algorithm and identifies learners in the same cluster as l_1. Then, we obtain the prediction for the test set $\mathcal{D}_{\text{test}}$ based on the simple average of the estimated predictors from those learners, as described in the formula (4). The "No Collaboration" method simply fits l_1's favored method on its own sub-dataset $\mathcal{D}_1$ and applies the estimator on the test set $\mathcal{D}_{\text{test}}$ to make predictions. The mean squared errors of the above two methods' predictions are also shown in Table 2. Overall, "Collaboration" has a smaller MSE than "No Collaboration." For $|\mathcal{M}_i| = 1$, a right-sided t-test of the MSE's of "No Collaboration" to that of "Collaboration" produces a p-value of 2.1×10^{-6}. We also observe significantly small p-values for other cases of $|\mathcal{M}_i|$. When the number of candidate models in $\mathcal{M}_i$ is larger, the MSE of the "Collaboration" method is smaller. The above is because more candidate models in the cross-validation part of the "select" step enable us to understand better the function relationship between the response and the predictors so that the similarity matrix can better capture the true underlying clusters. The prediction accuracy of the two methods is also stable across different choices of $\mathcal{M}_i$, in terms of both the size of $\mathcal{M}_i$, $|\mathcal{M}_i|$ and the methods in $\mathcal{M}_i$.

6. Real Data Applications

In this section, we apply the SEC algorithm in two real data examples.

6.1. Application 1: CT Image Data

We investigate the CT Image dataset in (Graf et al. 2011) that consists of 53500 CT slices and 385 variables. These 53500 CT slices are obtained from 97 CT scans, where 74 patients (43 male and 31 female) took at most a thorax scan and a neck scan. The response variable is the relative location of the CT slice on the axial axis. The relative location of the CT slice on the axial axis is critical for registering CT scans in a body atlas (Graf et al. 2011), which enables the comparison of different CT scans. This dataset has a natural sub-dataset structure since many CT slices are from the same CT scan that can be treated as a sub-dataset.

We divide the dataset into 97 sub-dataset/learners, each containing all the CT slices from a single CT scan. Our goal is to find any potential clustering structure (and the corresponding variable) that improves both scientific understanding and predictive performance. We randomly divide these 97 learners into two parts: the training set (64 learners) and the test set (33 learners). Similar to the data fairness example, for each of the 33 learners in the test set, we divide the sub-dataset into two sets of equal size.

For the training set, we consider three methods: "clustering (pooled)", "clustering (unpooled)", and "no clustering". The "no clustering" method directly trains a Random Forest model on the training set. The "clustering (pooled)" method first applies the SEC algorithm to classify the learner in the training set into clusters, with $\mathcal{M}_i = \{\text{RF}, \text{LASSO}\}$ for $i = 1, \ldots, 64$. Then it trains a Random Forest model separately in each identified cluster (with all the within-cluster sub-datasets pooled). In contrast, the "clustering (unpooled)" does not pool the sub-

Table 3. Results for the CT image data.

	Clustering (pooled)	Clustering (unpooled)	No clustering
MSE	94.09 (4.18)	103.07(3.79)	150.52 (2.10)
$\hat{K}$	2 (6 times) and 3 (94 times)		N/A

NOTE: The value in the parenthesis is the standard error of the averaged MSE over 100 replications, and $\hat{K}$ denotes the estimated clusters.

datasets in the cluster but trains a Random Forest in each sub-dataset. For sub-datasets/learners in the validation set, the "no clustering" method directly applies the trained random forest model to all the learners and obtains the overall mean squared error. The "clustering (pooled)" method first determines to cluster each learner belongs to and then applies the cluster-level trained random forest model. In contrast, the "clustering (unpooled)" method applied a weighted average as in Equation (4). We repeat the data splitting 100 times and summarize the results in Table 3. The results show that both options (unpooled and pooled) can significantly outperform that of the "no clustering" method. A right-sided paired t-test that compares the MSE of "clustering (unpooled)" and "clustering (pooled)" with that of "no clustering" produces p-values of 1.43×10^{-13} and 1.77×10^{-14}, respectively. The "clustering (unpooled)" improves the MSE by 31% than "no clustering," and the "clustering (unpooled)" has a slightly worse MSE compared with "clustering (pooled)." This demonstrates the promising performance of collaborative learning even without pooling data. We also looked for a scientific understanding of the identified clusters on the training set. So we investigated possible variables related to the cluster structure discovered by the SEC. Unfortunately, either the gender of the patient or whether the CT scan is from the thorax or neck is not available in the dataset (Graf et al. 2011). However, this example does show the possibility of finding essential variables related to the cluster structure if further information is provided. Additionally, we can significantly improve the predictive performance without assessing any patient private information but the CT images themselves.

6.2. Application 2: Electrical Grid Stability Data

This example is to demonstrate the performance of the SEC algorithm when the data are under adversarial attacks. The Electrical Grid Stability Data (Arzamasov, Bohm and Jochem 2018) consists of 10,000 observations and 14 variables. Among the 14 variables, two variables describe the system stability: one is categorical (stable/unstable), and the other is continuous (a positive value means a linearly unstable system). We use the continuous variable as the response. The other 12 variables are the input of the *Decentral Smart Grid Control* system.

We first divide the data into training set ($n_1 = 8000$) and the test set ($n_2 = 2000$). The training set is randomly divided into 50 learners, each with 160 observations. We may assume the data are stored in 50 servers, and some servers get attacked by hackers. Let $d = 0, 1, \ldots, 49$ denote the number of attacked learners. We set that the first d out of 50 learners are attacked. Each time a sub-dataset is "attacked," we change the response variable to the negative of its original value. We also assume that the 50th learner knows that its dataset is not attacked.

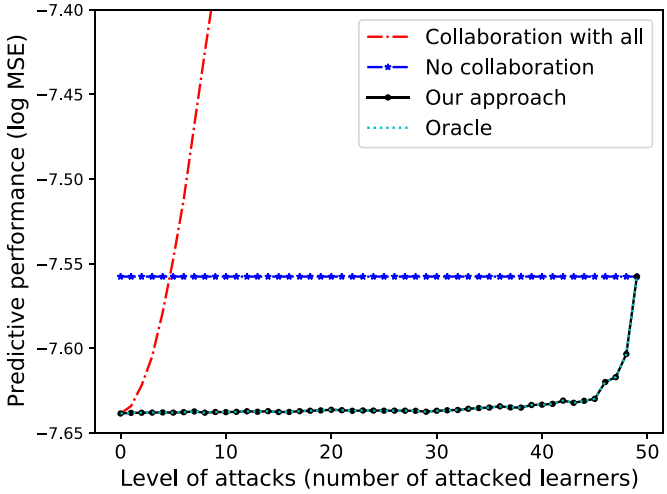


Figure 4. Prediction error (evaluated by MSE) as an increasing function of attack severity.

Under potential attacks, we consider four options of the 50th learner to perform data analysis, denoted as "Collaboration with all", "No collaboration," "Our approach," and "Oracle." The "Collaboration with all" option ignores the fact that some learners/sub-datasets are attacked and insist on collaborating with all the other learners. In the "No collaboration" option, a learner (say the 50th) trusts nobody but itself and uses its sub-dataset for learning. In the "Our approach" option, the 50 learners are clustered by SEC into "attacked" and "intact." Then the learners classified as intact will collaborate. The "Oracle" option means that an oracle knows which learners are attacked and collaborates with those intact ones. In collaboration, we allow the learners to share datasets. In other words, once a learner identifies collaborators, the learner pools the data and fits a linear regression.

The trained linear model is then applied to the test set to evaluate its performance (MSE). We plot the predictive performance against the number of attacked learners in Figure 4. We only present part of the red curve since it explodes as the level of attacks increases. The value of the red curve increases from -7.65 to -5.25 when the number of attacked learners increases from 0 to 49. As the proposed method accurately clusters all the intact learners, the performance curve of "Our approach" overlaps with that of "Oracle." We also see that the predictive performance of "Our approach" decreases when the level of attack (meaning the number of the attacked learners) increases. In particular, the decrease becomes very sharp when the number of attacked learners is greater than 45. One reason is that the linear model based on the information of one sub-dataset (with a sample size of 160 and 12 predictors) or two is enough to capture the underlying relationship. Indeed, the scale of MSE is very small (10^{-4}). So collaborating with more than five intact learners may not improve the prediction accuracy much compared with collaborating with only two intact learners.

The proposed SEC algorithm can be applied even though each learner can only access its own sub-dataset. Nevertheless, the SEC algorithm can be applied when each learner has access to all the sub-datasets. In such cases, we envision it as a pre-screening method to screen out contaminated sub-datasets, which improves modeling and prediction accuracy.

7. Conclusion

This article proposed a framework of *meta clustering* for selecting "qualified" collaborators for collaborative learning. If two datasets exhibit a similar underlying relationship between the response and predictors, they fall in the same cluster. We developed a clustering algorithm named SEC to perform meta clustering efficiently. It only requires the exchange of fitted functions instead of raw data to evaluate the similarity among datasets. We showed promising applications of the framework to enhance data fairness, improve single-learner prediction accuracy, and discover potential grouping structures of a dataset.

Supplementary Materials

The supplement materials include (i) a document that contains the technical proofs and additional discussions and (ii) Python-based codes used for the numerical studies.

Acknowledgments

We thank the anonymous reviewers and Editor for their valuable time and comments, which have helped us improve the original manuscript.

Disclosure Statement

The authors report there are no competing interests to declare.

Funding

This article is based upon work supported by the National Science Foundation under grant number ECCS-2038603.

References

Arzamasov, V., Bohm, K., and Jochem, P. (2018), "Towards Concise Models of Grid Stability," in *Proceedings of the IEEE SmartGridComm*, pp. 1–6. [8]

Battey, H., Fan, J., Liu, H., Lu, J., and Zhu, Z. (2015), "Distributed Estimation and Inference with Statistical Guarantees," arXiv preprint arXiv:1509.05457. [3]

Breiman, L. (1996), "Bagging predictors," *Machine Learning*, 24, 123–140. [6]

——— (2001), "Random Forests," *Machine Learning*, 45, 5–32. [5]

Claggett, B., Xie, M., and Tian, L. (2014), "Meta-Analysis with Fixed, Unknown, Study-Specific Parameters," *Journal of the American Statistical Association*, 109, 1660–1671. [3]

Diao, E., Ding, J., and Tarokh, V. (2021a), "Gradient Assisted Learning," arXiv preprint arXiv:2106.01425. [2]

——— (2021b), "HeteroFL: Computation and Communication Efficient Federated Learning for Heterogeneous Clients," in *Prof. ICLR*. [2]

——— (2021c), "SemiFL: Communication Efficient Semi-Supervised Federated Learning with Unlabeled Clients," arXiv preprint arXiv:2106.01432. [2]

Diao, E., Tarokh, V., and Ding, J. (2022), "Privacy-Preserving Multi-Target Multi-Domain Recommender Systems with Assisted Autoencoders," arXiv preprint arXiv:2110.13340. [2]

——— (2018), "Model Selection Techniques: An Overview," *IEEE Signal Processing Magazine*, 35, 16–34. [3,6]

Ding, J., Tramel, E., Sahu, A. K., Wu, S., Avestimehr, S., and Zhang, T. (2022), "Federated Learning Challenges and Opportunities: An Outlook," in *Proceedings of the ICASSP*, IEEE. [2]

Drucker, H., Burges, C. J., Kaufman, L., Smola, A. J., and Vapnik, V. (1997), "Support Vector Regression Machines," in *Advances in Neural Information Processing Systems*, pp. 155–161. [6]

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004), "Least Angle Regression," *The Annals of Statistics*, 32, 407–499. [6]

Fan, J., Wang, D., Wang, K., and Zhu, Z. (2019), "Distributed Estimation of Principal Eigenspaces," *The Annals of Statistics*, 47, 3009–3031. [3]

Friedman, J. H. (1991), "Multivariate Adaptive Regression Splines," *The Annals of Statistics*, 19, 1–67. [6]

——— (2001), "Greedy Function Approximation: A Gradient Boosting Machine," *The Annals of Statistics*, 29, 1189–1232. [6]

Gao, X. and Carroll, R. J. (2017), "Data Integration with High Dimensionality," *Biometrika*, 104, 251–272. [3]

Graf, F., Kriegel, H.-P., Schubert, M., Pölsterl, S., and Cavallaro, A. (2011), "2d Image Registration in ct Images using Radial Image Descriptors," in *Proceedings of the MICCAI*, pp. 607–614, Springer. [7,8]

Hardt, M., Price, E., and Srebro, N. (2016), "Equality of Opportunity in Supervised Learning," in *Advances in Neural Information Processing Systems*, pp. 3315–3323. [5]

Hector, E. C., and Song, P. X.-K. (2020a), "A Distributed and Integrated Method of Moments for High-Dimensional Correlated Data Analysis," *Journal of the American Statistical Association*, pp. 1–14. [3]

——— (2020b), "Joint Integrative Analysis of Multiple Data Sources with Correlated Vector Outcomes," arXiv preprint arXiv:2011.14996. [2]

Jensen, S. T., Chen, G., and Stoeckert Jr, C. J. (2007), "Bayesian Variable Selection and Data Integration for Biological Regulatory Networks," *The Annals of Applied Statistics*, 1, 612–633. [2]

Kamishima, T., Akaho, S., and Sakuma, J. (2011), "Fairness-Aware Learning through Regularization Approach," in *2011 IEEE 11th International Conference on Data Mining Workshops*, IEEE, pp. 643–650. [5]

Konecny, J., McMahan, H. B., Yu, F. X., Richtarik, P., Suresh, A. T., and Bacon, D. (2016), "Federated Learning: Strategies for Improving Communication Efficiency," in *NeurIPS Workshop*. [2]

Li, Q., and Li, L. (2018), "Integrative Linear Discriminant Analysis with Guaranteed Error Rate Improvement," *Biometrika*, 105, 917–930. [2]

Mackey, L., Talwalkar, A., and Jordan, M. I. (2015), "Distributed Matrix Completion and Robust Factorization," *Journal of Machine Learning Research*, 16, 913–960. [3]

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017), "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the AISTAT*, pp. 1273–1282. [2]

Ng, A. Y., Jordan, M. I., and Weiss, Y. (2002), "On Spectral Clustering: Analysis and an Algorithm," in *Advances in Neural Information Processing Systems*, pp. 849–856. [4]

Shen, J., Liu, R. Y., and Xie, M.-g. (2020), "iFusion: Individualized Fusion Learning," *Journal of the American Statistical Association*, 115, 1251–1267. [2]

Tan, X., Chang, C.-C. H., and Tang, L. (2021), "A Tree-based Federated Learning Approach for Personalized Treatment Effect Estimation from Heterogeneous Data Sources," arXiv preprint arXiv:2103.06261. [2]

Tang, L., and Song, P. X. (2016), "Fused Lasso Approach in Regression Coefficients Clustering: Learning Parameter Heterogeneity in Data Integration," *Journal of Machine Learning Research*, 17, 3915–3937. [2]

Tibshirani, R. (1996), "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society*, Series B, 58, 267–288. [6]

Tibshirani, R., Walther, G., and Hastie, T. (2001), "Estimating the Number of Clusters in a Data set via the Gap Statistic," *Journal of the Royal Statistical Society*, Series B, 63, 411–423. [4]

Verma, S., and Rubin, J. (2018), "Fairness Definitions Explained," in *2018 IEEE/ACM International Workshop on Software Fairness (FairWare)*, pp. 1–7, IEEE. [5]

Xian, X., Wang, X., Ding, J., and Ghanadan, R. (2020), "Assisted Learning: A Framework for Multi-Organization Learning," in *Proceedings of the NeurIPS*. [2]

Xie, M., Singh, K., and Strawderman, W. E. (2011), "Confidence Distributions and a Unifying Framework for Meta-Analysis," *Journal of the American Statistical Association*, 106, 320–333. [3]

Yang, G., Liu, D., Liu, R. Y., Xie, M., and Hoaglin, D. C. (2014), "Efficient Network Meta-Analysis: A Confidence Distribution Approach," *Statistical Methodology*, 20, 105–125. [3]

Yang, X., Yan, X., and Huang, J. (2019), "High-Dimensional Integrative Analysis with Homogeneity and Sparsity Recovery," *Journal of Multivariate Analysis*, 174, 104529. [2]

Zafar, M. B., Valera, I., Gomez Rodriguez, M., and Gummadi, K. P. (2017), "Fairness Beyond Disparate Treatment & Disparate Impact: Learning Classification without Disparate Mistreatment," in *Proceedings of the ICWWW*, pp. 1171–1180. [5]

Zhang, J., Ding, J., and Yang, Y. (2022), "Targeted Cross-Validation," *Bernoulli*. Available at: *https://www.bernoullisociety.org/index.php/publications/bernoulli-journal/bernoulli-journal-papers*. [6]

Zhang, Y., Duchi, J., and Wainwright, M. (2015), "Divide and Conquer Kernel Ridge Regression: A Distributed Algorithm with Minimax Optimal Rates," *Journal of Machine Learning Research*, 16, 3299–3340. [2]

Zhou, J., Ding, J., Tan, K. M., and Tarokh, V. (2021), "Model Linkage Selection for Cooperative Learning," *Journal of Machine Learning Research*, 22, 1–44. [2]

Zou, H., and Hastie, T. (2005), "Regularization and Variable Selection via the Elastic Net," *Journal of the Royal Statistical Society*, Series B, 67, 301–320. [6]