Meta Clustering for Collaborative Learning

Chenglong Ye, Jie Ding, Reza Ghanadan

Abstract

An emerging number of learning scenarios involve a set of learners/analysts each equipped with a unique dataset and algorithm, who may collaborate with each other to enhance their learning performance. From the perspective of a particular learner, a careless collaboration with task-irrelevant other learners is likely to incur modeling error. A crucial problem is to search for the most appropriate collaborators so that their data and modeling resources can be effectively leveraged. Motivated by this, we propose to study the problem of 'meta clustering', where the goal is to identify subsets of relevant learners whose collaboration will improve the performance of each individual learner. In particular, we study the scenario where each learner is performing a supervised regression, and the meta clustering aims to categorize the underlying supervised relations (between responses and predictors) instead of the raw data. We propose a general method named as Select-Exchange-Cluster (SEC) for performing such a clustering. Our method is computationally efficient as it does not require each learner to exchange their raw data. We prove that the SEC method can accurately cluster the learners into appropriate collaboration sets according to their underlying regression functions. Synthetic and real data examples show the desired performance and wide applicability of SEC to a variety of learning tasks.

Index Terms

Distributed computing; Fairness; Meta clustering; Regression.

I. Introduction

Collaborative learning has been an increasingly important area that aims to build a higher-level, simpler, and more accurate meta model by combining the data information from various learners. The data held by each learner can be regarded as a sub-dataset of an overarching dataset. These sub-datasets are usually heterogeneous and stored in decentralized locations due to various reasons, including: 1) each sub-dataset is from a unique research activity with domain-specific features; 2) the data is too large to be stored in one location; 3) the data privacy concern entails separate accesses to sub-datasets. This motivates the following general problem: If each learner holds a unique dataset that is not to be shared with others, how can they collaborate with each other in an efficient and robust manner? Here, the efficiency means both computational efficiency as a result of distributed computing resources, and statistical efficiency due to enlarged effective sample size; the robustness is against potential adversarial learners or irrelevant sub-datasets during the collaborative learning. This general question has led to several recent research on collaborative learning which will be elaborated in Section I-A.

In this paper, we aim to answer the following question: for any particular learner A, how should a particular learner choose collaborators? In particular, we suppose that each sub-dataset is of a supervised nature, consisting of predictor-label pairs (x, y). A learner tends to collaborate with those whose datasets exhibit the same (or similar) underlying x-y relationship. To that end, we propose to study the novel problem of 'clustering for supervised relations'. The idea is that sub-datasets exhibiting similar functional relationships (between x and y) should fall into the same class, so that their combination may be used for collaborative learning at a later stage. An alternative view of such a clustering is to categorize a number of sub-datasets into few meta-datasets, and thus offering better learning quality without inducing estimation biases. As such, we call the problem 'meta-clustering'. Different from the classical learning problem of data-level clustering, our goal here is to cluster datasets instead of single data points.

C. Ye is with the Department of Statistics, University of Kentucky. J. Ding is with the School of Statistics, University of Minnesota. R. Ghanadan is with Google AI. Manuscript on Oct. 2019.

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. A vanilla algorithm is for a particular learner to enumerate all the possible collaborators, and for each combination performs a joint (distributed) learning, and then apply cross-validations to identify the most appropriate collaborators. We will propose a meta-clustering algorithm that is much more computationally efficient than the vanilla method. The main idea is to let each pair of learners exchange their already-learned models, and evaluate them on the private data to calculate a similarity measure of the two sub-datasets/learners. Then a similarity matrix is constructed and a spectral clustering is performed based on that matrix. We show theoretical guarantees of the algorithm when the sample size of each sub-dataset is sufficiently large. Moreover, the number of clusters does not need to be specified in advance, and it can appropriately identified in a data-driven manner. Figure 1 illustrates the main idea of the proposed method. The method consists of three main steps, where the first is to train preliminary local models for each learner, the second is to exchange trained models among learners and build a similarity matrix, and the third is to perform a clustering based on the constructed matrix. A learner would then collaborate with those in the same cluster.

The contribution of our work is mainly three-fold. First, we propose to study the problem of clustering for datasets based on the underlying supervised relations. This problem, also named as 'meta-clustering', has not been studied in prior work to our best knowledge. The problem naturally fits the emerging need of robust collaborations in adversarial learning scenarios. Second, we propose a computationally efficient and theoretically guaranteed algorithm for meta-clustering. Third, the proposed method can be used in general supervised regression tasks that involve nonlinear and nonparametric learning models, and it can be used for a variety of learning tasks even if learners are not sure about the existence of latent functions. For example, we will show its use to significantly enhance the prediction performance under data fairness constraints, where approximately 50% prediction error reduction is achieved without using any sensitive variable.

A. Related work

We briefly describe the connection between the proposed framework and existing research topics.

Federated learning. When data is stored across decentralized servers/devices, directly sharing local datasets compromises data privacy. *Federated learning* (also known as *collaborative learning*) is a technique in machine learning that trains a global model on distributed datasets without compromising data privacy. In particular, the parameters of the local models, rather than the local datasets, are exchanged to generate the global model. Interested readers are referred to [21], [24], [26] and the references therein. The proposed meta-clustering framework serves as a tool of preliminary analysis for selecting "qualified" collaborators before applying any collaborative learning algorithm.

Data Integration. Data integration is a method that integrates information from different data sources. Either by sharing model parameters or directly combining datasets, data integration methods improve statistical performance when a global model (type) is assumed. Many methods have been proposed. For example, [27] developed a fused lasso approach to to learn parameter heterogeneity in linear model on different datasets. [22] proposed an integrative method of linear discriminant analysis (LDA) for multi-type data, which is theoretically shown to improves classification accuracy over the performance of LDA on a single data source. [18] proposed a Bayesian hierarchical model, in a variable selection framework, that integrates three types of data in gene regulatory networks: gene expression data, ChIP binding data and promoter sequence data.

In comparison to most data integration methods where statistical models (and parameters) are specified in each sub-dataset and thus estimated parameters are exchanged, the meta-learning framework allows different models for each learner and the estimated models are to be exchanged. No assumption is imposed on the form of the learning models. For example, one learner can use linear model to fit his/her sub-dataset while another leaner can use quadratic models. Indeed, for the purpose of clustering, the proposed SEC algorithm only exchanges the predicted values, without exchanging the parameters or the models.

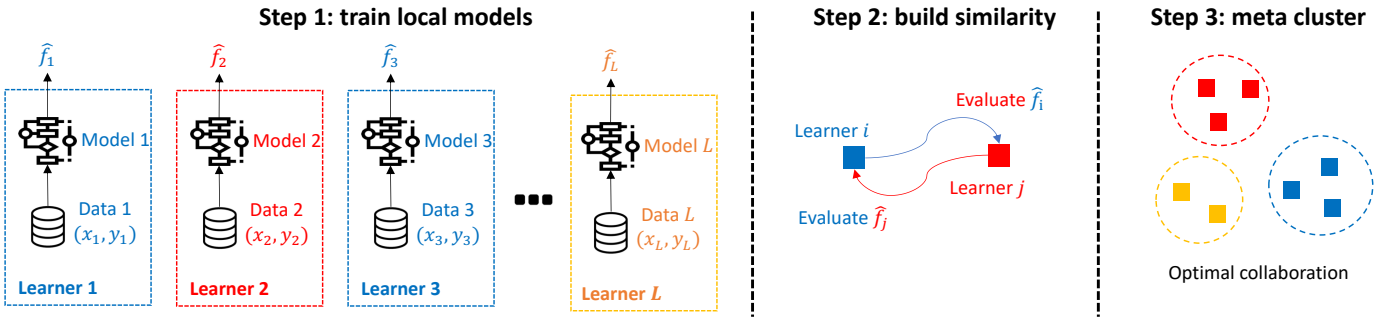


Fig. 1: An illustration of meta clustering of a set of learners/datasets, based on their underlying supervised relations.

Divide-and-conquer. Divide-and-conquer is a widely used method that involves multiple sub-datasets. It often partitions a large dataset into sub-datasets and then combines results (e.g., p-values, coefficients, etc.) obtained from each sub-dataset. The "conquer" step not only boosts computational efficiency but also adapts to different frameworks. For example, [35] proposed a method that randomly partitions the dataset into sub-datasets and fits a kernel ridge regression estimator in each sub-dataset. The simple average of the local predictors is used as the global estimator, which achieves minimax optimal convergence rates. [23] proposed the DFC (**D**ivide-**F**actor-**C**ombine) framework for noisy matrix factorization, which improves the scalability and enjoys estimation guarantees. [10] proposed a distributed PCA (**P**rinciple **C**omponent **A**nalysis) algorithm for data stored across multiple locations, which performs similarly as the PCA estimator based on the whole dataset. Different assumptions of the distributed sub-datasets are also investigated, such as independent cross-sectional datasets [31], independent sources/studies [2], [5], network meta-analysis [32], high-dimensional correlated data [17], data with responses/predictors having multiple measurements across different experiments [14], and so forth.

One goal of divide-and-conquer is to reduce computational cost via parallel computing across the sub-datasets, where one learner has access to all the sub-datasets, in contrast to each learner having access to one sub-dataset in our framework. In addition, divide-and-conquer methods assume that the underlying relationship between the response and the predictors for each sub-dataset is the same, thus combining results from all the sub-datasets is reasonable. However, the datasets in distributed storage may have potential subgroups, rather than the seemingly unified one label. For example, a dataset from one single study may be stored in distributed servers, and some servers may have data damaged or altered due to systematic problems or hacker attacks; an insurance claim dataset may consist of sub-datasets from different states, where sensitive variables (e.g., age, gender) are different according to different state laws; for data integration from different sources, data analysts are faced with issues of verifying data authenticity and relevance; in meta analysis and clinical trials, exploring or test heterogeneity is of great importance for statistical reproducible inference. In such cases, the underlying relationship between the response and the variables may be different across all sub-datasets. Identifying potential clusterings of the subs-datasets is of great importance for bias reduction and robust/efficient modeling. For divide-and-conquer methods, meta-clustering can be applied to analyze either if there are potential cluster structures on the whole dataset. If there are some cluster structures, then the random splitting in divide-and-conquer may lead to modeling bias.

The remainder of the paper is outlined below. In Section II, we describe the problem and propose the meta clustering method. In Section III-B, we present theoretical properties of the proposed method. In Section IV, we demonstrated a potential use of the method in fairness learning scenarios. In Sections V and VI, we show the performance of our method through more experimental studies. The proofs are included in the Appendix.

II. Problem

Suppose the dataset $\mathcal{D} := \{\mathcal{D}_i\}_{i=1}^L$ is the union of L sub-datasets, where the subscript $i = 1, ..., L$ represents a natural label within the dataset $\mathcal{D}$. For example, $\mathcal{D}_i$ can represent the sub-dataset stored in the i-th location/server, the sub-dataset from the i-th study in a meta-analysis, or the sub-dataset from the i-th 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 $\mathcal{M}_i$. We distinguish these two model classes mainly for technical convenience. Briefly speaking, we will assume that a parametric model (e.g., a linear regression model) has a better rate of convergence than a nonparametric one (e.g., a decision tree), and the latter is consistent in estimation. More detailed assumptions are included in the Appendix. In the degenerate case where there is only one learner, i.e., $l_1 = l_2 = \cdots = l_L$, the collaborative learning setup can be potentially used to enhance the computational efficiency. We will often use small letters to denote observed data and capital letters to denote random variables.

Suppose the sub-dataset $\mathcal{D}_i$ consists of n_i independent data points, denoted as $\mathcal{D}_i = \{(y_{i,j}, \boldsymbol{x}_{i,j}) : y_{i,j} \in \mathbb{R}, \mathbf{x}_{i,j} \in \mathbb{R}^p\}_{j=1}^{n_i}$, from the underlying model

$$Y_i = f_i(\boldsymbol{X}_i) + \varepsilon_i, \tag{1}$$

where $\boldsymbol{X}_1, ..., \boldsymbol{X}_L$ are p-dimensional random variables i.i.d. with cumulative density function $P_{\mathbf{X}}(\cdot)$, and the noise $\varepsilon_i \sim N(0, \sigma_i^2)$ is independent of $\boldsymbol{X}_i$. Moreover, for any $i_1, i_2 \in \{1, ..., L\}$, ε_{i_1} is independent of $\mathbf{X}_{i_2}$. We require that the L sub-datasets consist of the same p predictors.

Let $n := n_1 + ... + n_L$ denote the overall sample size from L sub-datasets. Throughout the paper, we assume that there exist K (fixed but unknown) data generating functions, denoted by $f_i \in \mathcal{F} = \{f^{(1)}, \ldots, f^{(K)}\}$. The learning model for estimating $f^{(i)}$ for any $i = 1, ..., K$ can be either parametric or nonparametric. Denote the Euclidean norm as $||\cdot||$. Define the L_2 norm $\|f\|_2 = \sqrt{\int f(x)^2 P_X(dx)}$ and the L_∞ norm $||f||_\infty = \operatorname{ess\,sup} |f| = \inf\{c \geq 0 : |f(\mathbf{X})| \leq c \text{ a.s.}\}$. Two models $f^{(i)}$ and $f^{(j)}$ are different if $||f^{(i)} - f^{(j)}||_\infty > 0$.

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

III. Method

The intuition of our method is that, for any pair of learners, if their sub-datasets correspond to the same data generating function, then their estimated functions (e.g. by cross validation within each sub-dataset) should perform similarly when applied to each other's sub-dataset. We propose the following three-step method where learners communicate with each other on their estimated regression functions. The method may be used as a pre-screening step for identifying relevant collaborators for downstream collaborative learning.

A. Algorithmic Description

We propose the following **S**elect-**E**xchange-**C**luster (SEC) algorithm.

Step 1 [Select]: Each learner uses its own sub-dataset to learn a model from a set of candidate methods $\mathcal{M}_i$. Suppose that each learner 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 $n_{i,1} = n_{i,2} = n_i/2$ where we assume n_i is even 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 $\hat{f}_{n_{i,1}}$. For learner l_i, denote the best method δ_i as the one that minimizes the mean squared error (MSE) on the test set $\mathcal{D}_{i,2}$, i.e.,

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

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 $\hat{f}_{n_i}$ and its fitted mean squared error as

$$\hat{e}_i := \sum_{(y,\boldsymbol{x})\in\mathcal{D}_i} \big(y - \hat{f}_{n_i}(\boldsymbol{x})\big)^2 / n_i.$$

To summarize, for each learner l_i, we have the non-shared data $\mathcal{D}_i$ and the sharable information $\{\delta_i, \hat{f}_{n_i}, \hat{e}_i\}$.
Step 2 [Exchange]: For any two learners, they exchange the sharable information $\{\delta_i, \hat{f}_{n_i}, \hat{e}_i\}$. In particular, denote v_{ij} as the dissimilarity between any two learners (l_i, l_j) with $i \neq j$. We apply the i-th learner's best estimator $\hat{f}_{n_i}$ to the j-th learner's dataset $\mathcal{D}_j$ and obtain its prediction loss

$$\hat{e}_{i\to j} := \frac{1}{n_j} \sum_{(y,\boldsymbol{x})\in\mathcal{D}_j} \big(y - \hat{f}_{n_i}(\boldsymbol{x})\big)^2,$$

where the subscript $i \to j$ denotes the information flow from l_i to l_j. Similarly, we apply $\hat{f}_{n_j}$ 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:

$$v_{ij} = |\hat{e}_{i\to j} - \hat{e}_j| + |\hat{e}_{j\to i} - \hat{e}_i|, \tag{3}$$

where we have $v_{ij} = v_{ji}$ 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 matrix S where the (i, j)-th component is $s_{ij} := \exp(-av_{ij})$. Here, a is a tuning parameter for computational convenience when $\min_{i,j} v_{ij}$ is large and thus s_{ij} are negligibly small for all pairs of (i, j). The similarity matrix S is symmetric with $s_{ii} = 1$ for $i = 1, \ldots, L$. Let $P = \{1, ..., L\}$. For a given K, we will find a partition $P := \cup_{i=1}^{K} S_i$ by applying a spectral clustering algorithm to the matrix S and partition the L learners into K groups. For completeness, we summarize the clustering step (Step 3) in Algorithm 1.
When K is unknown, we add a penalty term $K \cdot \lambda_n$ in to the k-means in step 2.d) of Algorithm 1, so that we jointly minimize

$$\sum_{t=1}^{K} \sum_{i,j\in P_t} \frac{1}{2|S_t|} ||\mathbf{u}_{(i)} - \mathbf{u}_{(j)}||^2 + K \cdot \lambda_n \tag{4}$$

over all possible partitions P and a grid of values of K. Here, $\mathbf{u}_{(i)}$ denotes the i-th row of U_* defined in Algorithm 1.

Remark 1 (Spectral clustering). There are different variants of spectral clustering in the literature. We build on the work of [25] due to technical convenience. We will show in Section III-B that our construction of the similarity matrix theoretically guarantees the performance of applying the spectral clustering algorithm.

Remark 2 (Selection of K). There are many ways to choose the penalty term in parametric models [6]. In the context we consider, picking an appropriate penalty term may be difficult because the rates of convergence of nonparametric methods in Step 1 may not be known. Here we suggest use the gap statistic [29] that looks for the elbow point in the curve of the sum of within-cluster mean-squared errors (i.e. the first term in (4) against different K's.

Algorithm 1 Pseudocode for the Step 3 of SEC algorithm

Input: Number of learners L, learners/datasets $\{\mathcal{D}_i\}_{i=1}^L$, number of clusters K (optional).
Output: K clusters of the learners $\{S_i\}_{i=1}^K$ as represented by a partition of the set $\{1, 2, ..., L\}$, learner's cluster labels $c_i, i = 1, \ldots, L$, estimates of each underlying regression function $\hat{f}^{(j)}$, $j = 1, ..., K$, and K itself (if not given as in put).

1) Calculate the similarity matrix $S \in \mathbb{R}_+^{L\times L}$, where each $s_{ij} = \exp(-av_{ij})$ and v_{ij} is given by (3).
2) If k is given, conduct the spectral clustering:
   a) Calculate the Laplacian L of S: $L = D^{-1/2}SD^{-1/2}$, where D is a diagonal matrix with $D_{ii} = \sum_{j=1}^L s_{ij}$.
   b) Compute the K largest eigenvectors of L, $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K$. Denote $U = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K] \in \mathbb{R}^{L\times K}$.
   c) Standardize each row of U to have unit ℓ_2 norm. Denote the standardized matrix as U_*.
   d) Apply *k*-means clustering to the rows of U_* into k clusters, and record the labels $c_i, i = 1, \ldots, L$.
3) Else:
   a) Record the eigenvalues of S sorted from small to large, and apply the Gap statistics to determine the K.
   b) Go back to 2).
4) Output the estimates.

Remark 3 (Future prediction). Though prediction is not the main focus of this paper, we suggest the procedure below to perform prediction once we obtain the clustering results from SEC. For any particular learner i, suppose that it belongs to cluster S_k. Other learners in the same cluster transmit their fitted model $\hat{f}_{n_j}$ to learner i. To make predictions for an incoming new observation $\mathbf{x}$, we use

$$\sum_{i\in P_k} \frac{n_i}{\sum_{i\in S_k} n_i} \hat{f}_{n_i}(\mathbf{x}) \tag{5}$$

as the prediction of the response. Note that the above prediction does not require direct sharing of datasets among learners. We emphasize that the obtained clustering results may also be used for more sophisticated downstream collaborative learning methods, where a learner only interacts with others in the same cluster.

B. Theoretical Properties

The following theorem shows that SEC can consistently identify the correct clusters for a large sample size.

Theorem 1. *Under assumptions in the Appendix, the labels $c_1, \ldots, c_L$ 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 $n \to \infty$.*

IV. Data Fairness

The proposed method can be applied to the data fairness problem. Fairness has been an enduring societal issue across different areas. Recently in statistics and machine learning, fairness has gained its attention in data itself and in machine learning. Biases in data collection and measurement, and methods/algorithms based on these data will not address (sometimes even worsen) the inequity for historically disadvantaged groups. Many works have been done to mathematically define fairness, discover unfairness, and algorithms are proposed to achieve fairness. For example, [8] treated individual fairness as classifying similar individuals similarly, where the individual similarity is captured by hypothetical task-specific metrics. [16] proposed a criterion, equal opportunity (or equalized odds), with respect to a particular sensitive variable, and demonstrated a way to adjust any predictor to remove such discrimination. [33] devised positive rate disparity and proposed a method that avoids disparities in mistreatment and treatment simultaneously. [30] compared differences among 20 fairness definitions for classification problems, explaining the rationale of

each definition and demonstrating each with a single case-study. For more detailed definitions of fairness, interested readers are referred to [13], [30] and the references therein. Based on the maximum likelihood principle, [19] proposed a prejudice remover regularizer (by considering the mutual information between the response and the sensitive variable) to any probabilistic models for classification. [34] proposed a classification algorithm that achieves both individual fairness and group fairness. [20] studied penalized linear regression with fairness constraints (i.e., the coefficient of determination of the sensitive variables over the predicted values using non-sensitive variables), which can be reduced to a convex optimization problem that has an exact solution.

We consider a linear regression setting where the sensitive variable is independent from 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: $Y = X_1 + 2X_2 - 2X_3 + 2X_4 + cS_i + \epsilon$, where $\mathbf{X} \sim N(0, I_4)$ is the non-sensitive variable, $\epsilon \sim N(0, 1)$ is the random noise, and S_i is the sensitive variable. Here we consider different scales of the coefficient of the sensitive variable, $c \in \{0.5, 1, 2, 3, 4, 5, 6, 7\}$. The sensitive variable S_i is generated from a standard normal $N(0, 1)$ distribution and is set to be fixed for each given i. This setting of a fixed value as a sensitive variable is sensible and reasonable in practice for the data fairness problem when there are multiple measures for the same subject (each subject is a natural learner). For example, if $\mathcal{D}$ is a longitudinal data and each sub-dataset represents a person, then the subject-specific sensitive variable (gender, race, age, home location, etc.) is the same for each person. We set S_i as continuous to add more generality, since categorical variables can be treated as continuous by introducing dummy variables in linear regression. We split the dataset into training set of 30 learners (e.g., $\{\mathcal{D}_i\}_{i=1}^{30}$) and test set of 20 learners (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 reason behind such splitting is that 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 50 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 is to directly build a linear regression model on the training set using the sensitive variable, i.e., without considering fairness constraints, which has the best predictive performance. The fairness method is to build a linear regression model on the training set without using the sensitive variable, since the sensitive variable is not allowed to be used in the modeling procedure (sometimes even not available in practice). 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 test set $\{\mathcal{D}_i^1\}_{i=31}^{50}$ to cluster the 20 learners into the clusters identified in the training set.[1] In the SEC algorithm, for simplicity, each learner l_i considers two candidate modeling methods: Random Forest [4] (RF) and linear regression (LR), i.e., $\mathcal{M}_i = \{\text{RF}, \text{LR}\}$, based on which the similarity matrix is constructed.

For the validation set, we evaluate the predictive performances of the models by the mean square error (MSE), which are presented in Table I. As shown in the table, when the importance (its coefficient c) of the sensitive variable is high, SEC-Fairness overall reduces the MSE of Fairness by about 50%. One possible reason is as following. The linear function between Y and the variables $\{X_1, ..., X_4\}$ actually only differs in the intercept per learner. The similarity between two learners as in the SEC algorithm will be small if the different of their sensitive variables $|S_i - S_j|$ is large. It is then more likely that SEC divides those with similar values of the sensitive variable into the same cluster. We still believe that the SEC-fairness method satisfies the fairness constraint since it does not utilize the sensitive variable at all. In addition, the sensitive variable is independent from the non-sensitive variables.

The oracle method is very stable in MSE (around 1) over different values of c, which is under

[1]To measure the similarity between a learner l_i and a group, we use the sum of the similarity of l_i with each individual learner in that group. Then the leaner l_i belongs to a group if its similarity with the group is larger than its similarity with any other group.

	SEC-Fairness		Fairness	Oracle
c	MSE	$\hat{K}$	MSE	MSE
0.5	1.18 (0.01)	2.6 (0.49)	1.34 (0.01)	1.05 (0.01)
1	1.45 (0.04)	2.72 (0.57)	2.01 (0.05)	1.04 (0.01)
2	1.98 (0.07)	2.66 (0.48)	4.31 (0.11)	1.00 (0.01)
3	2.90 (0.12)	2.92 (0.27)	8.65 (0.25)	1.10 (0.01)
4	12.18 (0.91)	2.18 (0.88)	21.01 (0.79)	0.93 (0.01)
5	8.94 (0.84)	2.98 (0.73)	23.70 (1.09)	0.96 (0.01)
6	13.51 (0.76)	2.58 (0.49)	42.36 (1.32)	0.98 (0.01)
7	22.71 (2.06)	2.46 (0.73)	45.42 (1.76)	1.00 (0.00)

TABLE I: Predictive performances of the three methods for the data fairness example. 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 50 replications.

expectation. When c is large, though performing better than Fairness, SEC-Fairness performs not too well if compared to the oracle method. One reason can be seen from the estimated number of clusters $\hat{K}$, which is in the interval $[2, 3]$. In this example, we select $\hat{K}$ by the gap statistic, with the goal of minimizing the with-in group distance. But here there is no such a true value of K (or in some sense we can treat the true K as 30 for the training set) since every learner/sub-dataset has a unique sensitive value and can be treated as a cluster itself. If we specify a large K before applying the SEC algorithm, the performance of SEC-Fairness should be better in this example. More generally, if the learner focuses more on improving prediction performance rather than finding potential groupings, we suggest to manually set a large K before applying the SEC algorithm, or try a grid of values of K and pick the one with the best predictive performance.

V. Simulations

In this section, we present three simulation settings. Each example in the simulations is repeated 50 times. Theoretically speaking, the data are not needed to be standardized because only the functional relation between Y and $\boldsymbol{X}$ matters. So one cluster may contain two datasets of which the responses or predictors are not in the same scale or range. However, the nonparametric method usually requires a compact support, which may cause some computational issues. Throughout the experiments, we preprocess $\boldsymbol{X}$ and Y in each learner by standardization.

A. Simulation 1: Clustering accuracy

This example is to demonstrate that the clustering accuracy of our method. A clustering result is called accurate if both the number of clusters is correctly identified and each learner's label matches the underlying truth (up to a permutation).

Suppose that there are 20 learners, each with 50 observations of (y, x), where x follows a standard Gaussian distribution with dimension $p = 5, 10, 20$. The data of the first 10 learners are generated from the underlying model $y = f_1(x) + \varepsilon_1 = \beta_1^{\mathrm{T}} x + \varepsilon_1$, where $\varepsilon_1 \sim N(0, \sigma^2)$, and $\beta_1 \in \mathbb{R}^p$. The data of the remaining 10 learners are generated from $y = f_2(x) + \varepsilon_2 = \beta_2^{\mathrm{T}} x + \varepsilon_2$, with $\varepsilon_2 \sim N(0, \sigma^2)$, and $\beta_2 \in \mathbb{R}^p$. We randomly generate β_1 and β_2 from the standard Gaussian distribution (both of β_1 and β_2 are set as fixed in each replicated experiment such that $\beta_1 \neq \beta_2$). The signal-to-noise ratio (SNR) is defined by $\mathbb{E}(\|\beta\|^2)/\mathbb{E}(\varepsilon^2)$, which reduces to p^2/σ^2 in this case. We set the levels of SNR to be one of the following: $2^0, \ldots, 2^7$. In the SEC algorithm, let each learner consider two candidate methods: LASSO [28] (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 learners. The averaged clustering accuracy over 50 replications is presented in Figure 2. It can be seen that the clustering accuracy increases as the SNR increases. Also, for a fixed SNR, smaller p tends to have better clustering accuracy. This is mainly because a more parsimonious model suffers from more estimation variance given the same amount of data. We also see that, for fixed p, the accuracy curve tends to be flat when SNR is larger than 2^5, showing the robustness of the SEC algorithm against low noise levels. We also present the result of a replication of the simulation with $p = 5$ and SNR$= 2^4$, with clustering accuracy being 100. The gap statistics used to choose the number of clusters K is plotted in Figure 3a and the eigenvectors in the spectral clustering algorithm is plotted in Figure 3b.

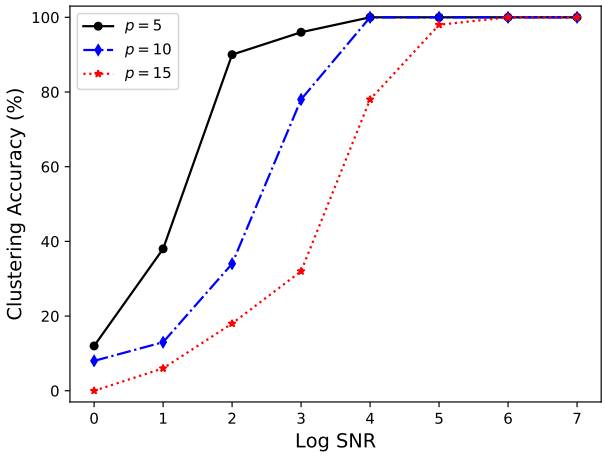
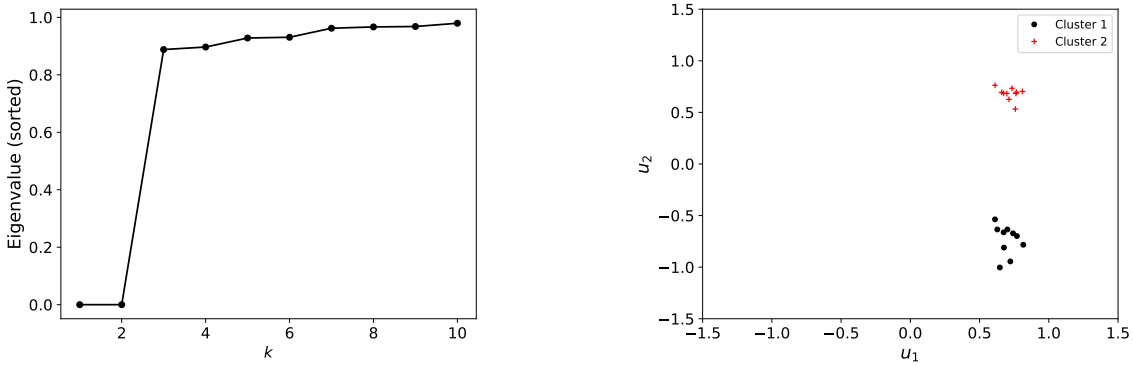


Fig. 2: Clustering accuracy of the SEC algorithm for Simulation 1.



(a) Eigenvalues (sorted) that were used to determine the most appropriate number of clusters.

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

Fig. 3: An illustration of the clustering results, based on a realization with $p = 5$, SNR=4.

B. Simulation 2: robustness against candidate models in the Cross Validation

In this example, we demonstrate that our method is robust against candidate models in the cross validation part of the Select step. Suppose that 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 [3], [11]. The sub-datasets of the first ten learners are generated from

$$y = f_1(X) + \varepsilon_1 = \sqrt{X_1^2 + (X_2X_3 - \frac{1}{X_2X_4})^2} + \varepsilon_1,$$

and the sub-datasets of remaining ten learners are generated from

$$y = f_2(X) + \varepsilon_2 = \arctan(X_2X_3 - \frac{1}{X_2X_4})/X_1 + \varepsilon_2,$$

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 $\varepsilon_1, \varepsilon_2 \sim N(0, 0.01)$ are independent. The remaining 496 predictors $\{X_5, ..., X_{500}\}$ follow a standard multivariate gaussian distribution $N(0, I_{496})$.

For each learner, we consider the pool of candidate models: Random Forest, K nearest neighbors, Support Vector Regression [7], Artificial Neural Network, Gradient Boosting [12], LASSO, Least Angler Regression [9](LARS), Elastic Net [36] (EN), Ridge Regression (Ridge), i.e.,

$$M_i = \{\text{RF, KNN, SVR, NN, GB, Lasso, Lars, EN, Ridge}\}$$

for $i = 1, ..., 20$. 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 II. 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.

To evaluate whether the SEC algorithm improves prediction accuracy, we focus on the first leaner l_1 in this example. We generate a test set $\mathcal{D}_{test}$ with size $n_{test} = 100$ that is generated from the model $y = f_1(X) + \varepsilon_1$. We treat this test set as a test set for the first learner l_1. We consider the following two modeling methods: No collaboration and Collaboration. The "Collaboration" method firsts applies the SEC algorithm and identify those learners that are from the same cluster as l_1. Then we obtain the prediction for the test set $\mathcal{D}_{test}$ based on the simple average of the estimated predictors from those learners, as described in Equation (5). 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 $\mathcal{D}_{test}$ to make predictions. The mean squared error of the predictions by the above two methods are also shown in Table II. The prediction accuracy of the two methods are also stable across different choices of $\mathcal{M}_i$, in terms of both the size of $\mathcal{M}_i$ and the types of methods in $\mathcal{M}_i$. When the number of candidate models increases in $\mathcal{M}_i$, the prediction accuracy of the "Collaboration" method increases. This is because more choices of candidate models in the Cross-validation part of the Select Step enables us to learn better the functional relation between the response and the predictors, so that the similarity matrix can better capture the true underlying clusters.

			Collaboration	No collaboration
$\lvert\mathcal{M}\rvert_i$	Accuracy	$\hat{K}$	MSE	MSE
1	66.0	2	0.100(0.056)	0.133(0.038)
3	74.0	2	0.095(0.053)	0.131(0.042)
5	58.0	2	0.087(0.049)	0.125(0.044)
7	70.0	2	0.096(0.056)	0.134(0.051)
9	64.0	2	0.060(0.018)	0.112(0.034)

TABLE II: Prediction performance of our method vs. classical methods for the robustness against candidate models example. The column "Accuracy" is the clustering accuracy of the SEC algorithm. The value in the parenthesis is the standard error of the averaged MSE over 50 replications, and $\hat{K}$ denotes the estimated number of clusters.

VI. Real Data Applications

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

A. Application 1: more accurate prediction in CT Image Data

We investigate the CT Image Data [15] 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 one thorax scan and at most one neck scan. The response variable is the relative location of the CT slice on the axial axis. 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 learners, each representing a CT scan. Our goal is to find whether there exists any potential clustering structure (and its 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 with the data fairness example, for each of the 33 leaners in the test set, we divide the sub-dataset into two sets of equal size.

For the training set, we consider two methods: clustering and no clustering. The "no clustering" method directly trains a Random Forest model on the training set. The "clustering" method first applies the SEC algorithm to cluster the learner in the training set, with $\mathcal{M}_i = \{\text{RF}, \text{LASSO}\}$ for $i = 1, ..., 64$. Then it trains a Random Forest model separately in each identified clusters using the pooled data. For the validation set, the "no clustering" method directly applies the trained random forest model and obtain the mean squared error. The "clustering" method first finds the similar learners for each learner in the validation set. Then it applies the trained random forest model corresponding to that cluster to the leaner.

We repeat the data splitting 50 times and summarize the results in Table III. It can be seen that the "clustering" method significantly outperforms that of the "no clustering" method. A right-sided paired *t-test* of the difference in MSE between "clustering" and "no clustering" produces a p-value 1.77e-14. To summarize, the SEC algorithm can be applied to datasets that can be naturally divided into sub-datasets, to improve prediction accuracy.

We also tried to look for more scientific understanding of the identified clusters on the training set. However, the dataset does not contain information of the gender of the patient, whether the CT scan is from thorax or neck. Indeed, this type of patient information is not available to the authors of the paper [15]. But again, through this example, even if we are not able to assess the private information, we still are able to improve the predictive performance significantly.

	Clustering	No Clustering
MSE	95.15 (6.38)	150.31 (4.59)
$\hat{K}$	2 (3 times) and 3 (47 times)	N/A

TABLE III: Results for the CT Image Data. The value in the parenthesis is the standard error of the averaged MSE over 50 replications, and $\hat{K}$ denotes the estimated clusters.

B. Application 2: robust learning in Electrical Grid Stability Data against adversaries

This example is to demonstrate the performance of the SEC algorithm when the data is under adversarial attacks. The Electrical Grid Stability Data [1] consists of 10000 observations and 14 variables. Among the 14 variables, two variables describe the system stability: one is categorical (stable/unstable) and the other is continuous (where a positive value means a linearly unstable system). We use the continuous variable as the response. The other 12 variables are the input variables 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 that the data is stored in 50 servers and some servers get attacked by hackers. Each time a sub-dataset is "attacked", for simplicity

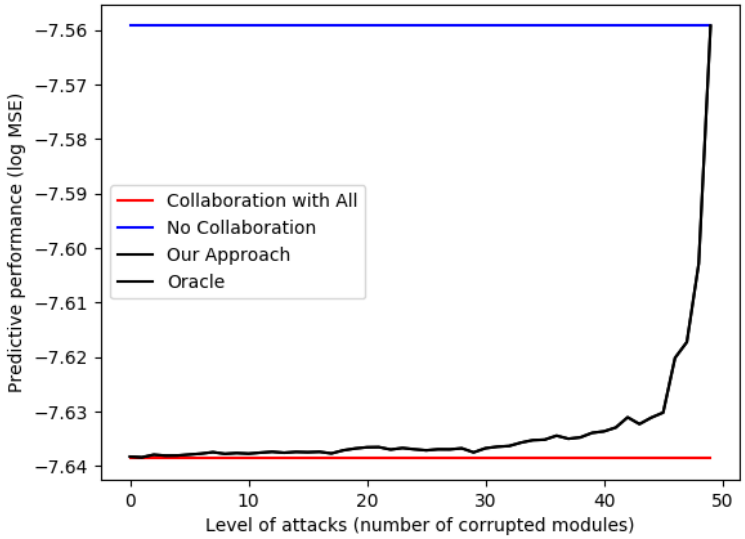


Fig. 4: Prediction error (evaluated by MSE) as an increasing function of attack severity.

and illustration, we change the response variable to the opposite number of its original value. We also assume that the 50-th learner knows for sure that his/her dataset is not attacked.

Under potential attacks, we consider four options of the 50-th learner to perform data analysis, denoted as Collaboration with all, No collaboration, Collaboration with non-attacked, and Oracle. The "Collaboration with all" option ignores the fact that some learners/sub-datasets are attacked and insists to collaborate with all the other leaners. The "No collaboration" option trusts nobody but only himself/herself, and simply just uses the 50-th sub-dataset as a safe option. The "Collaboration with non-attacked" option is to cluster the 50 learners into "attacked" and "intact", and collaborate with those leaners who are classified as intact. The "Oracle" option is that an oracle knows exactly which learners are attacked and collaborates with those intact ones. By collaboration, for simplicity, in this example we allow the learners to share datasets. So once a learner identifies his/her desired collaborators, the leaner just pool the datasets together and fit a linear regression model.

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. As shown by the results, our method accurately clusters all the intact learners, and thus the performance curve of "Collaboration with Non-attacked" overlaps with that of "Oracle". We also see that the predictive performance of "Collaboration with Non-attacked" decreases when the level of attack (the number of the attacked learners) increases. In particular, the decrement becomes very sharp when the number of attacked learners is greater than 45. One reason may be that the linear model based on the information of one sub-dataset (with sample size 160 and 12 predictors) or two is enough to capture the underlying relationship. Indeed, the scale of the of MSE is very small (around 0.0005). So collaborating with more than 5 intact leaners may not improve the prediction accuracy much in comparison to collaborating with only 2 intact learners.

REFERENCES

[1] V. Arzamasov, K. Böhm, and P. Jochem. Towards concise models of grid stability. In *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–6, Oct 2018.

[2] Heather Battey, Jianqing Fan, Han Liu, Junwei Lu, and Ziwei Zhu. Distributed estimation and inference with statistical guarantees. *arXiv preprint arXiv:1509.05457*, 2015.

[3] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[4] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

[5] Brian Claggett, Minge Xie, and Lu Tian. Meta-analysis with fixed, unknown, study-specific parameters. *Journal of the American Statistical Association*, 109(508):1660–1671, 2014.

[6] Jie Ding, Vahid Tarokh, and Yuhong Yang. Model selection techniques–an overview. *IEEE Signal Process. Mag.*, 35(6):16–34, 2018.

[7] Harris Drucker, Christopher JC Burges, Linda Kaufman, Alex J Smola, and Vladimir Vapnik. Support vector regression machines. In *Advances in neural information processing systems*, pages 155–161, 1997.

[8] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS ?12, page 214?226, New York, NY, USA, 2012. Association for Computing Machinery.

[9] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. *Ann. Stat.*, 32(2):407–499, 2004.

[10] Jianqing Fan, Dong Wang, Kaizheng Wang, and Ziwei Zhu. Distributed estimation of principal eigenspaces. *Annals of statistics*, 47(6):3009, 2019.

[11] Jerome H Friedman. Multivariate adaptive regression splines. *Ann. Stat.*, pages 1–67, 1991.

[12] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Ann. Statist.*, 29(5):1189–1232, 10 2001.

[13] Pratik Gajane and Mykola Pechenizkiy. On formalizing fairness in prediction with machine learning. *arXiv preprint arXiv:1710.03184*, 2017.

[14] Xin Gao and Raymond J Carroll. Data integration with high dimensionality. *Biometrika*, 104(2):251–272, 2017.

[15] Franz Graf, Hans-Peter Kriegel, Matthias Schubert, Sebastian Pölsterl, and Alexander Cavallaro. 2d image registration in ct images using radial image descriptors. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 607–614. Springer, 2011.

[16] Moritz Hardt, Eric Price, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3315–3323. Curran Associates, Inc., 2016.

[17] Emily C Hector and Peter X-K Song. A distributed and integrated method of moments for high-dimensional correlated data analysis. *Journal of the American Statistical Association*, pages 1–14, 2020.

[18] Shane T Jensen, Guang Chen, Christian J Stoeckert Jr, et al. Bayesian variable selection and data integration for biological regulatory networks. *The Annals of Applied Statistics*, 1(2):612–633, 2007.

[19] Toshihiro Kamishima, Shotaro Akaho, and Jun Sakuma. Fairness-aware learning through regularization approach. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 643–650. IEEE, 2011.

[20] Junpei Komiyama, Akiko Takeda, Junya Honda, and Hajime Shimao. Nonconvex optimization for regression with fairness constraints. In *International conference on machine learning*, pages 2737–2746, 2018.

[21] Jakub Konecny, H Brendan McMahan, Felix X Yu, Peter Richtarik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[22] Quefeng Li and Lexin Li. Integrative linear discriminant analysis with guaranteed error rate improvement. *Biometrika*, 105(4):917–930, 2018.

[23] Lester Mackey, Ameet Talwalkar, and Michael I. Jordan. Distributed matrix completion and robust factorization. *Journal of Machine Learning Research*, 16(28):913–960, 2015.

[24] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

[25] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

[26] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1310–1321. ACM, 2015.

[27] Lu Tang and Peter XK Song. Fused lasso approach in regression coefficients clustering: learning parameter heterogeneity in data integration. *Journal of Machine Learning Research*, 17(1):3915–3937, 2016.

[28] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[29] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[30] Sahil Verma and Julia Rubin. Fairness definitions explained. In *Proceedings of the International Workshop on Software Fairness*, FairWare 18, page 17, New York, NY, USA, 2018. Association for Computing Machinery.

[31] Minge Xie, Kesar Singh, and William E. Strawderman. Confidence distributions and a unifying framework for meta-analysis. *Journal of the American Statistical Association*, 106(493):320–333, 2011.

[32] Guang Yang, Dungang Liu, Regina Y. Liu, Minge Xie, and David C. Hoaglin. Efficient network meta-analysis: A confidence distribution approach. *Statistical Methodology*, 20:105 – 125, 2014. Re-sampling and Contemporary Inference: A tribute to Kesar Singh.

[33] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th international conference on world wide web*, pages 1171–1180, 2017.

[34] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International Conference on Machine Learning*, pages 325–333, 2013.

[35] Yuchen Zhang, John Duchi, and Martin Wainwright. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *Journal of Machine Learning Research*, 16(102):3299–3340, 2015.

[36] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.