Distributed Collaborative Feature Selection Based on Intermediate Representation

Xiucai Ye, Hongmin Li, Akira Imakura and Tetsuya Sakurai
Department of Computer Science, University of Tsukuba
yexiucai@cs.tsukuba.ac.jp, li.hongmin.xa@alumni.tsukuba.ac.jp, imakura@cs.tsukuba.ac.jp,
sakurai@cs.tsukuba.ac.jp

Abstract

Feature selection is an efficient dimensionality reduction technique for artificial intelligence and machine learning. Many feature selection methods learn the data structure to select the most discriminative features for distinguishing different classes. However, the data is sometimes distributed in multiple parties and sharing the original data is difficult due to the privacy requirement. As a result, the data in one party may be lack of useful information to learn the most discriminative features. In this paper, we propose a novel distributed method which allows collaborative feature selection for multiple parties without revealing their original data. In the proposed method, each party finds the intermediate representations from the original data, and shares the intermediate representations for collaborative feature selection. Based on the shared intermediate representations, the original data from multiple parties are transformed to the same low dimensional space. The feature ranking of the original data is learned by imposing row sparsity on the transformation matrix simultaneously. Experimental results on real-world datasets demonstrate the effectiveness of the proposed method.

1 Introduction

The dimensionality of data is often very high in many real-world applications [Jain and Zongker, 1997], making great challenges such as the curse of dimensionality, high computation and storage cost. There are mainly two distinct ways for dimensionality reduction to tackle these difficulties: feature extraction and feature selection. Feature extraction provides data transformation to form a low dimensional space, while feature selection aims to extract the amount of important features to represent the original data [Guyon and Elisseeff, 1997; Imakura *et al.*, 2019]. In the applications that need to retain the original representations of data variables, e.g, learning the risk factors of cancer, feature selection is preferred.

The feature selection methods can be classified into three main types: filter [He *et al.*, 2006], wrapper [Maldonado and Weber, 2009], and embedded methods [Hou *et al.*, 2011; Nie *et al.*, 2010; Ye *et al.*, 2016b]. The filter methods are usually computationally simple, but ignore the interactions among features, which may lead to undesired classification result. The wrapper methods detect the possible interactions among features by searching the feature subsets in a learning model, which provide better results than the filter methods [Saeys *et al.*, 2008]. However, the wrapper methods are usually computationally expensive. The embedded methods consider the interactions among features while having lower computational complexity than the wrapper methods, which incorporate feature selection as a part of data training process to search for the optimal subset of features [Ye *et al.*, 2016a].

Similar to most machine learning techniques, in the general setting of feature selection, a large number of data should be available to train a model for good generalization ability. However, in some applications, the data is distributed in multiple parties. The data in one party may be lack of some useful information to learn the most discriminative features. Thus, sharing data with multiple parties for collaborative feature selection can help to learn more useful information and solve the problem of information deficiency in one party. For example, consider learning the risk factors of cancer in the hospital. The risk factors learned in one hospital are limited to the data it holds. If several hospitals share their data for collaborate feature selection, a wealth of information contained in the shared data could mutually benefit all hospitals to learn the most important risk factors.

However, data sharing may be difficult in some organizations due to the limitations such as privacy requirement. Thus, an emerging challenge for feature selection is how to learn the data from multiple parties without revealing their original data. Although the current researches mainly focus on classification or clustering for privacy-aware distributed data setting [Chaudhuri *et al.*, 2011], some studies have been devoted to feature selection by considering the privacy preserving. Due to the simplicity of filter methods, most researchers consider privacy-aware feature selection based on the filter methods. Banerjee and Chakravarty [Banerjee and Chakravarty, 2011] propose a distributed filter based feature selection method by using the virtual dimension reduction technique with a secure sum protocol, which allows privacy-aware feature selection for multiple parties. Yang and Li [Yang and Li, 2014] design a private filter based feature selection method by adding noise as output perturbation according to the sensitivity analysis, but in centralized architecture. Sheikhalishahi and Martinelli [Sheikhalishahi and Fabio, 2017] consider the trade-off between feature utility and privacy score to remove the irrelevant features, and collaborative data classification is performed on the data with the remaining features from multiple parties. Including the filter methods, wrapper based feature selection methods also have been proposed by combining with the anonymization techniques [Jafer *et al.*, 2014]. However, these methods are in centralized architecture and have heavy computational cost.

In this paper, we propose a novel distributed method for embedded feature selection, which allows collaborative feature selection for multiple parties without revealing their original data. Existing studies have addressed privacy preserving for the filter and wrapper feature selection methods, to the best of our knowledge, there is no existing study addresses privacy preserving for the embedded feature selection methods. The contributions of the proposed method are summarized as follows.

- Different from the above related works, the proposed method does not rely on any security protocols. The proposed method performs collaborative feature selection based on the intermediate representations. Instead of the original data, each party shares the intermediate representations for feature learning.
- Embedded feature selection is performed in each party by learning the intermediate representations from multiple parties, which can solve the information deficiency in the local party.
- The proposed method is flexible and extendable, since many embedded feature selection methods can be incorporated to perform distributed collaborative feature selection with privacy preserving.
- Experimental results show that the proposed method can help a single party to select the important features and improve the classification result through collaborative feature selection.

2 Related Methods

2.1 Embedded Feature Selection

We introduce a framework of unsupervised embedded feature selection methods [Hou *et al.*, 2011] and incorporate it for collaborative feature selection, while the supervised case also can be incorporated in our method. Consider that n original data $X = [x_1, ..., x_n] \in R^{m\times n}$ are embedded in a low dimensional space by a transformation matrix $W \in R^{m\times q}$, where m and q are the dimensionalities of the original and embedded data. Let $Y = [y_1, ..., y_n] \in R^{q\times n}$ denote the embedded data matrix of X, where y_i is corresponding to x_i. By preserving the local data structure of X in Y, the objective function is formulated as

$$\min_{W,Y} \|Y - W^T X\|_F^2 + \alpha\|W\|_{2,1} + \beta tr(YLY^T), \tag{1}$$

where α and β are two balanced parameters; the term $\|W\|_{2,1}$ is calculated as $\|W\|_{2,1} = \sum_{i=1}^n \sqrt{\sum_{j=1}^m w_{i,j}^2}$, which is to ensure that W is sparse in rows; $tr(Y^T LY)$ is a promoting regularization term to preserve the local data structure in Y.

Many methods can be used to preserve the local data structure, such as LPP [He and Niyogi, 2004] and LLE [Roweis and Saul, 2000]. We use LPP to preserve the data similarity of X in Y.

$$\min_Y \sum\nolimits_{i,j=1}^n \|y_i - y_j\|_2^2 s_{ij}, \tag{2}$$

where s_{ij} is the pairwise similarity between x_i and x_j. Based on the k-nearest neighbor graph, s_{ij} is calculated as

$$s_{ij} = \begin{cases} \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2}), & x_i \text{ and } x_j \text{ are connected,} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Let $L = D - S$ be the Laplacian matrix, where S is the similarity matrix with s_{ij} as its entries, D is the $n\times n$ diagonal matrix with $D_{ii} = \sum_{j=1}^n s_{ij}$ on the diagonal. Then, equation (2) can be equivalently expressed as

$$\min_Y Tr(YLY^T). \tag{4}$$

The optimization problem in equation (1) is solved by updating Y and W alternatively until the objective function converges [Hou *et al.*, 2011]. Let w_i denote the i^{th} row of W. w_i corresponds to the weight of the i^{th} feature, thus, the sparsity constraint on rows makes W suitable for feature selection. The feature weights are ranked according to $\|w_i\|_2$ in descending order and the top rank features are selected.

2.2 Intermediate Representation

Intermediate representation has been proposed as an effective method for collaborative classification [Imakura and Sakurai, 2019]. Consider g parties and let X_i be the original data in party i. Party i calculates the intermediate representation of X_i as $\widetilde{X}_i = B_i X_i \in R^{p\times n}$ by a transformation matrix $B_i \in R^{p\times m}$. X_i, B_i and $\widetilde{X}_i$ are not revealed to other parties. To perform collaborative classification, $\widetilde{X}_i$ $(i = 1, .., g)$ from multiple parties are transformed to the same low dimensional space, where $\widehat{X}_i = M_i \widetilde{X}_i \in R^{q\times n}$, and $M_i \in R^{q\times p}$ is the transformation matrix in party i which satisfies

$$M_i B_i X \approx M_j B_j X, i \neq j. \tag{5}$$

That is, for an original data X, after the two transformations by $M_i B_i$ and $M_j B_j$ in parties i and j, the same data properties are retained, i.e., the two transformations of X tend to the same data point in the low dimensional space. Thus, classification can be performed on $\widehat{X}_i$ $(i = 1, .., g)$ in the low dimensional space.

Without sharing B_i, to find M_i that satisfies equation (5), some sharable data called anchor data are introduced. The anchor data can be the public data or dummy data constructed based on the training data. Let $X^a = [x_1^a, ..., x_r^a] \in R^{m\times r}$ denote the anchor data and $\widetilde{X}_i^a = B_i X^a$ is the intermediate representation of X^a in party i. $\widetilde{X}_i^a$ $(i = 1, .., g)$ are shared to calculate M_i. Set the target $Z^a \in R^{q\times r}$ that satisfies $Z^a \approx \widehat{X}_i^a$ $(i = 1, .., g)$. Z^a can be solved by

$$\min_{M_1,...,M_g,Z^a} \sum_{i=1}^g \|Z^a - M_i \widetilde{X}_i^a\|_F^2. \tag{6}$$

Directly solving equation (6) is difficult. An alternative way is to solve its minimal perturbation problem based on the singular value decomposition (SVD)[Ito and Murota, 2016]. Let

$$[(\widetilde{X}_1^a)^T, ..., (\widetilde{X}_g^a)^T] = [U_1, U_2] \left[\begin{array}{cc} \Sigma_1 & \\ & \Sigma_2 \end{array} \right] \left[\begin{array}{ccc} V_{11}^T & \cdots & V_{g1}^T \\ V_{12}^T & \cdots & V_{g2}^T \end{array} \right] \quad (7)$$

be SVD of the matrix combing $\widetilde{X}_i^a$. Then, Z^a can be set as

$$Z^a = U_1^T. \quad (8)$$

Based on equation (6), M_i can be computed by solving $\min_{M_i} \|Z^a - M_i \widetilde{X}_i^a\|_F^2$, that is

$$M_i = Z^a (\widetilde{X}_i^a)^\dagger. \quad (9)$$

3 Distributed Collaborative Feature Selection

In this section, we propose a distributed method for collaborative feature selection. We first show the steps and formulations of the proposed method and then provide an effective algorithm to solve the problem.

3.1 Steps and Formulations

In the embedded feature selection, the original data X is transformed to the low dimensional data Y by a transformation matrix W. Y is obtained by preserving the local data structure of X and the learned transformation matrix W is used to rank the features. If X is lack of some useful information, W can not learn the most discriminative features. In the proposed method, we consider the collaborative feature selection from multiple parties to solve the information deficiency in one party, meanwhile we consider privacy preserving that each party does not reveal the original data.

Inspired by [Imakura and Sakurai, 2019], we consider transforming the original data from multiple parties by two transformation matrices (e.g., B_i and M_i in party i) to the same low dimensional space, where the condition of equation (5) is satisfied. This can be realized by the following steps.

1) Set the anchor data $X^a \in R^{m \times r}$ and share them for all parties.

2) Apply LPP [He and Niyogi, 2004] to calculate the transformation matrix B_i based on X_i in party i. The intermediate representations of X_i and X^a are calculated as $\widetilde{X}_i = B_i X_i$ and $\widetilde{X}_i^a = B_i X^a$, respectively.

3) The intermediate representations of the anchor data from multiple parties, i.e., $\widetilde{X}_1^a, ..., \widetilde{X}_g^a$, are shared to calculate the transformed low dimensional data Z^a, as shown in equation (8). The transformation matrix M_i can be calculated as equation (9).

4) The transformations of the original data from multiple parties in the same low dimensional space can be calculated as $Z_i = M_i \widetilde{X}_i = M_i B_i X_i$ $(i = 1, .., g)$.

Different from [Imakura and Sakurai, 2019] that aims to perform classification on Z_i $(i = 1, .., g)$ from multiple parties, for the purpose of feature selection, we consider learning the transformation matrix $M_i B_i$ to rank the features in each party. Since B_i is distributed in each party without sharing, in each party we consider updating B_i to minimize the following objective function for feature selection.

$$\min_{B_i} \|Z_i - M_i B_i X_i\|_F^2 + \alpha \|(M_i B_i)^T\|_{2,1} + \beta tr(M_i B_i X_i L (M_i B_i X_i)^T), \quad (10)$$

where α and β are two balanced parameters. Note that the B_i in step **2)** can be seen as the initialization to generate the initial Z_i in step **4)**. After B_i being updated, Z_i can be updated based on $\widetilde{X}_i^a = B_i X^a$ according to equations (7) and (8). Let $W_i = (M_i B_i)^T$, equation (10) can be rewritten as

$$\min_{W_i} \|Z_i - W_i^T X_i\|_F^2 + \alpha \|W_i\|_{2,1} + \beta tr(W_i^T X_i L X_i^T W_i). \quad (11)$$

The term $\|W_i\|_{2,1}$ is to ensure that W_i is sparse in rows and the feature weights in party i are learned based on W_i.

Note that L in the term $tr(W_i^T X_i L X_i^T W_i)$ is calculated as that introduced in Section 2.1. We replace Y by $W_i^T X_i$ since the low dimensional data Z_i is not a variable in the objective function, while Z_i can be updated based on B_i. Z_i is calculated based on $\widetilde{X}_i^a$ $(i = 1, .., g)$, which are the intermediate representation of the anchor data from multiple parties. Thus, some properties of the original data from multiple parties can be preserved by the transform to Z_i. In addition, we also consider preserving the local data structure of X_i by adding the term $tr(W_i^T X_i L X_i^T W_i)$. In contrast to our method, the local feature selection in equation (1) only considers preserving the local data structure of X_i.

3.2 Solutions

We show the process of how to solve the objective function in equation (11).

Denote $\Theta(W_i) = \|Z_i - W_i^T X_i\|_F^2 + \alpha \|W_i\|_{2,1} + \beta tr(W_i^T X_i L X_i^T W_i)$.

$$\frac{\partial \Theta(W_i)}{\partial W_i} = 2 X_i X_i^T W_i - 2 X_i Z_i^t + 2\alpha H_i W_i + 2\beta (X_i L X_i^T) W_i, \quad (12)$$

where $H_i \in R^{m \times m}$ is a diagonal matrix with the i^{th} diagonal element as

$$H_{ii} = \frac{1}{2\|w_i\|_2}. \quad (13)$$

w_i is the i^{th} row of W. By setting $\frac{\partial \Theta(W_i)}{\partial W_i} = 0$, we have

$$W_i = (X_i X_i^T + \alpha H_i + \beta (X_i L X_i^T))^{-1} X_i Z_i^T. \quad (14)$$

Since $W_i = (M_i B_i)^T$, we obtain

$$B_i = M_i^{-1} W_i^T. \quad (15)$$

After B_i being updated, the intermediate representations $\widetilde{X}_i^a = B_i X^a$ can be updated. M_i can be updated based on $\widetilde{X}_i^a$. Thus, Z_i can be updated. Then, B_i can be updated again. We consider alternatively updating W_i and Z_i to optimize the objective function. Algorithm 1 shows the procedure of collaborative feature selection distributed in party i. Algorithm 1 will stop when the objective function tends to a constant or the change is very small.

Algorithm 1 Distributed collaborative feature selection

```
Input: Original data matrix $X_i$; Anchor data matrix$X^a$
Parameter: Balance parameters $\alpha$, $\beta$; Neighborhood size $k$; Dimensionalities of embedding $p$, $q$; Selected feature number $d$
Output: $d$ selected features
 1: Set $t = 0$ and Calculate $L$;
 2: Initialize $B_i^t$ by LPP; Calculate $(\widetilde{X}_i^a)^t = B_i^t X^a$;
 3: Collect $(\widetilde{X}_1^a)^t, ..., (\widetilde{X}_g^a)^t$ from multipler parties; Calculate $(Z^a)^t$ as in equation (8); Calculate $M_i^t$ as in equation (9);
 4: Calculate $Z_i^t = M_i^t B_i^t X_i$;
 5: Initialize $H^t = I_m$ ;
 6: repeat
 7:     Calculate $W_i^t$ as in equation (14);
 8:     Calculate $B_i^{t+1} = (M_i^t)^{-1}(W_i^t)^T$;
 9:     Calculate $(\widetilde{X}_i^a)^{t+1} = B_i^{t+1} X^a$;
10:     Collect $(\widetilde{X}_1^a)^{t+1}, ..., (\widetilde{X}_g^a)^{t+1}$ from other parties;
11:     Calculate $(Z^a)^{t+1}$ as in equation (8);
12:     Calculate $M_i^{t+1}$ as equation (9);
13:     Calculate $Z_i^{t+1} = M_i^{t+1} B_i^{t+1} X_i$
14:     Calculate $H_i^{t+1}$ according to equation (13);
15:     $t = t + 1$;
16: until convergence
17: Sort each feature according to $\|w_i\|_2$ in descending order and select the top $d$ ranked ones.
```

4 Discussions

4.1 Convergence Analysis

We solve the proposed method in an alternative way. Then, we show its converge behavior. From steps 8 and 13 in Algorithm 1, Z_i^{t+1} is calculated based on W_i^t. For the sake of convenience, we denote $Z_i^t = f(W_i^{t-1})$. Denote $\Theta(W_i^t) = \|f(W_i^{t-1}) - (W_i^t)^T X_i\|_F^2 + \alpha\|W_i^t\|_{2,1} + \beta tr((W_i^t)^T X_i L X_i^T W_i^t)$. We can show that the objective function in equation (11) will monotonically decrease in each iteration if $\Theta(W_i^{t+1}) \leq \Theta(W_i^t)$.

Firstly, when $f(W_i^{t-1})$ is fixed, we can prove that

$$\begin{aligned} &\|f(W_i^{t-1}) - (W_i^{t+1})^T X_i\|_F^2 + \alpha\|W_i^{t+1}\|_{2,1} \\ &\quad +\beta tr((W_i^{t+1})^T X_i L X_i^T W_i^{t+1}) \\ &\leq \|f(W_i^{t-1}) - (W_i^t)^T X_i\|_F^2 + \alpha\|W_i^t\|_{2,1} \\ &\quad +\beta tr((W_i^t)^T X_i L X_i^T W_i^t). \end{aligned} \tag{16}$$

We omit the detailed proof process of equation (16). A similar proof process can be found in [Hou *et al.*, 2011]. On the other hand, since $\|f(W_i^t) - (W_i^{t+1})^T X_i\|_F^2 \leq \|f(W_i^{t-1}) - (W_i^{t+1})^T X_i\|_F^2$, the following inequality holds.

$$\begin{aligned} &\|f(W_i^t) - (W_i^{t+1})^T X_i\|_F^2 + \alpha\|W_i^{t+1}\|_{2,1} \\ &\quad +\beta tr((W_i^{t+1})^T X_i L X_i^T W_i^{t+1}) \\ &\leq \|f(W_i^{t-1}) - (W_i^{t+1})^T X_i\|_F^2 + \alpha\|W_i^{t+1}\|_{2,1} \\ &\quad +\beta tr((W_i^{t+1})^T X_i L X_i^T W_i^{t+1}). \end{aligned} \tag{17}$$

From the inequalities (16) and (17), we have $\Theta(W_i^{t+1}) \leq \Theta(W_i^t)$. The objective function has lower bounds, such as zero, thus the above iteration will converge. Empirical results show that the convergence is fast and only several iterations (less than 10 iterations in the experiments) are needed to converge.

4.2 Relations with Other Methods

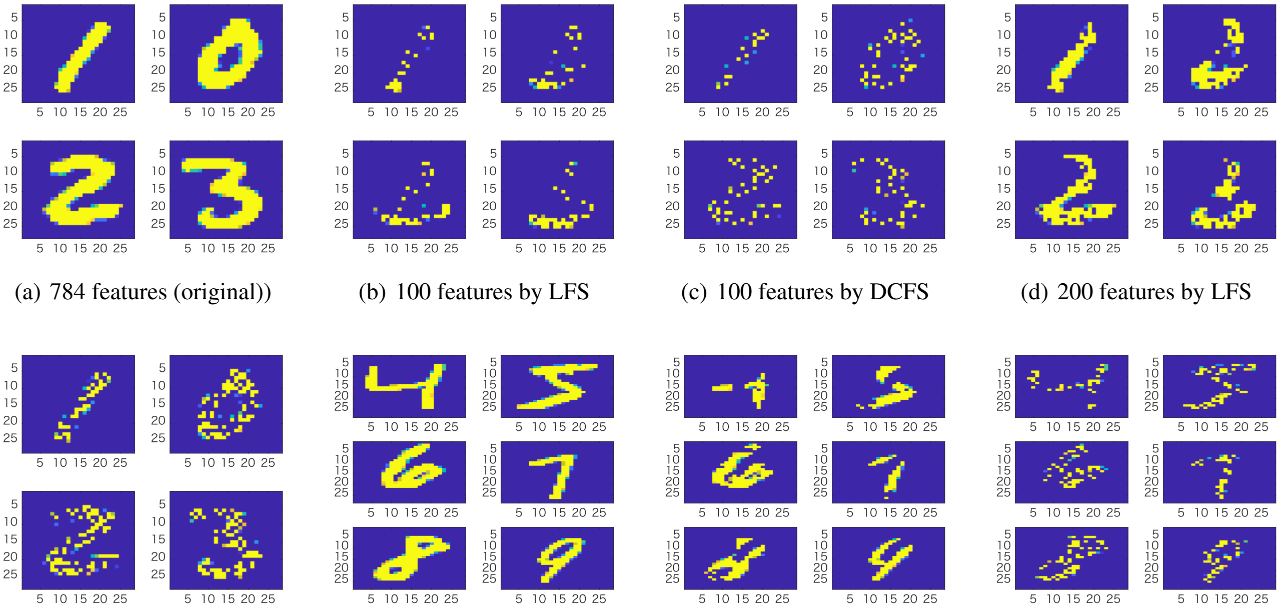
Firstly, considering the privacy preserving and distributed collaboration, the proposed method is related to the methods in [Banerjee and Chakravarty, 2011] and [Sheikhalishahi and Fabio, 2017]. Both the two methods apply secure sum protocol [Sheikh *et al.*, 2010] to obtain the sum of the data from multiple parties, and both the two methods focus on filter feature selection. The proposed method introduces a set of anchor data and each party shares the intermediate representations of the anchor data for collaborative feature selection without revealing their original data. Moreover, the proposed method focuses on embedded feature selection.

Secondly, the proposed method has a close relationship with the general embedded feature selection methods, such as the methods in [Hou *et al.*, 2011; Cai *et al.*, 2010; Ye and Sakurai, 2018]. The embedded feature selection methods consider local data structure preserving to find an embedded low dimensional data. In addition to preserving the local data structure, the proposed method utilizes the intermediate representations of the anchor data from multiple parties to find the low dimensional data. Furthermore, the proposed method can achieve privacy preserving and distributed collaboration.

Note that for privacy preserving, the proposed method only shares the intermediate representations of the anchor data from multiple parties. In the case that without privacy requirement while the data from multiple parties are difficult to collect for centralized analysis due to the huge data size, the proposed method is also applicable. A part of the original data collected from multiple parties can be used as the anchor data, and the process of feature selection is distributed in each party based on the shared intermediate representations of the anchor data.

4.3 Privacy Analysis

The proposed method does not use privacy-preserving techniques like cryptography or differential privacy that have a strong guarantee for privacy. Since differential privacy is considered to be too strong for some applications, some weakened versions of privacy have been proposed [Hall *et al.*, 2012]. Similarly, in our method, the term “privacy is preserved” can be defined as the case when the original data in each party cannot be (approximately) obtained by others. Here, we do not consider the privacy of the statistical datasets. According to the proposed method, the original data X_i in party i can be approximated only if one can obtain both the intermediate representation of X_i and the transformation matrix B_i. However, the intermediate representation of X_i and the transformation matrix B_i are not revealed to other parties. As shown in our algorithm, the intermediate representation of X_i is only distributed in party i, thus, the original dataset X_i



(a) 784 features (original))

(b) 100 features by LFS

(c) 100 features by DCFS

(d) 200 features by LFS

(e) 200 features by DCFS

(f) 784 features (original)

(g) 200 features by LFS

(h) 200 features by DCFS

Figure 1: Testing digitals with selected features in MNIST-d1 ((a), (b), (c), (d), (e)) and MNIST-d2 ((f), (g), (h))

cannot be (approximately) obtained by others. We will consider further analyzing the privacy in a more formal way for our method in future work.

5 Experimental Results

In this section, the proposed method is evaluated on several datasets. A dataset is divided into some parts. The data in one part is referred to as the data in one party. The proposed method is performed in each party for distributed collaborative feature selection. We apply k Nearest Neighbors (NN) classifier ($k = 1$) to test the performance of the proposed method. In each party, feature selection is performed on the training data to learn the features. The training data with the selected features are used to train the classifier and the test data with the selected features are used to test the result. Two widely used evaluation metrics, i.e., accuracy (ACC) and normalized mutual information (NMI) [Strehl and Ghosh, 2002], are used to evaluate the classification results.

We compare the proposed method, i.e., distributed collaborative feature selection (DCFS), with local feature selection (LFS) in each party using the method in Section 2.1. We also compare the case of collecting all data from each party for feature selection (AFS), and the baseline case using all the features (i.e., without feature selection) for classification. In the experiments, we generate 500 anchor data based on the original data in each party. The features of the anchor data are generated as random values between the maximum and minimum values of the corresponding features in the training data. Since the anchor data in each party is generated based on its original data, the anchor data gathered from all parties are related to the original data in all parties. The number of selected features is ranged from 1 to 300. α and β are tuned over $\{10^{-3}, 10^{-2}, 10^{-1}1, 10^{1}, 10^{2}, 10^{3}\}$. The neighborhood size is set as $k = 5$. Dimensionalities of embedding are set as $p = 10$ and $q = 5$. We report the best result of all the methods by using different parameters. All experiments are performed using MATLAB2018a on Mac OS X 10.13.6 (18C54) with core i7 CPU and 16GB ram.

5.1 Handwritten Digit Data

We first use the handwritten digit data (i.e., MNIST) [LeCun *et al.*, 1998] to generate two datasets for performance evaluation. MNIST consists of 5000 samples with 784 features. The two generated datasets are denoted as MNIST-d1 and MNIST-d2, respectively. MNIST-d1 contains the digitals 0, 1, 2, 3 and is divided into two parts. Party 1 consists of 270 samples of 1 and 10 samples of 0, 2, 3, respectively. Party 2 consists of 30 samples of 1 and 90 samples of 0, 2, 3, respectively. MNIST-d2 contains the digitals 4, 5, 6, 7, 8, 9 and is divided into six parts. Each part consists of about 90% of one digital and 10% of other digitals, e.g., party 1 consists of 275 samples of 4 and 5 samples of 5, 6, 7, 8, 9, respectively. We use five test data to test the performance of feature selection in MNIST-d1 and MNIST-d2, respectively. Each test data has 600 samples and all classes are equal in size. The mean results with standard deviation are reported.

Figure 1 shows some of the testing digitals with the selected features by using LFS and DCFS, respectively. We can see that, compared with LFS, the features selected by DCFS can better capture the data structure to represent the original data. In Figures 1 (b) and (d), the digital 0 is difficult to be recognized. In Figure 1 (g), 4 and 5 are difficult to be recognized.

(a) MNIST-d1

Party	Method	ACC	NMI
Party 1	LFS	81.3$_{1.0}$	63.5$_{7.6}$
	AFS	**87.2**$_{1.2}$	**72.0**$_{5.4}$
	Baseline	84.7$_{1.0}$	66.4$_{3.2}$
	DCFS	86.1$_{1.9}$	67.6$_{2.7}$
Party 2	LFS	96.7$_{1.0}$	90.7$_{2.8}$
	AFS	**97.2**$_{5.3}$	**92.7**$_{2.7}$
	Baseline	95.8$_{1.4}$	87.4$_{3.3}$
	DCFS	**97.2**$_{5.8}$	91.4$_{1.8}$

(b) MNIST-d2

Party	Method	ACC	NMI
Party 1	LFS	48.9$_{1.4}$	33.5$_{4.0}$
	AFS	**55.4**$_{1.3}$	**37.5**$_{1.7}$
	Baseline	51.2$_{0.9}$	36.8$_{2.2}$
	DCFS	52.8$_{0.9}$	37.4$_{3.1}$
Party 2	LFS	58.4$_{2.1}$	40.3$_{2.7}$
	AFS	**60.1**$_{0.6}$	**42.4**$_{2.5}$
	Baseline	58.6$_{1.4}$	41.2$_{2.3}$
	DCFS	58.9$_{2.5}$	41.0$_{2.2}$
Party 3	LFS	56.1$_{1.5}$	36.7$_{3.8}$
	AFS	**57.5**$_{2.2}$	**38.0**$_{2.7}$
	Baseline	55.9$_{1.2}$	37.2$_{1.7}$
	DCFS	56.7$_{0.9}$	37.9$_{2.3}$
Average of parties 4, 5, 6	LFS	48.8$_{1.9}$	30.7$_{2.7}$
	AFS	**50.1**$_{2.0}$	**31.6**$_{2.6}$
	Baseline	48.6$_{1.8}$	30.5$_{2.2}$
	DCFS	49.8$_{2.2}$	31.3$_{2.6}$

Table 1: Classification results on handwritten digit data

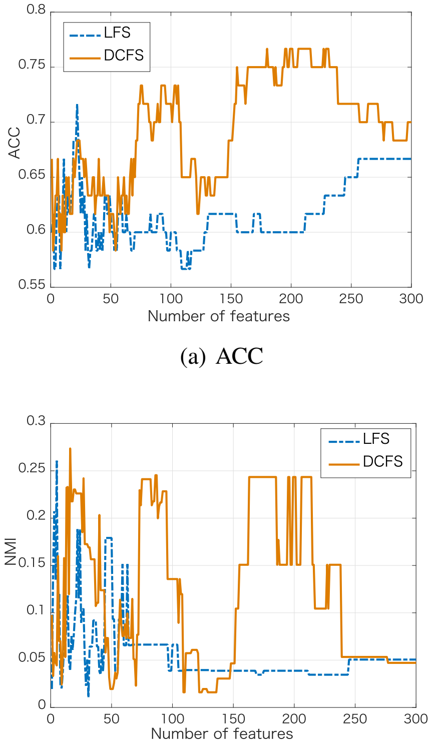
The classification results in each party on MNIST-d1 and MNIST-d2 are shown in Table 1. In each party, the best and the second best results are highlighted in bold-face type and underlined, respectively. For MNIST-d2, we show the results in three parties and the average result of the other three parties. We can see that feature selection can improve the results of classification. Collecting all data for centralized analysis obtain the best results in most cases. LFS performs worse than baseline in some parties due to the lack of information. The proposed method solves the information deficiency in the local party, thus performs better than LFS and also baseline on most parties.

5.2 Gene Expression Data

We use four gene expression datasets for performance evaluation. The datasets are Colon, TOX-171, Lung, and Lymphoma, which are downloaded from http://featureselection.asu.edu/datasets.php. We summarize the properties of the datasets in Table 2. Colon and TOX-171 are randomly divided into two parts with equal size, while Lung and Lymphoma are randomly divided into three parts with roughly equal size. For each dataset, 80% and 20% of the data are set as the training and test data in each party, We perform data divisions five times on each dataset and report the mean results with standard deviation.

Dataset	# of samples	# of features	# of clusters
Colon	62	2000	2
TOX-171	171	5748	4
Lung	203	3312	5
Lymphoma	96	4026	9

Table 2: Properties of datasets



(a) ACC

(b) NMI

Figure 2: Classification results on Colon vs. number of features

The classification results on the four gene expression datasets are shown in Table 3. The performance improvements by the three feature selection methods on gene expression datasets are more significant than that on MNIST. The proposed method performances better than local feature selection in each party. In party 1 on TOX-171, NMI of DCFS is better than AFS. That is because AFS focuses on preserving the local data structure in multiple parties. The proposed method can be seen as a trade-off between preserving the local data structure in a single party and in multiple parties. In some case, the trade-off can obtain better result.

Figure 2 shows the classification results on colon by varying the number of selected features. It is clear that DCFS performs better than LFS in terms of both ACC and NMI.

The classification results on Colon with different α and β are shown in Figure 3. As seen from Figure 3, when two parameters are changed within a certain range, the performance also changes within a certain range. The performance

(a) Colon

Party	Method	ACC	NMI
Party 1	LFS	$81.7_{1.1}$	$35.8_{1.8}$
	AFS	$\mathbf{85.0}_{3.7}$	$\mathbf{42.7}_{1.2}$
	Baseline	$65.0_{6.9}$	$6.5_{7.0}$
	DCFS	$\underline{83.3}_{8.3}$	$\underline{40.8}_{1.2}$
Party 2	LFS	$83.3_{5.9}$	$42.6_{8.5}$
	AFS	$\mathbf{88.3}_{9.5}$	$\mathbf{57.5}_{1.7}$
	Baseline	$76.7_{9.1}$	$31.1_{0.0}$
	DCFS	$\underline{85.0}_{6.9}$	$\underline{45.3}_{1.6}$

(b) TOX-171

Party	Method	ACC	NMI
Party 1	LFS	$58.4_{3.3}$	$39.6_{3.4}$
	AFS	$\mathbf{63.5}_{3.2}$	$\underline{43.7}_{2.2}$
	Baseline	$48.6_{4.7}$	$24.8_{9.8}$
	DCFS	$\underline{63.1}_{3.2}$	$\mathbf{46.1}_{4.3}$
Party 2	LFS	$58.2_{3.6}$	$37.2_{6.3}$
	AFS	$\mathbf{62.4}_{2.6}$	$\mathbf{45.1}_{3.0}$
	Baseline	$47.8_{3.3}$	$25.8_{5.3}$
	DCFS	$\underline{62.0}_{3.8}$	$\underline{44.1}_{4.0}$

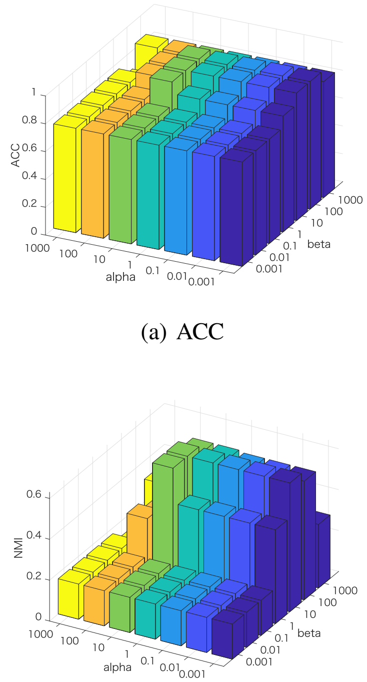
(c) Lung

Party	Method	ACC	NMI
Party 1	LFS	$96.0_{3.1}$	$86.4_{4.2}$
	AFS	$\mathbf{97.5}_{2.2}$	$\mathbf{88.7}_{2.2}$
	Baseline	$93.4_{3.8}$	$79.2_{6.5}$
	DCFS	$\underline{97.0}_{2.5}$	$\underline{87.2}_{4.3}$
Party 2	LFS	$93.0_{3.4}$	$79.1_{4.9}$
	AFS	$\mathbf{96.4}_{3.7}$	$\mathbf{84.0}_{5.0}$
	Baseline	$93.5_{4.3}$	$79.8_{4.8}$
	DCFS	$\underline{95.0}_{3.6}$	$\underline{82.3}_{4.7}$
Party 3	LFS	$94.5_{4.2}$	$82.7_{5.3}$
	AFS	$\mathbf{96.2}_{4.0}$	$\mathbf{83.8}_{5.2}$
	Baseline	$93.8_{4.3}$	$79.1_{5.1}$
	DCFS	$\underline{95.6}_{3.8}$	$\underline{83.2}_{4.5}$

(d) Lymphoma

Party	Method	ACC	NMI
Party 1	LFS	$79.4_{4.3}$	$81.0_{4.2}$
	AFS	$\mathbf{83.5}_{4.2}$	$\mathbf{81.7}_{4.2}$
	Baseline	$76.6_{4.7}$	$77.5_{6.2}$
	DCFS	$\underline{82.2}_{3.9}$	$\underline{81.4}_{4.5}$
Party 2	LFS	$79.1_{3.8}$	$80.6_{5.3}$
	AFS	$\mathbf{82.2}_{3.5}$	$\mathbf{82.3}_{4.0}$
	Baseline	$75.3_{4.3}$	$77.9_{5.6}$
	DCFS	$\underline{81.1}_{3.6}$	$\underline{81.9}_{4.1}$
Party 3	LFS	$78.2_{3.8}$	$80.7_{5.1}$
	AFS	$\mathbf{82.4}_{2.6}$	$\mathbf{83.5}_{3.8}$
	Baseline	$72.8_{4.3}$	$75.7_{5.3}$
	DCFS	$\underline{82.1}_{3.7}$	$\underline{82.5}_{4.2}$

Table 3: Classification results on gene expression data



(a) ACC

(b) NMI

Figure 3: Classification results on Colon vs. parameters α and β

in terms of ACC is more stable than that in terms of NMI.

6 Conclusion

In this paper, we proposed a novel distributed method for collaborative feature selection method by considering privacy preserving. The proposed method learns the intermediate representations of the anchor data from multiple parties. Embedded feature selection is performed in each party based on the intermediate representations, which can solve the information deficiency in the local party. We derive an effective algorithm to solve the optimization problem and present the convergence analysis. Experimental results show that the proposed method can improve the performance of feature selection in the local party. In the future work, we will use some statistical methods and investigate the practical techniques to improve the setting of anchor data for higher performance. We also consider to extend the proposed method for the supervised case and accelerate the algorithm.

Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful and constructive comments. The present study is supported in part by the Japan Science and Technology Agency (JST), ACT-I (No. JPMJPR16U6), the New Energy and Industrial Technology Development Organization (NEDO) and the Japan Society for the Promotion of Science (JSPS), Grants-in-Aid for Scientific Research (Nos. 17K12690, 18H03250).

References

[Banerjee and Chakravarty, 2011] Madhushri Banerjee and Sumit Chakravarty. Privacy preserving feature selection for distributed data using virtual dimension. In *Proceedings of ACM Conference on Information and Knowledge Management*, pages 2281–2284, 2011.

[Cai *et al.*, 2010] Deng Cai, Chiyuan Zhang, and Xiaofei He. Unsupervised feature selection for multi-cluster data. In *Proceedings of ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 333–342, 2010.

[Chaudhuri *et al.*, 2011] Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 9:1069–1109, 2011.

[Guyon and Elisseeff, 1997] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 1997.

[Hall *et al.*, 2012] Rob Hall, Rinaldo Alessandro, and Wasserman Larry. Random differential privacy. *Journal of Privacy and Confidentiality*, 4(2):43–59, 2012.

[He and Niyogi, 2004] Xiaofei He and Partha Niyogi. Locality preserving projections. *Neural information processing systems*, 16:153, 2004.

[He *et al.*, 2006] Xiaofei He, Deng Cai, , and Partha Niyogi. Laplacian score for feature selection. In *Advances in neural information processing systems*, pages 507–514, 2006.

[Hou *et al.*, 2011] Chenping Hou, Feiping Nie, Dongyun Yi, and Yi Wu. Feature selection via joint embedding learning and sparse regression. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 1324–1329, 2011.

[Imakura and Sakurai, 2019] Akira Imakura and Tetsuya Sakurai. Data collaboration analysis for distributed datasets. In *arXiv:1902.07535 [cs.LG]*, 2019.

[Imakura *et al.*, 2019] Akira Imakura, Momo Matsuda, Xiucai Ye, and Tetsuya Sakurai. Complex moment-based supervised eigenmap for dimensionality reduction. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, 2019.

[Ito and Murota, 2016] Shinji Ito and Kazuo Murota. An algorithm for the generalized eigenvalue problem for nonsquare matrix pencils by minimal perturbation approach. *SIAM Journal on Matrix Analysis and Applications*, 37:409–419, 2016.

[Jafer *et al.*, 2014] Yasser Jafer, Stan Matwinand, and Marina Sokolova. Task oriented privacy preserving data publishing using feature selection. In *in Advances in Artificial Intelligence - 27th Canadian Conference on Artificial Intelligence*, pages 143–154, 2014.

[Jain and Zongker, 1997] Anil Jain and Douglas Zongker. Feature selection: Evaluation, application, and small sample performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):153–158, 1997.

[LeCun *et al.*, 1998] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. The mnist database of handwritten digits. In *http://yann.lecun.com/exdb/mnist/*, 1998.

[Maldonado and Weber, 2009] Sebastián Maldonado and Richard Weber. A wrapper method for feature selection using support vector machines. information sciences. *Information Sciences*, 179(13):2208–2217, 2009.

[Nie *et al.*, 2010] Feiping Nie, Heng Huang, Xiao Cai, and Chris H. Ding. Efficient and robust feature selection via joint l2,1-norms minimization. In *Advances in neural information processing systems*, 2010.

[Roweis and Saul, 2000] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[Saeys *et al.*, 2008] Yvan Saeys, Thomas Abeel, and Yves Van de Peer. Robust feature selection using ensemble feature selection techniques. In *Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 313–325, 2008.

[Sheikh *et al.*, 2010] Rashid Sheikh, Beerendra Kumar, and Durgesh Kumar Mishra. A distributed k-secure sum protocol for secure multi-party computations. In *arXiv preprint arXiv:1003.4071*, 2010.

[Sheikhalishahi and Fabio, 2017] Mina Sheikhalishahi and Martinelli Fabio. Privacy-utility feature selection as a privacy mechanism in collaborative data classification. In *Proceedings of IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pages 244–249, 2017.

[Strehl and Ghosh, 2002] Alexander Strehl and Joydeep Ghosh. Cluster ensembles–a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.

[Yang and Li, 2014] Jun Yang and Yun Li. Differentially private feature selection. In *Proceedings of International Joint Conference on Neural Networks*, pages 4182–4189, 2014.

[Ye and Sakurai, 2018] Xiucai Ye and Tetsuya Sakurai. Unsupervised feature selection for microarray gene expression data based on discriminative structure learning. *Journal of Universal Computer Science*, 24(6):725–741, 2018.

[Ye *et al.*, 2016a] Xiucai Ye, Kaiyang Ji, and Tetsuya Sakurai. Global discriminant analysis for unsupervised feature selection with local structure preservation. In *Proceedings of the Twenty-Ninth International Florida Artificial Intelligence Research Society Conference*, pages 454–459, 2016.

[Ye *et al.*, 2016b] Xiucai Ye, Kaiyang Ji, and Tetsuya Sakurai. Unsupervised feature selection with correlation and individuality analysis. *International Journal of Machine Learning and Computing*, 6(1):36–41, 2016.