Another Use of SMOTE for Interpretable Data Collaboration Analysis

Akira Imakura[1,*], Masateru Kihira[1], Yukihiko Okada[1], and Tetsuya Sakurai[1]

[1]University of Tsukuba, 1-1-1 Tennodai, Ibaraki, Tsukuba 305-8573, Japan

[*]Email : imakura@cs.tsukuba.ac.jp

arXiv:2208.12458v1 [cs.LG] 26 Aug 2022

Abstract

Recently, data collaboration (DC) analysis has been developed for privacy-preserving integrated analysis across multiple institutions. DC analysis centralizes individually constructed dimensionality-reduced *intermediate representations* and realizes integrated analysis via *collaboration representations* without sharing the original data. To construct the collaboration representations, each institution generates and shares a shareable *anchor dataset* and centralizes its intermediate representation. Although, random anchor dataset functions well for DC analysis in general, using an anchor dataset whose distribution is close to that of the raw dataset is expected to improve the recognition performance, particularly for the interpretable DC analysis. Based on an extension of the synthetic minority over-sampling technique (SMOTE), this study proposes an anchor data construction technique to improve the recognition performance without increasing the risk of data leakage. Numerical results demonstrate the efficiency of the proposed SMOTE-based method over the existing anchor data constructions for artificial and real-world datasets. Specifically, the proposed method achieves 9 percentage point and 38 percentage point performance improvements regarding accuracy and essential feature selection, respectively, over existing methods for an income dataset. The proposed method provides another use of SMOTE not for imbalanced data classifications but for a key technology of privacy-preserving integrated analysis.

1 Introduction

1.1 Background and motivation

There is a growing demand for the integrated analysis of data owned by multiple organizations in a distributed manner [4, 9, 23]. In some real-world applications, such as financial, medical, and manufacturing data analyses, it is difficult to share the original data for analysis because of data confidentiality, and privacy-preserving analysis methods, in which datasets are collaboratively analyzed without sharing the original data.

Motivating examples are interorganizational and intraorganizational data collaborations. An interorganizational example is about a relationship among companies and banks. Typically, a company borrows from and transacts with a few banks. The credit and financial data are distributed among borrowing and transacting banks. Despite the difficulty of sharing distributed data, creditors and analysts desire to collaborate for credibility and future profitability predictions.

On the other hand, intraorganizational collaborations may be needed. Universities have their entrance examination, learning processes, and healthcare students' data. However, the

data are distributed and cannot be used for a smarter campus life. Despite the difficulty of sharing distributed data, students' mentors and counselors desire to collaborate to improve diagnosis and advice.

The federated learning systems [15] that Google introduced are attracting research attention as typical technologies for this topic. However, the conventional federated learning requires cross-institutional communication during each iteration [4, 15, 18, 21, 23, 27]. An integrated analysis of multiple institutions motivated this study. In this scenario, many cross-institutional communications can be a significant issue in social implementation.

1.2 Main purpose and contributions

We focus on *data collaboration (DC) analysis*, a non-model share-type federated learning that has recently been developed for supervised learning [9, 11, 12], novelty detection [13], and feature selection [28]. DC analysis centralizes the dimensionality-reduced *intermediate representations*. The centralized intermediate representations are transformed to incorporable forms called *collaboration representations* using a shareable *anchor dataset*. Then, the collaborative representation is analyzed as a single dataset. Unlike federated learnings, DC analysis does not require iterative computations with cross-institutional communications.

The DC analysis performance strongly depends on the anchor dataset, although random anchor data functions well in general [11–13]. The use of the anchor dataset, whose distribution is close to that of the raw dataset, is expected to improve the recognition performance of DC analysis. However, using the anchor dataset such that samples are close to the raw data samples causes data leakage. The anchor data establishment is essential for both the recognition performance and privacy of DC analysis.

This study specifically focuses on the interpretable DC analysis [9] which constructs an interpretable model based on DC framework. Then, we propose an anchor data construction technique based on an extension of the synthetic minority over-sampling technique (SMOTE) [3], which is a data augmentation method for classification of imbalanced datasets, to improve the recognition performance without increasing the risk of data leakage. The main contributions of this study are

- We propose an anchor data construction technique based on an extension of SMOTE for DC analysis which is a recent non-model share-type federated learning for privacy-preserving integrated analysis.
- The proposed SMOTE-based method improves the recognition performance of the interpretable DC analysis without increasing the risk of data leakage.
- Numerical results demonstrate the efficiency of the proposed SMOTE-based method over the existing anchor data constructions owing the contribution of the extension of SMOTE.
- The proposed method provides another use of SMOTE not for imbalanced data classifications but for a key technology of privacy-preserving integrated analysis.

2 Related works

2.1 Federated Learning

Recently, federated learning systems have been developed for distributed data analysis and privacy preservation. The concept of federated learning was first proposed by Google [15] typically for Android phone model updates [21]. Federated learning is primarily based on (deep) neural networks and updates iteratively the model [4, 15, 18, 21, 23, 27].

Federated stochastic gradient descent (FedSGD) and federated averaging (FedAvg) are standard techniques for updating the model [21]. FedSGD is a direct extension of the stochastic gradient descent method. During each iteration of the gradient descent method, each party locally computes a gradient from the shared model using the local dataset before sending it to the server. The shared gradients are averaged and used to update the model. In FedAvg, each party performs multiple batch updates using the local dataset and sends the updated model to the server. Then, the shared models are update via averaging. These federated learning also including more recent methods such as FedProx [19] and FedCodl [23], requires cross-institutional communication in each iteration.

For more details, we refer to [18, 27] and references therein.

2.2 Interpretable data collaboration (DC) analysis

Here, we describe DC analysis for analyzing the following horizontally and vertically partitioned data:

$$X = \begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,d} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ X_{c,1} & X_{c,2} & \cdots & X_{c,d} \end{bmatrix} \in \mathbb{R}^{n \times m}, \quad Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_c \end{bmatrix}.$$

Note that DC analysis is applicable to more complicated data distributions [12, 22].

DC analysis operates in two roles: *worker* and *master*. Workers have the private dataset $X_{i,j} \in \mathbb{R}^{n_i \times m_j}$ ($n = \sum_{i=1}^{c} n_i$ and $m = \sum_{j=1}^{d} m_j$) and corresponding ground truth Y_i, which must be analyzed without sharing $X_{i,j}$.

First, all workers generate the same anchor dataset $X^{\mathrm{anc}} = [X^{\mathrm{anc}}_{:,1}, X^{\mathrm{anc}}_{:,2}, \ldots, X^{\mathrm{anc}}_{:,d}] \in \mathbb{R}^{r \times m}$, which is shareable data consisting of public or dummy data randomly constructed. Then, using a dimensionality reduction function $f_{i,j}$, each worker constructs dimensionality-reduced intermediate representations

$$\widetilde{X}_{i,j} = f_{i,j}(X_{i,j}) \in \mathbb{R}^{n_i \times \widetilde{m}_{i,j}}, \quad \widetilde{X}^{\mathrm{anc}}_{i,j} = f_{i,j}(X^{\mathrm{anc}}_{:,j}) \in \mathbb{R}^{r \times \widetilde{m}_{i,j}}$$

where $\widetilde{m}_{i,j} < m_j$, and centralizes them to the master. A typical setting for dimensionality reduction function is non-supervised dimensionality reduction methods, such as principal component analysis (PCA) [14], locality preserving projection (LPP) [8], and nonnegative matrix factorization (NMF) [17] and supervised dimensionality reduction methods, such as linear discriminant analysis (LDA) [5], local Fisher discriminant analysis (LFDA) [26], Locality adaptive discriminant analysis (LADA) [20], and complex moment-based supervised eigenmap (CMSE) [10].

Algorithm 1 Interpretable data collaboration analysis

Input (for worker-side): $X_{i,j} \in \mathbb{R}^{n_i \times m_j}$ and Y_i individually
Output (for worker-side): Interpretable models t_i $(i = 1, 2, \ldots, c)$

Worker-side (i, j)

Generate X^{anc} and share with all workers
Generate $f_{i,j}$
Compute $\widetilde{X}_{i,j} = f_{i,j}(X_{i,j})$ and $\widetilde{X}^{\mathrm{anc}}_{i,j} = f_{i,j}(X^{\mathrm{anc}}_{:,j})$
Share $\widetilde{X}_{i,j}, \widetilde{X}^{\mathrm{anc}}_{i,j}$, and Y_i to master

Master-side

$\searrow$ Obtain $\widetilde{X}_{i,j}, \widetilde{X}^{\mathrm{anc}}_{i,j}$, and Y_i for all i and j
Set $\widetilde{X}_i, \widetilde{X}^{\mathrm{anc}}_i$, and Y
Compute G_i from $\widetilde{X}^{\mathrm{anc}}_i$ for all i
Compute $\widehat{X}_i = \widetilde{X}_i G_i$ for all i, and set $\widehat{X}$
Analyze $\widehat{X}$ and Y to obtain $h(\widehat{X}) \approx Y$
Compute $Y^{\mathrm{anc}}_i = h(\widetilde{X}^{\mathrm{anc}}_i G_i)$ for all i
$\swarrow$ Return Y^{anc}_i to each worker

Worker-side (i, j)

Obtain Y^{anc}_i
Analyze X^{anc} and Y^{anc}_i to obtain $t_i(X^{\mathrm{anc}}) \approx Y^{\mathrm{anc}}_i$

On the master-side, the following collaboration representation

$$\widehat{X}_{i,j} = g_i(\widetilde{X}_i) \in \mathbb{R}^{n_i \times \widehat{m}}, \quad \widetilde{X}_i = [\widetilde{X}_{i,1}, \widetilde{X}_{i,2}, \ldots, \widetilde{X}_{i,d}] \in \mathbb{R}^{n_i \times \widetilde{m}_i}$$

is set such that collaboration representations of the anchor data are approximately the same, where $\widetilde{m}_i = \sum_{j=1}^{d} \widetilde{m}_{i,j}$, practically by solving a minimal perturbation problem. The collaboration representations are then analyzed as a single dataset. We obtain the prediction result of the anchor data by applying the obtained model to the anchor data's collaboration representation. Sharing the prediction result of the anchor data with the workers, interpretable model are constructed on the worker-side using the anchor data and its prediction result.

The algorithm of the interpretable DC analysis is shown in Algorithm 1. For details, please refer to [9].

2.3 SMOTE

Classification performance is negatively impacted by imbalanced data for classification problems when the number of samples for each label in the training dataset differ significantly. In such scenario, under-sampling techniques, which remove majority data with many samples, and over-sampling techniques, which generate minority data with few samples, are used. Particularly, when the imbalance ratio is high, under-sampling methods must remove many samples, resulting in the deterioration of classification performance.

SMOTE is the most typical and pioneering work of over-sampling technique [3]. SMOTE generates a new dataset using a randomized interpolation with k nearest neighbors. Let $\boldsymbol{x}_i$ be a minority class sample and $\check{\boldsymbol{x}}_i^{(1)}, \check{\boldsymbol{x}}_i^{(2)}, \ldots, \check{\boldsymbol{x}}_i^{(\ell)}$ be ℓ samples randomly selected from k nearest neighbors with the same label, of which $\ell \leq k$, where k is generally set to a small value (five by default). Then, the new data is generated by

$$\boldsymbol{x}_i^{(j)} = \boldsymbol{x}_i + c_{i,j}(\check{\boldsymbol{x}}_i^{(j)} - \boldsymbol{x}_i), \quad j = 1, 2, \ldots, \ell,$$

where $c_{i,j}$ denotes a coefficient of the randomized interpolation, which is randomly set in $[0, 1]$.

Because SMOTE uniformly generates the new data from the minority samples, it increases the number of samples that are not necessarily important for classification. Therefore, improvement methods that intensively generate samples near the classification boundary, such as ADASYN [7], borderline SMOTE [6], and safe-level SMOTE [2], have been proposed.

3 Another use of SMOTE for interpretable data collaboration (DC) analysis

Let n and m be the number of samples and the dimensionality of the data, respectively. Additionally, let $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]^{\mathrm{T}} \in \mathbb{R}^{n \times m}$ and $Y = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n] \in \mathbb{R}^{n \times \ell}$ be a data matrix and the corresponding ground truth, respectively. In this study, for privacy-preserving analysis of multiple parties, we consider horizontal and vertical data partitioning, where data samples and features are partitioned into c and d parties, respectively, as follows:

$$X = \begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,d} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ X_{c,1} & X_{c,2} & \cdots & X_{c,d} \end{bmatrix}, \quad Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_c \end{bmatrix}. \tag{1}$$

Additionally, we assume that we have a small amount of shareable public data $X^{\mathrm{pub}} \in \mathbb{R}^{p \times m}$ with $p \ll n$. Note that there are no ground truth data for X^{pub}.

3.1 Existing anchor data construction

In most of the existing studies, [11–13], anchor data $X^{\mathrm{anc}} \in \mathbb{R}^{r \times m}$ were constructed as a random matrix in the range of the corresponding features as follows:

$$X^{\mathrm{anc}} = [x_{i,j}^{\mathrm{anc}}]_{1 \leq i \leq r, 1 \leq j \leq m}, \quad x_{i,j}^{\mathrm{anc}} \sim \mathcal{U}(x_j^{\min}, x_j^{\max}),$$

where $x_j^{\min}$ and $x_j^{\max}$ denote the minimum and maximum values of the j-th feature in the original data X, respectively, and $\mathcal{U}(a, b)$ denotes the uniform distribution on range $[a, b]$. Hereafter, we will refer to this method as a random anchor data construction method.

In [9] for interpretable DC collaboration analysis, a low-rank approximation method was used to construct the anchor data closer to the raw data. In each worker, the local anchor data $X_{i,j}^{\mathrm{approx}}$ are constructed using a low-rank approximation of $X_{i,j}$ with random perturbation as follows:

$$X_{i,j}^{\mathrm{approx}} = X_{i,j}^{\mathrm{TSVD}} + \delta E,$$

where $X_{i,j}^{\mathrm{TSVD}} \in \mathbb{R}^{n_i \times m_j}$ denotes a low-rank approximation based on the truncated SVD of $X_{i,j}$, δ denotes a perturbation parameter, and $E \in \mathbb{R}^{n_i \times m_i}$ denotes a random matrix. By sharing $X_{i,j}^{\mathrm{approx}}$ with all users, n samples of anchor data are generated, as follows:

$$
\begin{aligned}
X^{\mathrm{approx}} &= [X_{:,1}^{\mathrm{approx}}, X_{:,2}^{\mathrm{approx}}, \ldots, X_{:,d}^{\mathrm{approx}}] \\
&= \begin{bmatrix}
X_{1,1}^{\mathrm{approx}} & X_{1,2}^{\mathrm{approx}} & \cdots & X_{1,d}^{\mathrm{approx}} \\
X_{2,1}^{\mathrm{approx}} & X_{2,2}^{\mathrm{approx}} & \cdots & X_{2,d}^{\mathrm{approx}} \\
\vdots & \vdots & \ddots & \vdots \\
X_{c,1}^{\mathrm{approx}} & X_{c,2}^{\mathrm{approx}} & \cdots & X_{c,d}^{\mathrm{approx}}
\end{bmatrix} \in \mathbb{R}^{n \times m}.
\end{aligned}
$$

Next, to generate r samples of anchor data X^{anc}, we apply an augmentation technique using a linear combination if $r > n$, or we select r samples randomly otherwise (that is, $r \leq n$). Hereafter, we will refer to this method as a TSVD-based anchor data construction method.

3.2 Proposal for a SMOTE-based anchor data construction

In the TSVD-based anchor data construction method, the constructed anchor data X^{anc} is expected to be closer to the raw data X by increasing the rank of $X_{i,j}^{\mathrm{TSVD}}$ and decreasing δ, which improves the recognition performance of DC analysis. However, the data leakage risk increases. To improve the recognition performance without increasing the data leakage risk, we propose an anchor data construction technique based on an extension of SMOTE.

In each local party, we generate an anchor dataset X^{anc} using a SMOTE-based method from X^{pub} using the same random numbers. To generate X^{anc} that mimics the distribution of the raw data from X^{pub} with small samples ($p \ll n$), we extend SMOTE to change the range of parameter c as $[0, \alpha]$ to allow even extrapolation. Note that the conventional SMOTE considers only interpolation. Additionally, a larger value is taken for the number of neighbors k, which is generally set to a small value in conventional SMOTE (five by default). These broaden the data distribution. The contribution of these extensions of SMOTE for DC analysis will be evaluated in Section 4.4.

Note that conventional SMOTE is generally used for imbalance data. In contrast, this study performs over-sampling on all X^{pub} data to construct the anchor dataset that mimics the raw data to improve the recognition performance of DC analysis. For this reason, the classical SMOTE is used instead of its improvements, which heavily oversamples on the classification boundary, as typified by ANASYN.

The algorithm of the proposed SMOTE-based anchor data construction is summarized in Algorithm 2.

Here, we analyze the variance of the new samples. Let $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ be the original samples that are independently selected (k of k nearest neighbors are set as n) so that it has zero covariance, $\mathrm{cov}(\boldsymbol{x}_1, \boldsymbol{x}_2) = 0$. Here, we assume that the expected values of the samples are zero, $E[\boldsymbol{x}_1] = E[\boldsymbol{x}_2] = 0$. Additionally, let c be a uniform random number in $[0, \alpha]$ ($\alpha > 0$).

Then, the variance of the new sample $V[\boldsymbol{x}'] = V[\boldsymbol{x}_1 + c(\boldsymbol{x}_2 - \boldsymbol{x}_1)]$ can be written as

Algorithm 2 A SMOTE-based anchor data construction

Input (for worker-side): $X^{\mathrm{pub}} \in \mathbb{R}^{p \times m}$, the number of anchor data r and parameters α and k

Output (for worker-side): Anchor data $X^{\mathrm{anc}} \in \mathbb{R}^{r \times m}$

Normalize X^{pub}
for $i = 1, 2, \ldots, p$ **do:**
  Set k nearest neighbors of $\boldsymbol{x}_i^{\mathrm{pub}}$
  Randomly select r/p samples $\check{\boldsymbol{x}}_i^{(j)}$ $(j = 1, 2, \ldots, r/p)$ from k nearest neighbors
  for $j = 1, 2, \ldots, r/p$ **do:**
    Randomly set $c_{i,j} \in [0, \alpha]$ and compute $\boldsymbol{x}_i^{(j)} = \boldsymbol{x}_i + c_{i,j}(\check{\boldsymbol{x}}_i^{(j)} - \boldsymbol{x}_i)$.
  end for
end for
Set X^{anc} as all generated vectors $\boldsymbol{x}_i^{(j)}$
Denormalize X^{anc}

follows:

$$
\begin{aligned}
&V[\boldsymbol{x}_1 + c(\boldsymbol{x}_2 - \boldsymbol{x}_1)] \\
&\quad = V[(1-c)\boldsymbol{x}_1] + V[c\boldsymbol{x}_2] \\
&\quad = V[(1-c)]V[\boldsymbol{x}_1] + E[1-c]^2 V[\boldsymbol{x}_1] + V[c]V[\boldsymbol{x}_2] + E[c]^2 V[\boldsymbol{x}_2] \\
&\quad = \frac{\alpha^2}{12} V[\boldsymbol{x}_1] + \left(1 - \frac{\alpha}{2}\right)^2 V[\boldsymbol{x}_1] + \frac{\alpha^2}{12} V[\boldsymbol{x}_2] + \left(\frac{\alpha}{2}\right)^2 V[\boldsymbol{x}_2] \\
&\quad = \left(\frac{2}{3}\alpha^2 - \alpha + 1\right) V[\boldsymbol{x}_1],
\end{aligned}
$$

where we used $V[c] = \alpha^2/12$ and $E[c] = \alpha/2$. Thus, we have $V[\boldsymbol{x}'] = V[\boldsymbol{x}_1]$ when $\alpha = 1.5$, $V[\boldsymbol{x}'] < V[\boldsymbol{x}_1]$ when $\alpha < 1.5$, and $V[\boldsymbol{x}'] > V[\boldsymbol{x}_1]$ when $\alpha > 1.5$.

4 Numerical experiments

This section evaluates the efficiency of the proposed SMOTE-based anchor data construction method (Algorithm 2) for the interpretable DC analysis (**DC(SMOTE)**) and compares it with existing anchor data construction techniques: using a random matrix (**DC(rand)**) and a TSVD (**DC(TSVD)**) introduced in Section 3.1. We also evaluate a scenario in which the raw data is used for the anchor data (**DC(raw)**). We compared the interpretable DC analysis with the centralized analysis that shares the raw datasets (**Centralized**) and the local analysis that only uses local dataset $X_{i,j}$ (**Local**) to assess prediction accuracy. Note that **DC(raw)** and **Centralized** are considered ideal cases because the raw data cannot be shared in our target situation.

4.1 General settings

We used PCA for dimensionality reduction method on worker-side (Step 2 in Algorithm 1) for the interpretable DC analysis. We set $\widehat{m} = \widetilde{m}_i$. We set $p = 100$ for the number of public data X^{pub}. We also set the ground truth Y as a binary matrix whose (i, j)-th entry is 1 if the

training data $\boldsymbol{x}_i$ are in class j and 0 otherwise. This type of ground truth has been applied to various classification algorithms, including ridge regression and deep neural networks [1].

Here, we evaluate the efficiency of the anchor data construction methods regarding data confidentiality and recognition performance. For data confidentiality, we evaluated the similarity between the anchor data and the raw data by

- Earth mover's distance (EMD)

$$\mathrm{EMD}(X, X^{\mathrm{anc}}) = \min_{f_{i,j}} \sum_{i,j} f_{i,j} \|\boldsymbol{x}_i - \boldsymbol{x}_j^{\mathrm{anc}}\|_2,$$

  where $f_{i,j} = 0$ or 1, $\sum_{i=1}^{n} f_{i,j} = 1$ and $\sum_{j=1}^{r} f_{i,j} = 1$.

- Average minimum distance from the raw data (AMD(raw))

$$\mathrm{AMD}(\mathrm{raw}) = \mathrm{AMD}(X, X^{\mathrm{anc}}) = \frac{1}{n} \sum_{i=1}^{n} \min_{1 \le j \le r} \|\boldsymbol{x}_i - \boldsymbol{x}_j^{\mathrm{anc}}\|_2.$$

- Average minimum distance from the anchor data (AMD(anc))

$$\mathrm{AMD}(\mathrm{anc}) = \mathrm{AMD}(X^{\mathrm{anc}}, X) = \frac{1}{r} \sum_{i=1}^{r} \min_{1 \le j \le n} \|\boldsymbol{x}_i^{\mathrm{anc}} - \boldsymbol{x}_j\|_2.$$

For recognition performance, we evaluated the constructed interpretable model by

- Accuracy (ACC) of prediction result
  ACC is the ratio of correct predictions, defined as

$$\mathrm{ACC}(Y^{\mathrm{GT}}, Y^{\mathrm{Pred}}) = \frac{\text{number of correct predictions}}{\text{number of test samples}},$$

  where Y^{GT} and Y^{Pred} denote the ground truth and prediction result for the test data, respectively.

- Normalized mutual information (NMI)
  NMI is the mutual information (MI) score normalized to produce results between 0 (no mutual information) and 1 (perfect correlation), defined as

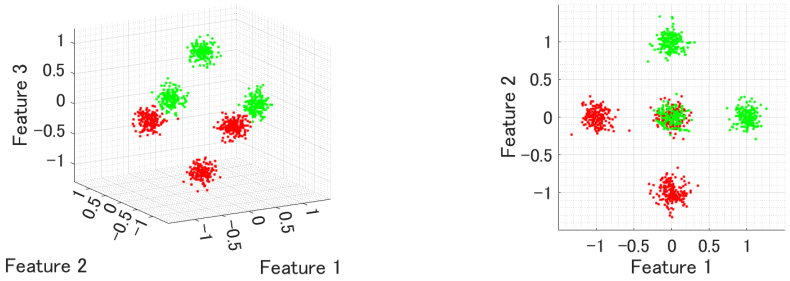
$$\mathrm{NMI}(Y^{\mathrm{Pred}}, Y^{\mathrm{GT}}) = \frac{I(Y^{\mathrm{Pred}}; Y^{\mathrm{GT}})}{\sqrt{H(Y^{\mathrm{Pred}}) H(Y^{\mathrm{GT}})}},$$

  where Y^{GT} and Y^{pred} are the ground truth and prediction result, respectively, and $I(Y^{\mathrm{Pred}}; Y^{\mathrm{GT}})$ and $H(\cdot)$ are the mutual information and entropy, respectively; see [25] for more details.

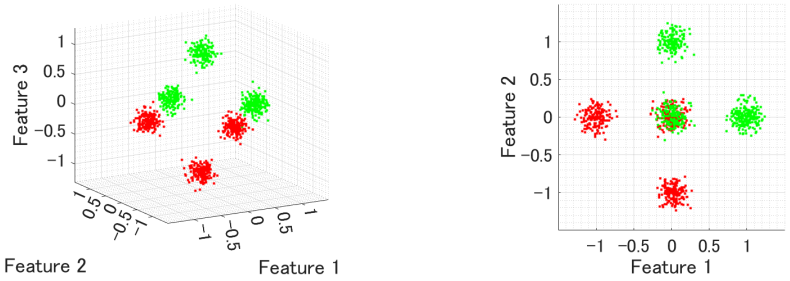
- Similarity of estimated top t essential features (Dice_t)
  We define similarity of estimated essential features as Dice index, defined as

$$\mathrm{Dice}_t(\mathcal{F}_t^*, \mathcal{F}_t^{\mathrm{Pred}}) = \frac{|\mathcal{F}_t^* \cap \mathcal{F}_t^{\mathrm{Pred}}|}{t},$$

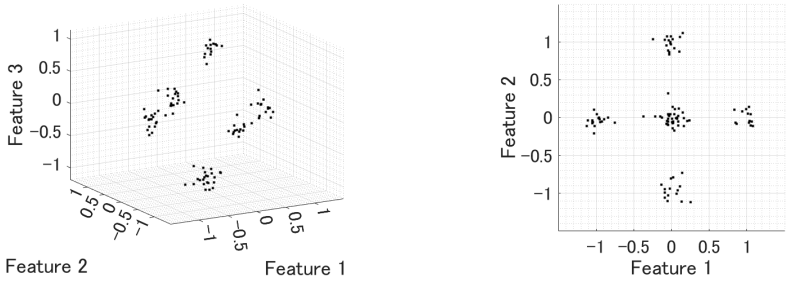
  where $\mathcal{F}_t^*$ and $\mathcal{F}_t^{\mathrm{Pred}}$ denote the estimated top t essential features computed by centralized analysis that share the raw data and the intermediate DC analysis. Note that $|\mathcal{F}_t^*| = |\mathcal{F}_t^{\mathrm{Pred}}| = t$.



(a) Training dataset



(b) Test dataset



(c) Public dataset

Figure 1: Training, test, and public datasets for the artificial problem.

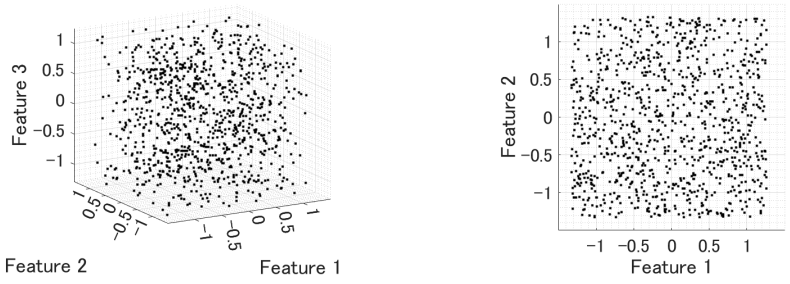
Numerical experiments were performed using Python and MATLAB [1].

4.2 Proof-of-concept for artificial dataset

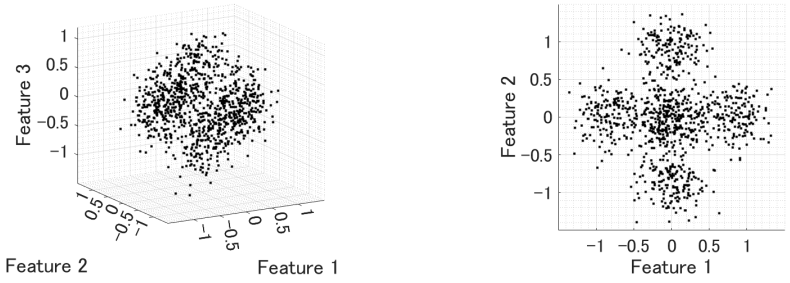
As a proof of concept of the proposed method, we used a 20-dimensional artificial data for two-class classification. Figure 1 shows features 1, 2, and 3 of the training, test, and public datasets, where the numbers of samples are $(n_{\text{train}}, n_{\text{test}}, p) = (1000, 1000, 100)$. The other 17 dimensions have random values. Note that features 1, 2, and 3 are essential features for classification.

We considered the case where the training dataset in Figure 1(a) is distributed into four

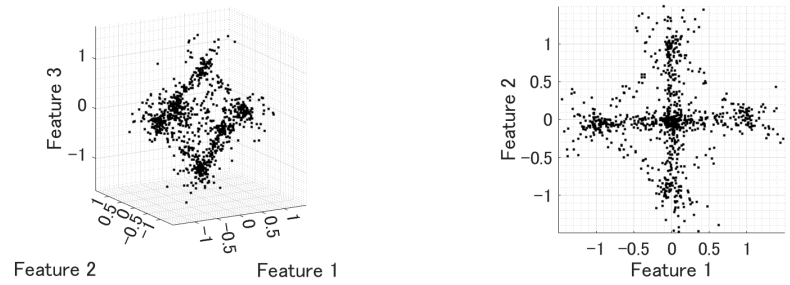
[1] Program codes are available from the corresponding author by reasonable request.



(a) Random anchor data construction



(b) TSVD-based anchor data construction



(c) SMOTE-based anchor data construction

Figure 2: Constructed anchor data for the artificial problem.

parties, $c = d = 2$, as

$$X = \begin{bmatrix} X_{1,1} & X_{1,2} \\ X_{2,1} & X_{2,2} \end{bmatrix} \in \mathbb{R}^{1000\times 20}, \quad X_{1,1}, X_{1,2}, X_{2,1}, X_{2,2} \in \mathbb{R}^{500\times 10}.$$

For horizontal (sample) partitioning, we randomly partitioned the dataset into two groups. For vertical (feature) partitioning, $X_{1,1}, X_{2,1}$ have odd features, and $X_{1,2}, X_{2,2}$ have even features. Note that the essential features (1, 2, and 3) are partitioned into two groups.

For the DC analysis, we set the dimensionality of intermediate representations to $\widetilde{m}_{i,j} = 5$ for all parties and the number of anchor data to $r = 1000$. We used the ridge regression for analyzing the collaboration representation (Step 9 in Algorithm 1) and the decision tree as interpretable model (Step 13 in Algorithm 1). We set five as the maximum number of branch node splits. For the TSVD-based method, we set the rank of TSVD to 3. For the proposed SMOTE-based method, we set $(k, \alpha) = (25, 1.5)$.

Table 1: Prediction results of each method for the artificial dataset.

Method	Recognition performance			Data confidentiality	
	NMI	ACC	Dice_3	AMD(raw)	AMD(anc)
Centralized	0.99±0.00	1.00±0.00	1.00±0.00		
Local	0.26±0.02	0.75±0.01	0.50±0.02		
DC(raw)	0.97±0.01	1.00±0.00	1.00±0.00	0.00±0.00	0.00±0.00
DC(rand)	0.52±0.05	0.85±0.02	0.92±0.03	2.15±0.00	2.23±0.00
DC(TSVD)	0.84±0.07	0.95±0.02	0.93±0.03	1.87±0.00	1.58±0.00
DC(SMOTE)	0.97±0.01	0.99±0.00	1.00±0.00	2.02±0.00	1.90±0.01

As numerical results, we show constructed the anchor data using each method in Figure 2. We also show the average and standard error of recognition performance (NMI, ACC, and Dice_3) and data confidentiality (AMD(raw) and AMD(anc)) for each method across 20 trials in Table 1.

We note that unlike the random anchor data construction, the TSVD-based and SMOTE-based methods generated anchor data along the distribution of the raw data. Regarding recognition performance (NMI, ACC, and Dice_3 in Table 1), **DC(SMOTE)** correctly finds the top 3 essential features and shows high recognition performance comparable to **DC(raw)** and **Centralized**. Specifically, **DC(SMOTE)** achieved 7 percentage point (NMI), 4 percentage point (ACC), and 7 percentage point (Dice_3) performance improvements over **DC(TSVD)**, respectively. Note that **DC(TSVD)** performs better performance than **Local** and **DC(rand)**, but worse than **DC(SMOTE)**. Regarding data confidentiality (AMD(raw) and AMD(anc) in Table 1), **DC(SMOTE)** shows larger values than **DC(TSVD)**, which indicates that **DC(SMOTE)** has better data confidentiality than **DC(TSVD)**.

Overall, **DC(SMOTE)** demonstrates a high level of both recognition accuracy and data confidentiality for the artificial dataset.

4.3 Evaluation regarding recognition performance and data confidentiality on a credit rating dataset

Here, we evaluate the trade-off between recognition performance and data confidentiality of each method for a credit rating dataset "CreditRating_Historical.dat" from the MATLAB Statistics and Machine Learning Toolbox. The dataset contains five financial ratios, i.e., Working capital / Total Assets (WC_TA), Retained Earnings / Total Assets (RE_TA), Earnings Before Interests and Taxes / Total Assets (EBIT_TA), Market Value of Equity / Book Value of Total Debt (MVE_BVTD), and Sales / Total Assets (S_TA), and industry sector labels from 1 to 12 for 3932 customers. The dataset also includes credit ratings from "AAA" to "CCC" for all customers. The categorical variable "Industry sector labels" was transformed into 12 dimensional dummy variables. Note that this dataset is simulated, not real.

We sought to predict credit rating using the five financial ratios and industry sector labels. We considered the case where the training dataset with 3,000 samples is distributed into four parties, $c = d = 2$, as

$$X = \begin{bmatrix} X_{1,1} & X_{1,2} \\ X_{2,1} & X_{2,2} \end{bmatrix} \in \mathbb{R}^{3000 \times 17}$$

Table 2: Recognition performance of each method for the CreditRating_Historical.dat.

Method	NMI	ACC	Dice_2
Centralized	0.61±0.01	0.74±0.01	1.00
Local	0.46±0.01	0.60±0.00	
DC(raw)	0.61±0.01	0.74±0.01	1.00
DC(rand)	0.51±0.04	0.54±0.07	1.00
DC(TSVD) rank=1	0.45±0.07	0.52±0.08	0.55
DC(TSVD) rank=2	0.58±0.02	0.71±0.02	1.00
DC(TSVD) rank=3	0.60±0.01	0.73±0.02	1.00
DC(SMOTE)	0.61±0.01	0.74±0.01	1.00

Table 3: Data confidentiality of each method for the CreditRating_Historical.dat.

Method	EMD	AMD(raw)	AMD(anc)
DC(raw)	0.00±0.00	0.00±0.00	0.00±0.00
DC(rand)	52.84±9.49	1.53±0.17	8.40±2.49
DC(TSVD) rank=1	2.12±0.07	1.72±0.13	0.63±0.01
DC(TSVD) rank=2	1.17±0.04	0.28±0.00	0.85±0.02
DC(TSVD) rank=3	1.17±0.04	0.26±0.00	0.85±0.02
DC(SMOTE)	1.15±0.26	0.36±0.07	0.50±0.17

with

$$X_{1,1}, X_{2,1} \in \mathbb{R}^{1500\times 3}, \quad X_{1,2}, X_{2,2} \in \mathbb{R}^{1500\times 14},$$

where, $X_{1,1}, X_{2,1}$ have the 1st group of features "WC_TA", "RE_TA", and "EBIT_TA", and $X_{1,2}, X_{2,2}$ have the 2nd group of features "MVE_BVTD", "S_TA", and "Industry sector labels" as features.

For the DC analysis, we set the dimensionality of intermediate representations to $\widetilde{m}_{i,j} = m_j - 1$ for all parties and the number of anchor data to $r = 2500$. We used the XGBoost method with default parameters in Python for analyzing the collaboration representation (Step 9 in Algorithm 1) and for interpretable model (Step 13 in Algorithm 1). For the TSVD-based method, we set the rank of TSVD to 1–3. For the proposed SMOTE-based method, we set $(k, \alpha) = (99, 1.5)$.

As numerical results, we show the average and standard error of recognition performance (NMI, ACC, and Dice_2) and data confidentiality (EMD, AMD(raw), and AMD(anc)) for each method across ten trials in Table 2 and 3, respectively.

Table 2 shows that the DC methods correctly finds the top 2 essential features except **DC(TSVD)** with rank 1. Additionally, the proposed **DC(SMOTE)** shows a high recognition performance comparable to **DC(raw)** and **Centralized**. However, regarding data confidentiality (Table 3), **DC(SMOTE)** shows larger AMD(raw), smaller AMD(anc), and almost the same EMD compared with **DC(TSVD)** with ranks 2 and 3. This result indicates that **DC(SMOTE)** shows almost the same data confidentiality as **DC(TSVD)** with rank 2 and 3.

Table 4: Recognition performance of each method for Adult with Artificial data partitioning scenario.

Method	NMI	ACC	Dice_5
Centralized	0.34±0.00	0.87±0.00	1.00±0.00
Local	0.22±0.00	0.83±0.00	0.50±0.00
DC(raw)	0.33±0.00	0.87±0.00	1.00±0.00
DC(rand)	0.12±0.05	0.72±0.12	0.24±0.12
DC(TSVD) rank=2	0.00±0.01	0.76±0.00	0.46±0.18
DC(TSVD) rank= $m_j - 1$	0.00±0.01	0.76±0.00	0.54±0.13
DC(SMOTE)	0.27±0.02	0.85±0.01	0.92±0.10

4.4 Evaluation regarding recognition performance for income dataset with two data distribution scenarios

Here, we evaluate the recognition performance of each method for an income dataset "Adult" from the UCI Machine Learning Repository. The prediction task is to determine whether a person makes more than $50,000 per year. We used five continuous features and seven categorical features, excluding "fnlwgt"and "ducation". The seven categorical variables were transformed into 86-dimensional dummy variables.

We considered the case where the training dataset with 30,000 samples is distributed into four parties, $c = d = 2$, as

$$X = \begin{bmatrix} X_{1,1} & X_{1,2} \\ X_{2,1} & X_{2,2} \end{bmatrix} \in \mathbb{R}^{15{,}000 \times 91}$$

with

$$X_{1,1}, X_{2,1} \in \mathbb{R}^{15{,}000 \times m_1}, \quad X_{1,2}, X_{2,2} \in \mathbb{R}^{15{,}000 \times m_2},$$

where samples were randomly partitioned, and features were partitioned based on two distribution scenarios:

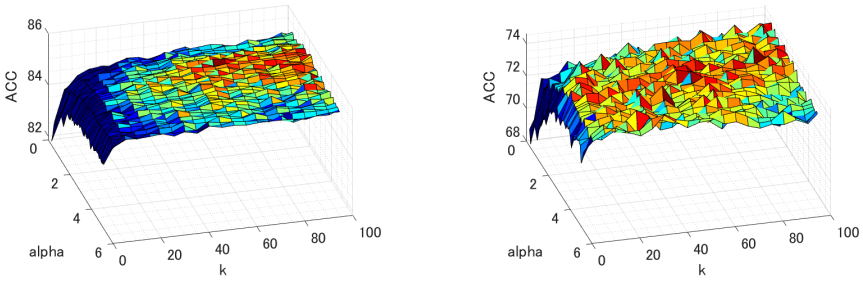
- Artificial:
  91 features are partitioned into two groups using even or odd indices. Here, $m_1 = 46$ and $m_2 = 45$.
- Feature type:
  91 features are partitioned into two groups using continuous or categorical features. Here, $m_1 = 5$ and $m_2 = 86$.

For the DC analysis, we set the dimensionality of intermediate representations to $\widetilde{m}_{i,j} = m_j - 1$ for all parties and the number of anchor data to $r = 2500$. We used the XGBoost method with default parameters in Python for analyzing the collaboration representation (Step 9 in Algorithm 1) and for interpretable model (Step 13 in Algorithm 1). For the TSVD-based method, we set the rank of TSVD to 2 and $m_j - 1$. For the proposed SMOTE-based method, we set $(k, \alpha) = (99, 1.5)$.

Tables 4 and 5 show that the proposed **DC(SMOTE)** has a high recognition performance comparable to **DC(raw)** and **Centralized** compared with **DC(rand)** and **DC(TSVD)**. Additionally, the proposed **DC(SMOTE)** finds the top 5 essential features with higher rates

Table 5: Recognition performance of each method for Adult with Feature type data partitioning scenario.

Method	NMI	ACC	Dice_5
Centralized	0.34±0.00	0.87±0.00	1.00±0.00
Local	0.22±0.00	0.83±0.00	0.50±0.00
DC(raw)	0.32±0.00	0.87±0.00	0.98±0.06
DC(rand)	0.13±0.04	0.78±0.03	0.40±0.13
DC(TSVD) rank=2	0.22±0.01	0.82±0.00	0.46±0.09
DC(TSVD) rank= $m_j - 1$	0.21±0.01	0.81±0.01	0.48±0.10
DC(SMOTE)	0.26±0.03	0.85±0.01	0.80±0.13



(a) "CreditRating_Historical.dat" (b) "Adult"

Figure 3: Parameter dependency of the proposed method.

than **DC(rand)** and **DC(TSVD)** even for Feature type data distribution scenario. Specifically, **DC(SMOTE)** achieved 9 percentage point (ACC) and 38 percentage point (Dice_5) performance improvements over **DC(TSVD)** for Artificial data distribution scenario, and 4 percentage point (ACC) and 32 percentage point (Dice_5) performance improvements over **DC(TSVD)** for Feature type data distribution scenario.

4.5 Evaluation for parameter dependency

To evaluate the parameter dependency of the proposed method, we used the "CreditRating_Historical.dat" and "Adult". We evaluated the recognition performance of the proposed method by varying $k = 1, 5, 10, \ldots, 95, 99$ and $\alpha = 0.1, 0.2, \ldots, 5.0$. Other parameters were set as in Sections 4.2 and 4.3.

Figure 3 shows the average ACC of the proposed method. This result implies that the proposed method with the standard parameter settings of the conventional SMOTE ($\alpha = 1$ and a small k) provides low accuracy. Instead, the accuracy could be improved by the extension using a large α and a large k.

Table 6: Recognition performance (average $\pm$ standard error) for real-world problems (the first five of ten datasets).

Dataset	Method	NMI	ACC
Carcinom	Centralized	0.66 ± 0.03	54.58 ± 3.06
$m = 9182$	Local	0.50 ± 0.04	40.67 ± 3.12
$n = 174$	DC(TSVD)	0.66 ± 0.04	54.06 ± 5.30
	DC(SMOTE)	0.66 ± 0.03	56.16 ± 3.53
CLL-SUB-111	Centralized	0.22 ± 0.03	60.40 ± 2.86
$m = 11340$	Local	0.16 ± 0.02	56.81 ± 1.64
$n = 111$	DC(TSVD)	0.29 ± 0.08	52.06 ± 5.43
	DC(SMOTE)	0.08 ± 0.02	56.80 ± 3.15
GLA-BRA-180	Centralized	0.37 ± 0.05	62.22 ± 3.74
$m = 49151$	Local	0.28 ± 0.02	55.74 ± 1.75
$n = 180$	DC(TSVD)	0.33 ± 0.04	61.67 ± 2.41
	DC(SMOTE)	0.29 ± 0.06	59.44 ± 2.79
jaffe	Centralized	0.68 ± 0.02	38.95 ± 1.43
$m = 676$	Local	0.64 ± 0.01	42.36 ± 1.51
$n = 213$	DC(TSVD)	0.59 ± 0.04	31.46 ± 2.46
	DC(SMOTE)	0.62 ± 0.04	40.67 ± 6.82
leukemia	Centralized	0.74 ± 0.14	94.64 ± 2.95
$m = 7129$	Local	0.39 ± 0.06	80.39 ± 2.39
$n = 72$	DC(TSVD)	0.39 ± 0.10	81.61 ± 3.94
	DC(SMOTE)	0.50 ± 0.13	87.68 ± 3.50

4.6 Performance evaluation on real-world data

Next, we evaluate the performance of the methods when applied to the binary and multi-class classification problems presented by [16, 24] and feature selection datasets [2]. Note that these datasets were used in [9].

We considered the case where each dataset is distributed into six parties: $c = 2$ and $d = 3$. The performance of each method was evaluated using a five-fold cross-validation framework.

For the DC analysis, we set the dimensionality of intermediate representations to $\widetilde{m}_{i,j} = 15$ for all parties and the number of anchor data to $r = 2500$. We used the ridge regression for analyzing the collaboration representation (Step 9 in Algorithm 1) and the decision tree for interpretable model (Step 13 in Algorithm 1). We set five as the maximum number of branch node splits. For the TSVD-based method, we set the rank of TSVD to 20. For the proposed SMOTE-based method, we set $(k, \alpha) = (50, 1.5)$.

The numerical results for each method are presented in Tables 6 and 7 for ten datasets. Tables 6 and 7 show that the recognition performance of the proposed method is better than that of **DC(TSVD)** and **Local** on most datasets.

[2] available at `http://featureselection.asu.edu/datasets.php`.

Table 7: Recognition performance (average $\pm$ standard error) for real-world problems (the second five of ten datasets).

Dataset	Method	NMI	ACC
lung	Centralized	0.69 ± 0.05	88.14 ± 2.08
$m = 3312$	Local	0.52 ± 0.03	78.06 ± 1.50
$n = 203$	DC(TSVD)	0.64 ± 0.05	86.74 ± 1.57
	DC(SMOTE)	0.65 ± 0.03	85.64 ± 2.11
pixraw10P	Centralized	0.68 ± 0.02	38.00 ± 1.79
$m = 10000$	Local	0.63 ± 0.04	41.00 ± 3.12
$n = 100$	DC(TSVD)	0.61 ± 0.03	29.00 ± 0.89
	DC(SMOTE)	0.65 ± 0.04	35.00 ± 1.41
Prostate_GE	Centralized	0.40 ± 0.10	84.09 ± 3.63
$m = 5966$	Local	0.28 ± 0.04	75.30 ± 2.20
$n = 102$	DC(TSVD)	0.31 ± 0.07	78.36 ± 2.37
	DC(SMOTE)	0.58 ± 0.08	90.18 ± 2.01
TOX-171	Centralized	0.37 ± 0.03	59.70 ± 2.62
$m = 5789$	Local	0.30 ± 0.01	49.52 ± 1.51
$n = 171$	DC(TSVD)	0.34 ± 0.04	49.68 ± 3.72
	DC(SMOTE)	0.38 ± 0.02	60.24 ± 2.12
warpAR10P	Centralized	0.64 ± 0.02	30.00 ± 2.96
$m = 2400$	Local	0.59 ± 0.02	34.87 ± 1.39
$n = 130$	DC(TSVD)	0.59 ± 0.03	30.77 ± 3.61
	DC(SMOTE)	0.58 ± 0.05	30.77 ± 3.61

4.7 Remarks

In numerical experiments, the proposed **DC(SMOTE)** exhibits a high level of both recognition accuracy and data confidentiality for artificial and real-world datasets. Numerical experiments demonstrate that the recognition performance of the proposed **DC(SMOTE)** is improved owing the contribution of the extension of SMOTE using a large α and a large k.

5 Conclusions

This study was motivated by an integrated analysis of multiple financial institutions. Here, extensive cross-institutional communication can be a significant problem in social implementation. DC analysis has recently been developed as one of the more logical options for an integrated analysis with small cross-institutional communications. In this study, we specifically focused on the interpretable DC analysis and proposed a SMOTE-based technique for anchor data construction to improve the accuracy and privacy. For the proposed method, we extended SMOTE to allow even extrapolation and used a large k of k-nearest neighbors. Numerical results demonstrate the efficiency of the proposed SMOTE-based method over the existing anchor data constructions owing to the contribution of the extension of SMOTE. Specifically, the proposed method achieves 9 percentage point and 38 percentage point performance improvements regarding accuracy and essential feature selection, respectively, over

existing methods for an income dataset.

Privacy-preserving integrated analyses are an essential challenge to address in real-world applications, such as medical, financial, and manufacturing data analyses. The DC analysis using the proposed SMOTE-based anchor data construction is a breakthrough technology for such types of distributed data analyses. Additionally, the proposed method provides another use of SMOTE not for imbalanced data classifications but for a key technology of privacy-preserving integrated analysis.

On the other hand, in social implementation, the integrated analysis faces some difficulties, such as loss of data and batch effects. This study did not consider these difficulties to be inherent in social implementation. In future, we will intend to address these difficulties. We also intend to further analyze the confidentiality of the proposed SMOTE-based method and develop software.

Acknowledgements

This work was supported in part by the New Energy and Industrial Technology Development Organization (NEDO), Japan Science and Technology Agency (JST) (No. JPMJPF2017), the Japan Society for the Promotion of Science (JSPS), Grants-in-Aid for Scientific Research (Nos. JP19KK0255, JP21H03451, JP22H00895, JP22K19767).

References

[1] C. M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer-Verlag Berlin, Heidelberg, 2006.

[2] C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap, Safe-level-SMOTE: safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem, in: Pacific-Asia conference on knowledge discovery and data mining, Springer, 2009.

[3] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, Journal of artificial intelligence research 16 (2002) 321–357.

[4] S. Feng, Vertical federated learning-based feature selection with non-overlapping sample utilization, Expert Systems with Applications 208 (2022) 118097.

[5] R. A. Fisher, The use of multiple measurements in taxonomic problems, Annals of human genetics 7 (2) (1936) 179–188.

[6] H. Han, W.-Y. Wang, B.-H. Mao, Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning, in: International conference on intelligent computing, Springer, 2005.

[7] H. He, Y. Bai, E. A. Garcia, S. Li, ADASYN: adaptive synthetic sampling approach for imbalanced learning, in: 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence), IEEE, 2008.

[8] X. He, P. Niyogi, Locality preserving projections, in: Advances in neural information processing systems, 2004.

[9] A. Imakura, H. Inaba, Y. Okada, T. Sakurai, Interpretable collaborative data analysis on distributed data, Expert Systems with Applications 177 (2021) 114891.

[10] A. Imakura, M. Matsuda, X. Ye, T. Sakurai, Complex moment-based supervised eigenmap for dimensionality reduction, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019.

[11] A. Imakura, T. Sakurai, Data collaboration analysis framework using centralization of individual intermediate representations for distributed data sets, ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering 6 (2020) 04020018.

[12] A. Imakura, X. Ye, T. Sakurai, Collaborative data analysis: Non-model sharing-type machine learning for distributed data, in: Uehara H., Yamaguchi T., Bai Q. (eds) Knowledge Management and Acquisition for Intelligent Systems. PKAW 2021. Lecture Notes in Computer Science, vol. 12280, 2021.

[13] A. Imakura, X. Ye, T. Sakurai, Collaborative novelty detection for distributed data by a probabilistic method, in: Proceedings of The 13th Asian Conference on Machine Learning (ACML 2021), 2021.

[14] I. T. Jolliffe, Principal component analysis and factor analysis, in: Principal component analysis, Springer, 1986, pp. 115–128.

[15] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, D. Bacon, Federated learning: Strategies for improving communication efficiency, in: NIPS Workshop on Private Multi-Party Machine Learning, 2016.

[16] Y. LeCun, The MNIST database of handwritten digits, http://yann. lecun. com/exdb/mnist/.

[17] D. D. Lee, H. S. Seung, Algorithms for non-negative matrix factorization, in: Proceedings of the 13th International Conference on Neural Information Processing Systems, MIT Press, 2000.

[18] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, B. He, A survey on federated learning systems: Vision, hype and reality for data privacy and protection, arXiv preprint (2019) arXiv:1907.09693.

[19] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks, Proceedings of Machine Learning and Systems 2 (2020) 429–450.

[20] X. Li, M. Chen, F. Nie, Q. Wang, Locality adaptive discriminant analysis, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, AAAI Press, 2017.

[21] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, et al., Communication-efficient learning of deep networks from decentralized data, arXiv preprint (2016) arXiv:1602.05629.

[22] A. Mizoguchi, A. Imakura, T. Sakurai, Application of data collaboration analysis to distributed data with misaligned features, Informatics in Medicine Unlocked 32 (2022) 101013.

[23] X. Ni, X. Shen, H. Zhao, Federated optimization via knowledge codistillation, Expert Systems with Applications 191 (2022) 116310.

[24] F. Samaria, A. Harter, Parameterisation of a stochastic model for human face identification, in: Proceeding of IEEE Workshop on Applications of Computer Vision, 1994.

[25] A. Strehl, J. Ghosh, Cluster ensembles—a knowledge reuse framework for combining multiple partitions, Journal of machine learning research 3 (Dec) (2002) 583–617.

[26] M. Sugiyama, Dimensionality reduction of multimodal labeled data by local Fisher discriminant analysis, Journal of machine learning research 8 (May) (2007) 1027–1061.

[27] Q. Yang, Y. Liu, T. Chen, Y. Tong, Federated machine learning: Concept and applications, ACM Transactions on Intelligent Systems and Technology 10 (2) (2019) Article 12.

[28] X. Ye, H. Li, A. Imakura, T. Sakurai, Distributed collaborative feature selection based on intermediate representation, in: The 28th International Joint Conference on Artificial Intelligence (IJCAI-19), 2019.