KHÓA LUẬN TỐT NGHIỆP QUY DẪN TỪ TRƯỜNG HỢP TRUNG BÌNH VỀ TRƯỜNG HỢP XẤU NHẤT DỰA TRÊN ĐỘ ĐO GAUSS

Sinh viên: Phan Thành Nhân.
Giảng viên hướng dẫn: TS Lê Văn Luyện.

ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP. HỒ CHÍ MINH

Contents

1. Introduction and Motivation
2. ComImp algorithm
    - ComImp algorithm
    - ComImp algorithm for combining datasets
3. Combining datasets with dimension reduction (PCA-ComImp)
4. Choices of missing data imputation methods
5. Experiments

Introduction and Motivation

Example

Suppose that we have two datasets

$$\mathcal{D}_1 = \begin{pmatrix} \text{person} & \text{height (cm)} & \text{weight (kg)} \\ 1 & 120 & 80 \\ 2 & 150 & 70 \\ 3 & 140 & 80 \\ 4 & 135 & 85 \end{pmatrix},$$

$$\mathcal{D}_2 = \begin{pmatrix} \text{person} & \text{weight (kg)} & \text{calo/meal} \\ 5 & 90 & 100 \\ 6 & 85 & 150 \\ 7 & 92 & 170 \end{pmatrix}.$$

Introduction and Motivation

Motivation

Combining the information from these datasets together can help increase the sample sizes and may allow more efficient model training, prediction, and inferences.
This motivates us to propose ***ComImp*** (Combine datasets based on Imputation), a framework that allows vertically combine datasets based on missing data imputation.

ComImp algorithm

Input:

1. datasets $\mathcal{D}_i = \{\mathcal{X}_i, \mathbf{y}_i\}, i = 1, ..., r$
2. $\mathcal{F}_i$: set of features in $\mathcal{X}_i$
3. $g(\mathcal{X}, \mathcal{H})$: a transformation that rearranges features in $\mathcal{X}$ to follow the order of the features in the set of features $\mathcal{H}$ and insert empty columns if a feature in $\mathcal{H}$ does not exist in $\mathcal{X}$
4. Imputer I.

Procedure:

```
1: $\mathcal{F} = \bigcup_{i=1}^{r} \mathcal{F}_i$
2: $\mathcal{X}_i^* \leftarrow g(\mathcal{X}_i, \mathcal{F})$
3: $\mathcal{X}^* = \begin{pmatrix} \mathcal{X}_1^* \\ \mathcal{X}_2^* \\ \vdots \\ \mathcal{X}_r^* \end{pmatrix}, \mathbf{y} = \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_r \end{pmatrix}$
4: $\mathcal{X} \leftarrow$ imputed version of $\mathcal{X}^*$ using imputer I.
```

Return: $\mathcal{D} = \{\mathcal{X}, \mathbf{y}\}$, the combined dataset of $\mathcal{D}_1, ..., \mathcal{D}_r$.

ComImp algorithm for combining datasets

As an illustration, we use the datasets $\mathcal{D}_1, \mathcal{D}_2$. A newly merged dataset $\mathcal{D}$ can be formed by the following process:

1. The union of set of features $\mathcal{F}$ consists of *height, weight, and calo/meal*,
2.

$$\mathcal{X}_1^* = \begin{pmatrix} \text{person} & \text{height} & \text{weight} & \text{(calo/meal)} \\ 1 & 120 & 80 & * \\ 2 & 150 & 70 & * \\ 3 & 140 & 80 & * \\ 4 & 135 & 85 & * \end{pmatrix},$$

$$\mathcal{X}_2^* = \begin{pmatrix} \text{person} & \text{height} & \text{weight} & \text{calo/meal} \\ 5 & * & 90 & 100 \\ 6 & * & 85 & 150 \\ 7 & * & 92 & 170 \end{pmatrix},$$

ComImp algorithm for combining datasets

3. Stack $\mathcal{X}_1^*, \mathcal{X}_2^*$ vertically, and stack $\mathbf{y}_1, \mathbf{y}_2$ vertically

$$\mathcal{X}^* = \begin{pmatrix} \text{person} & \text{height} & \text{weight} & \text{calo/meal} \\ 1 & 120 & 80 & * \\ 2 & 150 & 70 & * \\ 3 & 140 & 80 & * \\ 4 & 135 & 85 & * \\ 5 & * & 90 & 100 \\ 6 & * & 85 & 150 \\ 7 & * & 92 & 170 \end{pmatrix},$$

and

$$\mathbf{y} = \begin{pmatrix} 80 \\ 90 \\ 85 \\ 95 \\ 100 \\ 95 \\ 82 \end{pmatrix}$$

ComImp algorithm for combining datasets

4. $\mathcal{X}$ = imputed version of $\mathcal{X}^*$. Suppose that mean imputation is being used, then

$$\mathcal{X} = \begin{pmatrix} \text{(person)} & \text{(height)} & \text{(weight)} & \text{(calo/meal)} \\ 1 & 120 & 80 & 140 \\ 2 & 150 & 70 & 140 \\ 3 & 140 & 80 & 140 \\ 4 & 135 & 85 & 140 \\ 5 & 136.25 & 90 & 100 \\ 6 & 136.25 & 85 & 150 \\ 7 & 136.25 & 92 & 170 \end{pmatrix},$$

5. $\mathcal{D} = (\mathcal{X}, \mathbf{y})$.

Combining datasets with dimension reduction (PCA-ComImp)

Input:

1. Datasets $\mathcal{D}_i = \{\mathcal{X}_i, \mathbf{y}_i\}, i = 1, 2$ consists of $\mathcal{D}_i^{(tr)} = \{\mathcal{X}_i^{(tr)}, \mathbf{y}_i^{(tr)}\}, i = 1, 2$ as the training sets and $\mathcal{D}_i^{(ts)} = \{\mathcal{X}_i^{(ts)}, \mathbf{y}_i^{(ts)}\}$ as the test sets,
2. $\mathcal{F}_i$: set of features in $\mathcal{X}_i$,
3. $g(\mathcal{X}, \mathcal{H})$: a transformation that rearranges features in $\mathcal{X}$ to follow the order of the features in the set of features $\mathcal{H}$ and insert empty columns if a feature in $\mathcal{H}$ does not exist in $\mathcal{X}$
4. Imputer I, classifier $\mathcal{C}$.

Combining datasets with dimension reduction (PCA-ComImp)

Procedure:

```
$\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2,\ \mathcal{S} = \mathcal{F}_1 \cap \mathcal{F}_2,\ \mathcal{Q}_1 = \mathcal{F}_1 \setminus \mathcal{F}_2,\ \mathcal{Q}_2 = \mathcal{F}_2 \setminus \mathcal{F}_1$
$(\mathcal{R}_i^{(tr)}, V) \leftarrow pca(\mathcal{Q}_i^{(tr)})$ and $\mathcal{R}_i^{(ts)} \leftarrow \mathcal{Q}_i^{(ts)} V$
$\mathcal{H} = \mathcal{S} \cup \mathcal{R}_1 \cup \mathcal{R}_2$
$\mathcal{X}_i^* \leftarrow g(\mathcal{S} \cup \mathcal{R}_i, \mathcal{H}), i = 1, 2$
$\mathcal{X}^* = \begin{pmatrix} \mathcal{X}_1^* \\ \mathcal{X}_2^* \end{pmatrix}, \mathbf{y} = \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix}$
$\mathcal{X} \leftarrow$ imputed version of $\mathcal{X}^*$ using imputer I.
```

Return: $\mathcal{D} = \{\mathcal{X}, \mathbf{y}\}$, the combined dataset of $\mathcal{D}_1, \mathcal{D}_2$.

Choices of missing data imputation methods

Theorem

Assume that we have two datasets $\mathcal{D}_1 = \{\mathbf{U}, \mathbf{y}\}, \mathcal{D}_2 = \{\mathbf{V}, \boldsymbol{z}\}$ *where* $\mathbf{U}$ *,* $\mathbf{V}$ *are inputs, and* $\mathbf{y}, \boldsymbol{z}$ *are the labels, such that*

$$\mathbf{U} = (\mathbf{1}_n \ \ \mathbf{u}_1) = \begin{pmatrix} 1 & u_{11} \\ 1 & u_{21} \\ \vdots & \vdots \\ 1 & u_{n1} \end{pmatrix}, \quad \mathbf{V} = (\mathbf{1}_m \ \ \mathbf{v}_1 \ \ \mathbf{v}_2) = \begin{pmatrix} 1 & v_{11} & v_{12} \\ 1 & v_{21} & v_{22} \\ \vdots & \vdots & \vdots \\ 1 & v_{m1} & v_{m2} \end{pmatrix}, \tag{1}$$

where $u_{ij}, v_{ij} \in \mathbb{R}$*, and*

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \boldsymbol{z} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_m \end{pmatrix}, \mathbf{Y} = \begin{pmatrix} \mathbf{y} \\ \boldsymbol{z} \end{pmatrix}. \tag{2}$$

Choices of missing data imputation methods

Theorem

Next, let $\bar{\mathbf{v}}_2$ be the mean of $\mathbf{v}_2$. If mean imputation is being used for ComImp then the resulting combined input is

$$\mathbf{X} = \begin{pmatrix} 1_n & \mathbf{u}_1 & \bar{\mathbf{v}}_2 \\ 1_m & \mathbf{v}_1 & \mathbf{v}_2 \end{pmatrix}. \tag{3}$$

Then, we have the following relation between the sum of squared errors (SSE) of the model fitted on $\mathcal{D} = \{\mathbf{X}, \mathbf{Y}\}$,

$$SSE_{\mathcal{D}} \geq SSE_{\mathcal{D}_1} + SSE_{\mathcal{D}_2}. \tag{4}$$

Choices of missing data imputation methods

Proof: Note that equation (4) is equivalent to

$$\mathbf{Y}'(I-\mathbf{H}_x)\mathbf{Y} \leq \mathbf{y}'(I-\mathbf{H}_u)\mathbf{y} + \mathbf{z}'(I-\mathbf{H}_v)\mathbf{z}, \tag{5}$$

where

$$\mathbf{H}_u = \mathbf{U}(\mathbf{U}'\mathbf{U})^{-1}\mathbf{U}', \tag{6}$$

$$\mathbf{H}_v = \mathbf{V}(\mathbf{V}'\mathbf{V})^{-1}\mathbf{V}', \tag{7}$$

$$\mathbf{H}_x = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'. \tag{8}$$

Hence, we want to prove that

$$\mathbf{y}'\mathbf{H}_u\mathbf{y} + \boldsymbol{z}'\mathbf{H}_v\boldsymbol{z} \geq \mathbf{Y}'\mathbf{H}_x\mathbf{Y}, \tag{9}$$

Choices of missing data imputation methods

Without loss of generality, assume that the data is centered so that $\sum_{i=1}^{n} u_{i1} = 0$, $\sum_{i=1}^{n} v_{i1} = 0$, $\sum_{i=1}^{n} v_{i2} = 0$, so $\bar{\mathbf{v}}_2 = 0$. Next, let $\alpha = \sum_{i=1}^{n} y_i$, $\alpha_1 = \sum_{i=1}^{n} y_i u_{i1}$, $a_1 = \sum_{i=1}^{n} u_{i1}^2$, we have

$$\mathbf{y}'\mathbf{U} = \begin{pmatrix} \sum_{i=1}^{n} y_i & \sum_{i=1}^{n} y_i u_{i1} \end{pmatrix} = \begin{pmatrix} \alpha & \alpha_1 \end{pmatrix},$$

$$\mathbf{U}'\mathbf{U} = \begin{pmatrix} 1 & \sum_{i=1}^{n} u_{i1} \\ \sum_{i=1}^{n} u_{i1} & \sum_{i=1}^{n} u_{i1}^2 \end{pmatrix} = \begin{pmatrix} n & 0 \\ 0 & a_1 \end{pmatrix},$$

$$\mathbf{U}'\mathbf{y} = \begin{pmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} y_i u_{i1} \end{pmatrix} = \begin{pmatrix} \alpha \\ \alpha_1 \end{pmatrix}.$$

Choices of missing data imputation methods

This implies

$$\begin{aligned}\mathbf{y}'\mathbf{H}_u\mathbf{y} &= \mathbf{y}'\left(\mathbf{U}(\mathbf{U}'\mathbf{U})^{-1}\mathbf{U}'\right)\mathbf{y} = \left(\mathbf{y}'\mathbf{U}\right)\left(\mathbf{U}'\mathbf{U}\right)^{-1}\left(\mathbf{U}'\mathbf{y}\right)\\ &= \frac{\alpha^2}{n} + \frac{\alpha_1^2}{a_1}. \qquad (10)\end{aligned}$$

Similarly, let $\beta = \sum_{i=1}^m z_i$, $\beta_1 = \sum_{i=1}^m z_i v_{i1}$, $\beta_2 = \sum_{i=1}^m z_i v_{i2}$, $b_1 = \sum_{i=1}^m v_{i1}^2$, $b_1 = \sum_{i=1}^m v_{i2}^2$, $c = \sum_{i=1}^m v_{i1}v_{i2}$. Then,

$$\begin{aligned}\mathbf{z}'\mathbf{V} &= \begin{pmatrix}\sum_{i=1}^m z_i & \sum_{i=1}^m z_i v_{i1} & \sum_{i=1}^m z_i v_{i12}\end{pmatrix}\\ &= \begin{pmatrix}\beta & \beta_1 & \beta_2\end{pmatrix}\end{aligned}$$

Choices of missing data imputation methods

$$\mathbf{V}'\mathbf{V} = \begin{pmatrix} m & \sum\limits_{i=1}^{m} v_{i1} & \sum\limits_{i=1}^{m} v_{i2} \\ \sum\limits_{i=1}^{m} v_{i1} & \sum\limits_{i=1}^{m} v_{i1}^2 & \sum\limits_{i=1}^{m} v_{i1}v_{i2} \\ \sum\limits_{i=1}^{m} v_{i2} & \sum\limits_{i=1}^{m} v_{i1}v_{i2} & \sum\limits_{i=1}^{m} v_{i2}^2 \end{pmatrix}$$

$$= \begin{pmatrix} m & 0 & 0 \\ 0 & b_1 & c \\ 0 & c & b_2 \end{pmatrix},$$

$$\mathbf{V}'\mathbf{z} = \begin{pmatrix} \sum\limits_{i=1}^{m} z_i \\ \sum\limits_{i=1}^{m} z_i v_{i1} \\ z_i v_{i12} \end{pmatrix} = \begin{pmatrix} \beta \\ \beta_1 \\ \beta_2 \end{pmatrix}.$$

Choices of missing data imputation methods

Therefore,

$$\begin{aligned}\mathbf{z}'\mathbf{H}_z\mathbf{z} &= \mathbf{z}'\left(\mathbf{V}(\mathbf{V}'\mathbf{V})^{-1}\mathbf{V}'\right)\mathbf{z} = \left(\mathbf{z}'\mathbf{V}\right)\left(\mathbf{V}'\mathbf{V}\right)^{-1}\left(\mathbf{V}'\mathbf{z}\right)\\ &= \frac{\beta^2}{m} + \frac{\beta_1^2 b_2 + \beta_2^2 b_1 - 2\beta_1\beta_2 c}{b_1 b_2 - c^2}\\ &= \frac{\beta^2}{m} + \frac{\left(\beta_1 b_2 - \beta_2 c\right)^2}{b_2\left(b_1 b_2 - c^2\right)} + \frac{\beta_2^2}{b_2}. \end{aligned} \tag{11}$$

Besides,

$$\begin{aligned}\mathbf{Y}'\mathbf{X} &= \begin{pmatrix}\sum_{i=1}^{n} y_i + \sum_{i=1}^{m} z_i & \sum_{i=1}^{n} y_i u_{i1} + \sum_{i=1}^{m} z_i v_{i1} & \sum_{i=1}^{m} z_i v_{i12}\end{pmatrix}\\ &= \begin{pmatrix}\alpha + \beta & \alpha_1 + \beta_1 & \beta_2\end{pmatrix}\end{aligned}$$

Choices of missing data imputation methods

$$\mathbf{X}'\mathbf{X} = \begin{pmatrix} m & \sum_{i=1}^{n} u_{i1} + \sum_{i=1}^{m} v_{i1} & \sum_{i=1}^{m} v_{i2} \\ \sum_{i=1}^{n} u_{i1} + \sum_{i=1}^{m} v_{i1} & \sum_{i=1}^{n} u_{i1}^2 + \sum_{i=1}^{m} v_{i1}^2 & \sum_{i=1}^{m} v_{i1} v_{i2} \\ \sum_{i=1}^{m} v_{i2} & \sum_{i=1}^{m} v_{i1} v_{i2} & \sum_{i=1}^{m} v_{i2}^2 \end{pmatrix}$$

$$= \begin{pmatrix} m & 0 & 0 \\ 0 & a_1 + b_1 & c \\ 0 & c & b_2 \end{pmatrix},$$

$$\mathbf{X}'\mathbf{Y} = \begin{pmatrix} \sum_{i=1}^{n} y_i + \sum_{i=1}^{m} z_i \\ \sum_{i=1}^{n} y_i u_{i1} + \sum_{i=1}^{m} z_i v_{i1} \\ \sum_{i=1}^{m} z_i v_{i12} \end{pmatrix} = \begin{pmatrix} \alpha + \beta \\ \alpha_1 + \beta_1 \\ \beta_2 \end{pmatrix}.$$

Choices of missing data imputation methods

Hence,

$$\begin{aligned}\mathbf{Y}'\mathbf{H}_x\mathbf{Y} &= \mathbf{Y}'\left(\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\right)\mathbf{Y} = \left(\mathbf{Y}'\mathbf{X}\right)\left(\mathbf{X}'\mathbf{X}\right)^{-1}\left(\mathbf{X}'\mathbf{Y}\right)\\ &= \frac{(\alpha+\beta)^2}{m} + \frac{\left[\alpha_1 b_2 + \beta_1 b_2 - \beta_2 c\right]^2}{b_2\left[a_1 b_2 + b_1 b_2 - c^2\right]} + \frac{\beta_2^2}{b_2}. \qquad (12)\end{aligned}$$

Now, applying Cauchy-Schwarz inequality, we have

$$\frac{\alpha^2}{n} + \frac{\beta^2}{m} \geq \frac{(\alpha+\beta)^2}{m+n}, \qquad (13)$$

$$\begin{aligned}\frac{\alpha_1^2}{a_1} + \frac{(\beta_1 b_2 - \beta_2 c)^2}{\beta_2\left(b_1 b_2 - c^2\right)} &= \frac{1}{b_2}\left[\frac{\alpha_1^2 b_2^2}{a_1 b_2} + \frac{(\beta_1 b_2 - \beta_2 c)^2}{b_1 b_2 - c^2}\right]\\ &\geq \frac{(\alpha_1 b_2 + \beta_1 b_2 - \beta_2 c)^2}{b_2\left(a_1 b_2 + b_1 b_2 - c^2\right)}. \qquad (14)\end{aligned}$$

From (10), (11), (12), (13) and (14), we have
$\mathbf{y}'\mathbf{H}_u\mathbf{y} + \boldsymbol{z}'\mathbf{H}_v\boldsymbol{z} \geq \mathbf{Y}'\mathbf{H}_x\mathbf{Y}$, as desired.

Choices of missing data imputation methods

Factors should be considered when choosing the imputer for ComImp: speed, imputation quality, and the ability to predict sample by sample.

- When the datasets to be combined are small, methods that are slow but can give promising imputation quality, such as kNN, MICE, missForest, can be considered.
- Matrix decomposition methods may not be suitable when the data is not of low rank, and are not suitable for imputation in online learning, which requires the handling of each sample as it comes.
- When the uncertainty of imputed values is of interest, then Bayesian techniques can be used. However, Bayesian imputation techniques can be slow.
- DIMV is a scalable imputation method that estimates the conditional distribution of the missing values based on a subset of observed entries. As a result, the method provides an explainable imputation, along with the confident region, in a simple and straight forward manner.

Experiments

Datasets

Bảng: Descriptions of datasets used in the experiments

Dataset	# Classes	# Features	# Samples
Seed	3	7	210
Wine	3	13	178
Epileptic Seizure	2	178	11,500
Gene	5	20531	801

Experiments

Regression simulation

1. Run a Monte Carlo experiment for regression with 10000 repeats. For each loop, we generate a dataset $\mathcal{D}_1$ of 300 samples, and a dataset $\mathcal{D}_2$ of 200 samples based on the following relation

$$Y = 1 + X_1 + 0.5X_2 + X_3 + \epsilon \tag{15}$$

where $\mathbf{X} = (X_1, X_2, X_3)^T$ follows a multivariate Gaussian distribution with the following mean and covariance matrix

$$\boldsymbol{\mu} = \begin{pmatrix} 1 \\ 2 \\ 0.5 \end{pmatrix}, \boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.5 & 0.3 \\ 0.5 & 1 & 0.4 \\ 0.3 & 0.4 & 1 \end{pmatrix} \tag{16}$$

and $\epsilon \sim \mathcal{N}(0, 0.2\, I)$, where I is the identity matrix.

Experiments

Regression simulation

2. Delete the first feature of $\mathcal{D}_1$ and the second feature of $\mathcal{D}_2$.
3. Added Gaussian noise with variance 0.05 to the second feature and Gaussian noise with variance 0.1 to the third feature of $\mathcal{D}_2$.
4. $\mathcal{D}_1, \mathcal{D}_2$ is split again into training and testing sets ratio 7:3, respectively.

Results

Bảng: mean ± variance of MSE on test sets of regression models fitted on $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}$, respectively.

model	f_1	f_2	f
MSE	3.327 ± 0.093	3.044 ± 0.131	0.734 ± 0.002

Experiments of merging two datasets

Conduct experiments on the Seed dataset and repeat the experiment 1000 times. Follow these steps for each iteration:

1. Delete the first two columns of the input in $\mathcal{D}_1$ and split it into training and testing sets of equal sizes.
2. Delete the last columns of the input data in $\mathcal{D}_2$ and split it into the training and testing sets of equal sizes.
3. Fit models f_1, f_2 on the training set of $\mathcal{D}_1, \mathcal{D}_2$, respectively.
4. Use ComImp to combine the training sets of $\mathcal{D}_1, \mathcal{D}_2$ to the training sets of $\mathcal{D}$, and do similarly for the testing portions
5. Fit a model f on the training set of $\mathcal{D}$.

Experiments

Results

Bảng: mean ± variance of the accuracy on test sets of models fitted on $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}$ of the Seed dataset, respectively.

	test set of $\mathcal{D}_1$	
Classifier	f_1	f
Logistic Regression	0.909 ± 0.001	0.913 ± 0.001
SVM	0.917 ± 0.001	0.925 ± 0.001

	test set of $\mathcal{D}_2$	
Classifier	f_2	f
Logistic Regression	0.858 ± 0.006	0.896 ± 0.003
SVM	0.882 ± 0.003	0.890 ± 0.003

Experiments

Experiments of merging three datasets

We conduct experiments on the Wine dataset.

1. Randomly split the data into three datasets $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$
2. Delete the first column of the input of $\mathcal{D}_1$, the last 8 columns of $\mathcal{D}_2$, and the 5^{th} and 6^{th} column of the input of $\mathcal{D}_3$

Experiments

Results

Bảng: mean ± variance of the accuracy on test sets of models fitted on $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, \mathcal{D}$ of the Wine dataset, respectively.

	test set of $\mathcal{D}_1$	
Classifier	f_1	f
Logistic Regression	0.912 ± 0.004	0.941 ± 0.002
SVM	0.934 ± 0.003	0.955 ± 0.001

	test set of $\mathcal{D}_2$	
Classifier	f_2	f
Logistic Regression	0.735 ± 0.010	0.781 ± 0.007
SVM	0.793 ± 0.007	0.826 ± 0.005

	test set of $\mathcal{D}_3$	
Classifier	f_3	f
Logistic Regression	0.946 ± 0.003	0.971 ± 0.001
SVM	0.953 ± 0.002	0.968 ± 0.001

Experiments

ComImp with transfer learning

We conduct experiments on multi-variate time series datasets related to EEG signals.

1. Randomly split the data into two datasets $\mathcal{D}_1, \mathcal{D}_2$ with 6:4 ratio.
2. Simulate the bad EEG recordings with missing values. Delete the first 16 columns of $\mathcal{D}_1$ and the last 17 columns of $\mathcal{D}_2$.
3. Split $\mathcal{D}_1, \mathcal{D}_2$ into training and testing sets with a 7:3 ratio.

ComImp with transfer learning

To compare our methods with transfer learning, we conduct the following two procedures:

- We train a fully connected two-layered neural network (f_1) on the training portion of $\mathcal{D}_1$. Next, we transfer the weights of f_1 to train a fully connected network f_2 on the training portion of $\mathcal{D}_2$ and fine-tune f_2.
- For our method,
  1. Use ComImp to combine $\mathcal{D}_1$ and $\mathcal{D}_2$, which gives us $\mathcal{D}$ with $\mathcal{D}_{train}$ is the merge between the training portion of $\mathcal{D}_1$ and $\mathcal{D}_2$.
  2. $\mathcal{D}_{test_1}$ corresponding to the testing portion of $\mathcal{D}_1$, and $\mathcal{D}_{test_2}$ corresponding to the testing portion of $\mathcal{D}_2$. We train a model f on $\mathcal{D}_{train}$.
  3. Transfer the weights of f onto the training portion of $\mathcal{D}_1$ and fine-tune the model, which gives us model f_1^{ComImp}.
  4. Do similarly for $\mathcal{D}_2$, which gives us f_2^{ComImp}.

Run the model for 10,000 epochs.

Experiments

Results

Bảng: Comparison of transfer learning and transfer learning with ComImp on the EEG dataset.

	test set of $\mathcal{D}_1$		**test set of $\mathcal{D}_2$**	
Classifier	f_1	f_1^{ComImp}	f_2	f_2^{ComImp}
Transfer Learning	0.774	0.796	0.839	0.849

Experiments

Data imputation performance for missing datasets

We conduct experiments on the Wine dataset.

1. Delete the first two columns of the input in $\mathcal{D}_1$ and split it into training and testing sets of equal sizes.
2. Delete the last two columns of the input data in $\mathcal{D}_2$.
3. Generate missing data at missing rates of $20\%, 40\%, 60\%$ on each training/testing set.
4. Use ComImp to combine the corresponding training and testing sets of $\mathcal{D}_1, \mathcal{D}_2$ to the training and testing sets of $\mathcal{D}$.

For the ComImp approach, the missing values are automatically filled after merging the datasets. For the non-ComImp approach, we use softImpute to impute missing values. Then, we fit a models f_1, f_2, f on the training set of $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}$, respectively.

Experiments

Results

Bảng: mean ± variance of the accuracy on test sets of models fitted on $\mathcal{D}_1$ of the Wine dataset.

		test set of $\mathcal{D}_1$	
missing rate	Classifier	f_1	f
20%	Logistic Regression	0.917 ± 0.001	0.908 ± 0.001
	SVM	0.939 ± 0.001	0.939 ± 0.001
40%	Logistic Regression	0.883 ± 0.002	0.877 ± 0.002
	SVM	0.901 ± 0.001	0.901 ± 0.001
60%	Logistic Regression	0.834 ± 0.002	0.831 ± 0.002
	SVM	0.843 ± 0.002	0.846 ± 0.002
80%	Logistic Regression	0.762 ± 0.004	0.762 ± 0.003
	SVM	0.762 ± 0.004	0.766 ± 0.003

Experiments

Results

Bảng: mean ± variance of the accuracy on test sets of models fitted on $\mathcal{D}_2$ of the Wine dataset.

		test set of $\mathcal{D}_2$	
missing rate	Classifier	f_2	f
20%	Logistic Regression	0.859 ± 0.007	0.874 ± 0.004
	SVM	0.882 ± 0.006	0.893 ± 0.004
40%	Logistic Regression	0.804 ± 0.009	0.837 ± 0.005
	SVM	0.814 ± 0.009	0.846 ± 0.005
60%	Logistic Regression	0.733 ± 0.010	0.783 ± 0.007
	SVM	0.716 ± 0.012	0.779 ± 0.008
80%	Logistic Regression	0.645 ± 0.012	0.712 ± 0.008
	SVM	0.603 ± 0.014	0.692 ± 0.010

Experiments

Combining datasets using dimension reduction

We conduct experiments on the Gene dataset.

1. Split it into $\mathcal{D}_1, \mathcal{D}_2$ with a ratio $7:3$, and then split each of them into halves for training and testing.
2. Delete the first 10,000 columns of the input in $\mathcal{D}_1$ and the last 10,000 columns of the input in $\mathcal{D}_2$.
3. Apply ComImp with PCA and train a neural network f on the merged data.
4. Train a separate model f_1 on $\mathcal{D}_1$, and f_2 on $\mathcal{D}_2$

Repeat the experiment 100 times

Experiments

Results

We conduct experiments on the Gene dataset.

1. Split it into $\mathcal{D}_1, \mathcal{D}_2$ with a ratio $7:3$, and then split each of them into halves for training and testing.
2. Delete the first 10,000 columns of the input in $\mathcal{D}_1$ and the last 10,000 columns of the input in $\mathcal{D}_2$.
3. Apply ComImp with PCA and train a neural network f on the merged data.
4. Train a separate model f_1 on $\mathcal{D}_1$, and f_2 on $\mathcal{D}_2$

Repeat the experiment 100 times

Experiments

Results

Bảng: mean ± variance of the accuracy on test sets of regression models fitted on $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}$ of the Gene dataset, respectively.

	test set of $\mathcal{D}_1$	
Classifier	f_1	f
Logistic Regression	0.998 ± 0.002	0.998 ± 0.002
SVM	0.995 ± 0.007	0.997 ± 0.003

	test set of $\mathcal{D}_2$	
Classifier	f_2	f
Logistic Regression	0.996 ± 0.006	0.996 ± 0.006
SVM	0.955 ± 0.021	0.971 ± 0.012

Experiments

A case analysis of when OsImp may fail

- This experiment is conducted on the Seed dataset with the same setup as in experiments of merging two datasets except that we delete the first three features of $\mathcal{D}_1$ and the last four features of $\mathcal{D}_2$.
- The Seed dataset has only seven features, and therefore, $\mathcal{D}_1$ and $\mathcal{D}_2$ have only one overlapping feature.

Experiments

Results

Bảng: mean ± variance of the accuracy on test sets of models fitted on $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}$ of the Seed dataset, respectively.

	test set of $\mathcal{D}_1$	
Classifier	f_1	f
Logistic Regression	0.912 ± 0.001	0.911 ± 0.001
SVM	0.926 ± 0.001	0.923 ± 0.001

	test set of $\mathcal{D}_2$	
Classifier	f_2	f
Logistic Regression	0.730 ± 0.017	0.675 ± 0.020
SVM	0.827 ± 0.008	0.826 ± 0.010

Thank you for your attention!