Forward Stagewise Regression on Incomplete datasets

Marcelo B. A. Veras[1], Diego P. P. Mesquita[1], João P. P. Gomes[1], Amauri H. Souza Junior[2] and Guilherme A. Barreto[3]

[1] Federal University of Ceará, Department of Computer Science
Fortaleza, Ceará, Brazil
{marcelobveras, diegoparente, jpaulo}@lia.ufc.br
[2] Federal Institute of Ceará, Department of Computer Science
Maracanaú, Ceará, Brazil
amauriholanda@ifce.edu.br
[3] Federal University of Ceará, Department of Teleinformatics Engineering
Fortaleza, Ceará, Brazil
guilherme@deti.ufc.br

Abstract. The Forward Stagewise Regression (FSR) algorithm is a popular procedure to generate sparse linear regression models. However, the standard FSR assumes that the data are fully observed. This assumption is often flawed and pre-processing steps are applied to the dataset so that FSR can be used. In this paper, we extend the FSR algorithm to directly handle datasets with partially observed feature vectors, dismissing the need for the data to be pre-processed. Experiments were carried out on real-world datasets and the proposed method reported promising results when compared to the usual strategies for handling incomplete data.

1 Introduction

Missing data is a common occurrence in many real-world domains that may have a significant effect on the results of machine learning algorithms. Roughly speaking, in the problem of learning from incomplete datasets, a machine learning algorithm has to learn from input vectors where some of its attributes are unknown. Possible reasons for the absence of these attributes are transmission and storage problems, operator failure, measurement error and etc [1].

According to Little and Rubin in [2], understanding the missingness mechanism is fundamental to the task of designing solutions to handle the missing data problem. Missing data mechanisms are usually classified into three main groups: Missing Completely at Random (MCAR), Missing at Random (MAR) and Not Missing at Random (NMAR). In MCAR, the missingness of a component is independent of its real value and any value of other components on the dataset. This characterization is often seen as very restrictive and various authors consider that it is very unlikely in real-world applications [3]. A more realistic approach is the MAR mechanism. In MAR, the missingness of a component is independent of the value itself but can be related to the observed values.

Finally, MNAR characterizes a whole different situation where the instance is not missing at random. In MNAR the missing probability is related to the value of the missing component and handling such problems usually requires a model of the missingness mechanism. In this work, we consider the case where the probability of a component being missing is not related to its value, hence we adopt the less restrictive option, assuming that the missing data is MAR.

Considering the MAR framework, the simplest strategy to handle missing data is the Listwise Deletion (LD). In this method, only fully observed input vectors are used to build the learning model. Although LD is simple and popular, it may lead to poor modeling as the number of vectors with missing components increases [4]. In such cases, a better solution consists in filling the missing components with likely values. The so-called imputation strategies comprise a variety of methods mostly based on either probabilistic models or regression methods [4]. In the probabilistic approach, the vectors in the dataset are assumed to be i.i.d. random variables and inference is carried out to estimate the missing values. The Conditional Mean Imputation (CMI, [5]) is a widely used statistical imputation method in which the missing components are filled according to their expected values given the observed components of the same vector. In general, one can assume the data follow any distribution, being the multivariate normal distribution the most common use.

It is worth noting that, in the context of machine learning, data imputation based methods consist of pre-processing steps, *i.e.*, the learning process only starts when the missing data vectors are filled or deleted. Recently, [6] and [1] propose variants of machine learning methods that can handle missing data directly and thus do not require any pre-processing step. In addition to being elegant solutions, those methods also achieved promising results.

The Forward Stagewise Regression (FSR, [8]) algorithm is a linear regression sparse model. According to Hastie *et. al* [9], there are two main reasons that explain why sparse linear models are preferable to non-sparse ones (e.g., linear models coupled with least-squares estimation). First, sparse models often produce lower variance predictions, and hence good generalization. Second, models with reduced number of nonzero coefficients tend to represent only strong effects of the data, thus eliminating details that may be important to a further analysis. The FSR follows a strategy for constructing a sequence of sparse regression estimates: it starts with all coefficients equal to zero, and iteratively updates the coefficient of the variable that achieves the maximal correlation with the current residual [7].

In this paper we propose a new variant of the FSR algorithm with a built-in mechanism to handle missing data. The proposed model is based on the estimation of the expected correlations between each feature and the vector of residuals at each iteration. To compute the necessary steps, we assume that the data are normally distributed. Results show that our method is able to outperform LD and CMI strategies in various real-world datasets.

The remainder of the paper is organized as follows. Section 2 overviews the FSR algorithm. Section 3 introduces the proposed method to extend the FSR

to incomplete data. Section 4 reports the empirical assessment of the proposal, comparing it to the CMI and LD strategies. Conclusions are given in Section 5.

2 Forward Stagewise Regression

Consider a regression setup in which you are given a set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ of input/output training examples, such that $\mathbf{x}_1, \cdots, \mathbf{x}_N$ are p-dimensional input column vectors and $y_1, \cdots, y_N$ are their respective scalar outputs. Furthermore, define the $N \times p$ matrix $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N]^T$ and the column vector $\mathbf{y} = [y_1, \cdots, y_N]^T$. We assume a linear relationship between the input and output variables (a linear model) of the form:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\theta} + \mathbf{r}, \tag{1}$$

where $\mathbf{r} \in \mathbb{R}^N$ denotes a column vector of residuals and $\boldsymbol{\theta} = [\theta_1, \cdots \theta_p]^T$ represents the parameters of the linear model.

The goal in sparse linear estimation is to provide an estimate $\hat{\boldsymbol{\theta}}$ of the parameters $\boldsymbol{\theta}$ such that the l_2-norm of the residuals is small while having as many as possible entries in $\hat{\boldsymbol{\theta}}$ with values equal to zero. This is usually achieved by the following minimization problem:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}'} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}'\|_2 + \lambda\|\boldsymbol{\theta}'\|_1, \tag{2}$$

where $\|\cdot\|_2$ and $\|\cdot\|_1$ denote the l_2 and l_1 norms, respectively, and we use $\boldsymbol{\theta}'$ to distinguish from the actual parameter vector. This formulation leads to a quadratic programming problem and thus many numerical methods can be used to solve it [8]. Among the various methods, the Forward Stagewise Regression algorithm leads to an approximate solution by means of simple iteractive procedure.

The Forward Stagewise Regression algorithm computes $\hat{\boldsymbol{\theta}}$ by iteratively selecting and increasing the value of one of its coefficients $\hat{\theta}_j$ according to the correlation between $\mathbf{X}_j$ and a vector of residuals $\mathbf{r}$. Henceforth, we use $\mathbf{X}_j$ to denote the jth column of $\mathbf{X}$, that is, $\mathbf{X}_j = [x_{1j}, x_{2j}, \cdots, x_{Nj}]^T$. In other words, $\mathbf{X}_j$ comprises the values of the jth feature of all input points. At the beginning of the FSR, the estimates $\hat{\boldsymbol{\theta}}$ are set to zero so that the vector $\mathbf{r}$ reduces to $\mathbf{y}$. At each iteration, both parameters and residuals are updated. The FSR algorithm is detailed in the following steps:

1. Start with $\hat{\boldsymbol{\theta}}^{(0)} = \mathbf{0}$ and $\mathbf{r}^{(0)} = \mathbf{y}$. In addition, standardize the columns of $\mathbf{X}$ to have zero mean and unit variance.
2. For each iteration $t = 1, 2, \ldots$
3. Find the feature index $j \in \{1, \cdots, p\}$ most correlated with the residual variable at instant $t - 1$ [*].

[*] We are assuming that the vectors $\{\mathbf{x}_i\}$ are realizations of a p-dimensional random variable. Likewise, $\mathbf{r}$ comprises N samples from the residual random variable. We use the method-of-moments estimator for the correlation between jth variable and the residual variable, that is, $\frac{1}{N}\mathbf{X}_j^T\mathbf{r}$.

4. Update the parameter estimate according to:

$$\hat{\theta}_j^{(t)} \leftarrow \hat{\theta}_j^{(t-1)} + \delta_j^{(t)}, \quad \text{such that} \quad \delta_j^{(t)} = \begin{cases} \epsilon, & \text{if } \mathbf{X}_j^T \mathbf{r}^{(t-1)} > 0, \\ -\epsilon, & \text{otherwise.} \end{cases}, \tag{3}$$

where the step-size $\epsilon > 0$ is a pre-defined constant.

5. Update the vector of residuals as follows:

$$\mathbf{r}^{(t)} \leftarrow \mathbf{r}^{(t-1)} - \delta_j^{(t)} \mathbf{X}_j. \tag{4}$$

6. Go back to step 2 until the residuals are uncorrelated with all the predictors.

3 Proposed Method

We now consider the case where some instances of $\mathbf{X}$ have one or more missing entries. We are interested in reformulating the FSR algorithm to handle such case. In this matter, we first need to tackle the problem of estimating the correlation between the j-th feature and the residual variable, i.e., the value of $\mathbf{X}_j^T \mathbf{r}$ when some entries of $\mathbf{X}_j$ and/or $\mathbf{r}$ are missing. Under this scenario, we can consider the missing components of $\mathbf{X}$ as random variables. Thus, in the general case where any entry of $\mathbf{X}$ can be missing, the expected value of the desired correlation is given by

$$\begin{aligned}
\mathbb{E}\big[\mathbf{X}_j^T \mathbf{r}\big] &= \mathbb{E}\big[\mathbf{X}_j^T \mathbf{y} - \mathbf{X}_j^T \mathbf{X}\boldsymbol{\theta}\big] \\
&= \mathbb{E}[\mathbf{X}_j^T \mathbf{y}] - \mathbb{E}[\mathbf{X}_j^T \mathbf{X}\boldsymbol{\theta}] \\
&= \sum_{i=1}^{N} \big(y_i \mathbb{E}[x_{i,j}]\big) - \sum_{i=1}^{N} \big(\mathbb{E}[x_{i,j} \mathbf{x}_i^T \boldsymbol{\theta}]\big) \\
&= \sum_{i=1}^{N} \big(y_i \mathbb{E}[x_{i,j}] - \mathbb{E}[x_{i,j}]\mathbb{E}[\mathbf{x}_i^T \boldsymbol{\theta}] + \mathrm{Cov}[x_{i,j}, \mathbf{x}_i^T \boldsymbol{\theta}]\big) \\
&= \sum_{i=1}^{N} \left(y_i \mathbb{E}[x_{i,j}] - \left(\mathbb{E}[x_{i,j}] \sum_{k=1}^{p} \theta_k \mathbb{E}[x_{i,k}] + \sum_{k=1}^{p} \theta_k \mathrm{Cov}[x_{i,k}, x_{i,j}] \right) \right) \\
&= \sum_{i=1}^{N} \left(y_i \mathbb{E}[x_{i,j}] - \sum_{k=1}^{p} \theta_k \big(\mathbb{E}[x_{i,k}]\mathbb{E}[x_{i,j}] + \mathrm{Cov}[x_{i,j}, x_{i,k}] \big) \right)
\end{aligned} \tag{5}$$

In the missing data scenario, there is uncertainty only on the unobserved/missing entries of $\mathbf{X}$, as the observed values are constants, i.e., $\mathbb{E}[x_{i,j}] = x_{i,j}$ if $x_{i,j}$ is not missing. Likewise, $\mathrm{Cov}[x_{i,j}, x_{i,k}] = 0$ if $x_{i,j}$ or $x_{i,k}$ are not missing.

Eq. (5) expresses the expected correlation as a function of the expected values of the inputs and the covariance between different attributes of the same input vector. Let M_i denote the indices of the unobserved entries of $\mathbf{x}_i$. Furthermore, let $O_i = \{1, \ldots, p\} \setminus M_i$. Thus, the vector $\mathbf{x}_i$ can be divided into two parts $[\mathbf{x}_{i,O_i}, \mathbf{x}_{i,M_i}]$.

We are interested in computing the expected value of $\mathbf{X}_j\mathbf{r}$ conditioned on the observed values of $\mathbf{X}$. For that, as shown in Eq. (5), we need to compute the expected value of, and the covariance between, the missing entries of each training point $\mathbf{x}_i$ conditioned on the observed entries of the same vector, compactly written as $\mathbb{E}[\mathbf{x}_{i,M_i}|\mathbf{x}_{i,O_i}]$ and $\mathrm{Cov}[\mathbf{x}_{i,M_i}|\mathbf{x}_{i,O_i}]$.

According to [1], under the assumption that $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, we can obtain $\mathbb{E}[\mathbf{x}_{i,M_i}|\mathbf{x}_{i,O_i}]$ and $\mathrm{Cov}[\mathbf{x}_{i,M_i}|\mathbf{x}_{i,O_i}]$ as follows:

$$\mathbb{E}[\mathbf{x}_{i,M}|\mathbf{x}_{i,O}] = \boldsymbol{\mu}_M + \boldsymbol{\Sigma}_{MO}\boldsymbol{\Sigma}_{OO}^{-1}(\mathbf{x}_{i,O} - \boldsymbol{\mu}_O), \tag{6}$$

$$\mathrm{Cov}[\mathbf{x}_{i,M}|\mathbf{x}_{i,O}] = \boldsymbol{\Sigma}_{MM} - \boldsymbol{\Sigma}_{MO}\boldsymbol{\Sigma}_{OO}^{-1}\boldsymbol{\Sigma}_{OM}, \tag{7}$$

where we omitted the dependence of i in M and O for simplicity. The subscripts OO, OM, MO and MM refer to the subsets of the full covariance matrix $\boldsymbol{\Sigma}$ between missing and observed variables of $\mathbf{x}_i$. Additional details can be found in [1]

The FSR for incomplete data can be summarized as follows:

1. Start with $\hat{\boldsymbol{\theta}}^{(0)} = \mathbf{0}$ and $\mathbf{r}^{(0)} = \mathbf{y}$. In addition, standardize the columns of $\mathbf{X}$ to have zero mean and unit variance.
2. For each iteration $t = 1, 2, \ldots$
3. Find the feature index j most correlated with the residual variable at instant $t-1$:
$$j = \arg\min_{k=1,\ldots,p} \mathbb{E}[\mathbf{X}_k\mathbf{r}^{(t-1)}|\mathbf{X}_O], \tag{8}$$
where $\mathbf{X}_O$ refers to all pairs of indexes (i, j) at which $x_{i,j}$ is observed.
4. Update the parameter estimate according to:
$$\hat{\theta}_j^{(t)} \leftarrow \hat{\theta}_j^{(t-1)} + \delta_j^{(t)}, \quad \text{such that} \quad \delta_j^{(t)} = \begin{cases} \epsilon, & \text{if } \mathbb{E}[\mathbf{X}_j\mathbf{r}^{(t-1)}|\mathbf{X}_O] > 0, \\ -\epsilon, & \text{otherwise.} \end{cases}. \tag{9}$$
5. Update the vector of residuals as follows:
$$\mathbf{r}^{(t)} \leftarrow \mathbf{r}^{(t-1)} - \delta_j^{(t)}\mathbb{E}[\mathbf{X}_j|\mathbf{X}_O]. \tag{10}$$
6. Go back to step 2 until the residuals are uncorrelated with all the predictors.

Remarkably, Eq. (5) can be written concisely as:

$$\mathbb{E}\big[\mathbf{X}_j\mathbf{r}\big] = \mathbb{E}\big[\mathbf{X}_j\big]^T\mathbf{y} - \mathbb{E}\big[\mathbf{X}_j\big]^T\mathbb{E}\big[\mathbf{X}\big]\boldsymbol{\theta} - \sum_{i=1}^{N}\mathbf{e}_j^T\mathrm{Cov}[\mathbf{x}_i]\boldsymbol{\theta} \tag{11}$$

where $\mathbf{e}_j$ is the jth vector of the canonical basis of $\mathbb{R}^p$ and $\mathrm{Cov}[\mathbf{x}_i]$ is the full covariance matrix of the example $\mathbf{x}_i$ (which is the covariance matrix of the missing entries padded with zeros in the components related to the observed entries).

Therefore, one can conclude that the proposed method differs from common imputation strategies as the last term in Eq. (11) takes into account the uncertainty concerning the missing data entries.

4 Performance Evaluation

To asses the performance of the proposed method, named Forward Stagewise Regression for Incomplete datasets (FSRI), we carried out a set of experiments with 5 arbitrary real-world datasets, available at [10]. We compare FSRI to standard methods used to handle missing data. For each dataset, we varied the amount of inputs with missing variable from 10% to 50%. The description of the datasets is presented in Table 1.

Table 1. Datasets description.

	# Features	# Training samples	# Test samples
Wine	13	100	78
CPU	9	139	70
Cancer	32	129	65
Automobile Price	15	106	53
Forest Fire	4	344	173

The FSRI was compared to the Listwise Deletion (LD) and the Conditional Mean Imputation (CMI). Both CMI and LD were used as pre-processing steps and the standard FSR was used to generate the linear models. For FSRI and CMI, we estimate the parameters of the data distribution using the Expectation Conditional Maximization (ECM, [5]) algorithm for datasets with missing values.

All methods were compared based on two criteria, the Mean Square Error (MSE) between $\mathbf{y}$ and the results of each model and Mean Squared Difference between the Coefficients (MSDC) of each linear model and the linear model obtained by a FSR on the same dataset without missing values. All experiments were repeated 500 times. Table 2 shows the average MSE obtained in the experiments.

Beforehand, it is important to clarify that some of the LD results are not filled which indicates that the FSR algorithm was not able to converge due to the significant number of discarded examples. Concerning the other AMSE values, one can see that the average MSE for all methods increase as the number of missing data increases. However, it is noticeable that FSRI had the lowest AMSE for all datasets and missing data percentages. This performance gap is even more significant in the experiments with the highest number of missing data.

Along with the MSE, we computed the MSDC metric to quantify the difference between the linear model generated by each method and an ideal linear model obtained by a FSR on a complete (no missing data) dataset. We decided to compare the methods on several instants during the learning process. The instants are defined according to the norm of the weights generated by each method. We considered the norm obtained by the FSR in the complete dataset as the maximum norm and evaluated all method at 3 different ratios of this norm. Such procedure was adopted to provide a fair comparison since different

methods show weight vectors with varying norms at each iteration. Tables 3, 4 and 5 show the MSDC values for the ratios 0.3, 0.45 and 0.6.

As can be noticed, the difference between the weight vectors generated by each method and the ideal linear model increased with the number of missing data. Once again FSRI had the best overall performance being less sensible to the presence of missing data.

Table 2. Average MSE between the outputs of each linear model and the target outputs. The number of input vectors with missing entries varies from 10% to 50%.

Wine					
	10%	20%	30%	40%	50%
FSRI	6.3404	6.5319	6.7531	7.3600	8.1164
CMI	6.6190	7.5510	10.7476	22.9362	54.9262
LD	12.0314	23.6968	35.8139	-	-
CPU					
	10%	20%	30%	40%	50%
FSRI	2.7838e+05	2.8849e+05	2.9230e+05	3.2206e+05	3.2950e+05
CMI	2.8381e+05	3.0676e+05	3.4010e+05	4.5708e+05	6.2148e+05
LD	3.3261e+05	4.1271e+05	7.8900e+05	1.3920e+06	-
Automobile Price					
	10%	20%	30%	40%	50%
FSRI	4.2238e+08	4.2386e+08	4.1773e+08	4.2275e+08	4.1695e+08
CMI	4.4615e+08	4.9690e+08	7.7811e+08	1.8076e+09	3.4648e+09
LD	1.5076e+09	1.1875e+09	-	-	-
Cancer					
	10%	20%	30%	40%	50%
FSRI	8.5119e+04	8.2159e+04	8.0475e+04	7.9577e+04	7.7541e+04
CMI	9.2285e+04	9.4305e+04	9.9716e+04	1.1450e+05	1.6995e+05
LD	1.4298e+05	-	-	-	-
Forest-Fire					
	10%	20%	30%	40%	50%
FSRI	6.4320e+05	6.7355e+05	6.8613e+05	6.9128e+05	7.0489e+05
CMI	6.4374e+05	6.7554e+05	6.9017e+05	6.9693e+05	7.1871e+05
LD	6.4787e+05	7.2750e+05	9.2299e+05	1.2284e+06	1.6003e+06

Table 3. Average MSDC between each linear model and the linear model obtained by a FSR on the same dataset. In this experiment we set 0.30 of the maximum norm as the comparison point

Wine					
	10%	20%	30%	40%	50%
FSRI	0.0028	0.0060	0.0114	0.0168	0.0297
CMI	0.0031	0.0077	0.0151	0.0232	0.0402
LD	0.0345	0.1261	0.2264	0.2236	0.2635
CPU					
	10%	20%	30%	40%	50%
FSRI	949.5511	1642.1833	2362.9053	2609.0817	2905.4348
CMI	994.5057	1767.8236	2581.6698	2890.8464	3208.9734
LD	2858.6407	4416.8752	4713.3837	5112.4878	5491.9145
Automobile Price					
	10%	20%	30%	40%	50%
FSRI	4.99136e+05	7.89447e+05	1.37572e+06	1.64525e+06	2.34146e+06
CMI	5.38478e+05	9.24256e+05	1.65136e+06	2.06429e+06	3.04505e+06
LD	6.07145e+06	1.43461e+07	1.09962e+07	1.48738e+07	2.57570e+07
Cancer					
	10%	20%	30%	40%	50%
FSRI	315.5959	689.3359	879.4547	939.1615	941.2634
CMI	321.0407	714.4983	984.0727	1214.7930	1433.9887
LD	2738.7149	-	-	-	-
Forest-Fire					
	10%	20%	30%	40%	50%
FSRI	1.3064	2.6853	3.1192	4.2199	5.1891
CMI	1.3349	2.9743	3.6451	5.192	6.3540
LD	1.3788	3.7616	7.3096	10.2142	12.6646

Table 4. Average MSDC between each linear model and the linear model obtained by a FSR on the same dataset. In this experiment we set 0.45 of the maximum norm as the comparison point

Wine					
	10%	20%	30%	40%	50%
FSRI	0.0033	0.0065	0.0125	0.0194	0.0348
CMI	0.0032	0.0087	0.0179	0.0283	0.0520
LD	0.0425	0.1681	0.3012	0.3126	0.2575
CPU					
	10%	20%	30%	40%	50%
FSRI	1277.7359	2385.4040	3638.0817	4105.8572	4838.1707
CMI	1370.3084	2707.6797	4261.6337	4913.7103	5758.5741
LD	4757.6309	7546.5486	8578.5215	9597.8471	11329.4762
Automobile Price					
	10%	20%	30%	40%	50%
FSRI	6.39862e+05	1.08062e+06	1.88349e+06	2.19982e+06	3.00759e+06
CMI	6.97030e+05	1.26571e+06	2.29862e+06	2.74333e+06	3.96610e+06
LD	7.66190e+06	1.52016e+07	2.54035e+07	2.24063e+07	-
Cancer					
	10%	20%	30%	40%	50%
FSRI	815.4529	1432.2428	1835.1505	1571.4583	1542.1451
CMI	859.2785	1783.3939	2393.9583	2772.8872	3309.5005
LD	5932.4867	-	-	-	-
Forest-Fire					
	10%	20%	30%	40%	50%
FSRI	1.8247	3.6323	4.1453	5.2285	7.4526
CMI	1.8937	4.2270	5.4768	8.0466	10.3822
LD	2.3055	6.5959	13.2102	18.6168	23.3338

Table 5. Average MSDC between each linear model and the linear model obtained by a FSR on the same dataset. In this experiment we set 0.60 of the maximum norm as the comparison point

Wine					
	10%	20%	30%	40%	50%
FSRI	0.0033	0.0084	0.0169	0.0262	0.0426
CMI	0.0037	0.0104	0.0221	0.0350	0.0630
LD	0.0511	0.1926	0.3183	0.5117	-
CPU					
	10%	20%	30%	40%	50%
FSRI	1191.6427	2369.9071	3796.9097	4585.2540	5591.5145
CMI	1313.4053	2841.6674	4745.0750	5810.1135	7133.1241
LD	5174.1832	8978.9877	11144.0691	12598.5600	18165.6937
Automobile Price					
	10%	20%	30%	40%	50%
FSRI	8.20843e+05	1.55785e+06	2.47574e+06	3.08841e+06	3.95091e+06
CMI	8.92548e+05	1.80668e+06	3.03822e+06	3.66187e+06	5.14811e+06
LD	9.34456e+06	1.83650e+07	2.34479e+07	-	-
Cancer					
	10%	20%	30%	40%	50%
FSRI	1471.5130	2002.8079	2187.5142	2280.6016	1995.2143
CMI	1561.4253	2983.8447	4298.5120	4711.2356	5891.7460
LD	7616.6000	-	-	-	-
Forest-Fire					
	10%	20%	30%	40%	50%
FSRI	2.0451	4.1852	4.8350	5.6707	9.1165
CMI	2.1624	5.2306	6.8008	10.2454	14.4056
LD	3.2613	9.3837	20.2777	28.8248	36.5733

5 Conclusions

In this paper we proposed a variant of the Forward Stagewise Regression algorithm for incomplete datatsets. In the proposed method, named FSRI, we considered the inputs as normally distributed random variables and modified the steps of FSR such that weights are incremented according to the expected correlation of the residuals and each of the features. FSRI was compared to popular strategies to handle missing values and achieved promising results.

It is worth highlighting that the performance of FSRI can be significantly degraded if the normality assumption of the training set does not hold. Hence we are currently working to extend the FSRI formulation for non-Gaussian datasets using nonparametric/semi-parametric models

Acknowledgments

The authors acknowledge the support of CNPq (Grant 456837/2014-0 and research fellowship).

References

1. Emil Eirola, Gauthier Doquire, Michel Verleysen and Amaury Lendasse: Distance estimation in numerical data sets with missing values. Information Sciences. 240, 115–128, 2013

2. Roderick J.A. Little, Donald B. Rubin: Statistical Analysis with Missing Data. Wiley Interscience. 2nd ed, 2002
3. Yufeng Ding and Jeffrey S. Simonoff: An Investigation of Missing Data Methods for Classification Trees Applied to Binary Response Data. Journal of Machine Learning Research. 11, 131–170, 2010
4. Garcia-Laencina, Pedro J. and Sancho-Gomez, Jose-Luis and Figueiras-Vidal, Anibal R.: Pattern Classification with Missing Data: A Review. Neural Computing and Applications. 19, 263–282, 2010
5. Meng, Xiao-Li and Donald B. Rubin : Maximum Likelihood Estimation via the ECM Algorithm Biometrika, 1993, **80**, pp. 267 - 278
6. Lluís A. Belanche and Vladimer Kobayashi and Tomás Aluja : Handling missing values in kernel methods with application to microbiology data Neurocomputing, 2014, **141**, pp. 110 - 116
7. Tibshirani, Ryan J. : A General Framework for Fast Stagewise Algorithms Journal of Machine Learning Research, 2015, **16**, pp. 2543 - 2588
8. Trevor Hastie, Jonathan Taylor, Robert Tibshirani, Guenther Walther : orward stagewise regression and the monotone lasso Electronic Electronic Journal of Statistics, 2007, **1**
9. Hastie, T. and Tibshirani, R. and Friedman, J: The Elements of Statistical Learning. Springer New York Inc. 2nd ed, 2009
10. Frank A., Asuncion A.: UCI Machine Learning Repository University of California, Irvine, School of Information and Computer Sciences, 2010