Missing Value Learning

Zhi-Lin Zhao, Chang-Dong Wang, Kun-Yu Lin, Jian-Huang Lai
School of Data and Computer Science, Sun Yat-sen University
Guangzhou, China 510006
XinHua College, Sun Yat-sen University
Guangzhou, China 510520
zhaozhl7@mail2.sysu.edu.cn,changdongwang@hotmail.com,kunyulin14@outlook.com,stsljh@mail.sysu.edu.cn

ABSTRACT

Missing value is common in many machine learning problems and much effort has been made to handle missing data to improve the performance of the learned model. Sometimes, our task is not to train a model using those unlabeled/labeled data with missing value but process examples according to the values of some specified features. So, there is an urgent need of developing a method to predict those missing values. In this paper, we focus on learning from the known values to learn missing value as close as possible to the true one. It's difficult for us to predict missing value because we do not know the structure of the data matrix and some missing values may relate to some other missing values. We solve the problem by recovering the complete data matrix under the three reasonable constraints: feature relationship, upper recovery error bound and class relationship. The proposed algorithm can deal with both unlabeled and labeled data and generative adversarial idea will be used in labeled data to transfer knowledge. Extensive experiments have been conducted to show the effectiveness of the proposed algorithms.

CCS CONCEPTS

•**Information systems** →**Data cleaning;** *Information extraction;*

KEYWORDS

Missing Value; Unsupervised Learning; Supervised Learning; Generative Adversarial

ACM Reference format:
Zhi-Lin Zhao, Chang-Dong Wang, Kun-Yu Lin, Jian-Huang Lai. 2017. Missing Value Learning. In *Proceedings of The 26th ACM International Conference on Information and Knowledge Management, Pan Pacific, Singapore, Nov. 2017 (CIKM'17),* 4 pages.
DOI: 10.475/123_4

CIKM'17, Pan Pacific, Singapore

DOI: 10.475/123_4

1 INTRODUCTION

Missing value is common in many machine learning problems such as classification, clustering and regression. For missing values, we can replace with mean, median and values estimated by Matrix Completion (MC) [8] or Multiple Imputation by Chained Equations (MICE) [12] before using the data to train a model. Alternatively, we can remove the examples with missing values or leave the missing values. But sometimes, we want to make a statistics according to the values of some specified features. For example, if there are records of some cardiac patients and we want to know the resting blood pressure of each patients but some of them are missing. Furthermore, if the iris data with label is incomplete and we want to reclassify iris according to the length of sepal, we have to recover the length of sepal for the examples which have missed the feature. So missing value learning is very important to further analyze data. But it is still a challenge to learn the missing values from other known data precisely due to the lack of knowledge about the structure of the data matrix as well as the complex dependencies of features.

Matrix Completion is a technology to recover missing values from very limited information and widely used in rating prediction for recommendation algorithms [7]. But matrix completion technology strongly relies on an assumption that the recovered data matrix is low-rank or approximately low-rank. Without this assumption, the matrix will be ill-posed [1], while there is no theoretical interpretation why the matrix should be low-rank. Besides, the algorithm is time-consuming and can not work well if the data matrix is not low-rank because the assumption is incorrect. In order to improve the efficiency and the effectiveness of matrix completion, matrix factorization, e.g., Singular Value Decomposition (SVD) [11] decomposes the data matrix into two low-rank latent feature matrixes and recover the data matrix by the product of the two matrixes but it has the same shortage as matrix completion.

Relationship exists between features and we can learn a missing value according to other features by linear regression. But if examples have more than one missing value, the simple linear regression may not work because the input data is missing for the regression. In order to overcome this issue, MICE initializes the missing values with expected values. Then it uses the data to train several regression models (The number of model is equal to the number of features.) and replaces the missing values with the predictions of those models. These updates are repeated a number of times, as specified by the maximum number of iterations. This

method needs to train a group of regression models several times separately and it's not an end-to-end model. So it can not automatically adapt to data well and the integrating degree is not enough. Although the model is very simple, it lacks formal theoretical justification. Ignoring the missing values, Sparse Linear Methods (SLIM) [9] self-represents raw data by regressing and it can not predict missing value well because it just fits the raw data.

In this paper, we focus on learning the missing values as close as possible to the true ones and proposed an end-to-end missing value learning model. The model can predict the missing values for both unlabeled and labeled data which can adapt to more situations. The proposed algorithm can map the data with missing values to the complete data directly and does not need to impose structural assumptions on the data. We recover the complete data matrix under the three reasonable constraints: feature relationship, upper recovery error bound and class relationship. In order to deal with the problem of multiple missing values, all the operations are performed on the complete data which will be adjusted automatically rather than the raw data which is quite different from MICE and MC. For the labeled data, generative adversarial [6] idea is used to transfer the knowledge of label information to better learn the missing values.

2 MISSING VALUE LEARNING

Let $R \in \mathbb{R}^{m\times d}$ be the raw data matrix with missing values where m and d represent the number of examples and features respectively. We denote $P \in \mathbb{R}^{m\times d}$ and $W \in \mathbb{R}^{d\times d}$ as the complete data matrix and the weight matrix.

2.1 Algorithm Overview

The proposed algorithm begins with an initialization of a complete data matrix P and constantly updates the matrix under three constraints:

1) Feature relationship: we assume each feature is related to other features and use regression to learn each features. So the data matrix is self-representable. Because the raw data matrix has missing values so the regression is performed on P which will be learned rather than R. But a feature may just be related to few features, so the weight matrix W may be sparse.

2) Upper recovery error bound: P is the recovered matrix of R, so the known elements in R should be close to the corresponding elements in P and we should set a upper error bound between P and R.

3) Class relationship: with label information, we can use it to further improve the quality of missing value learning by transfer learning. Based on the idea of K-Means [3], we assume that the learned examples in P are good if those examples are close to the centre of the class they belong to.

If the data do not contain label information, we can get a Unsupervised Missing Value learning (UMVL) model $\mathcal{G}(P, W)$ according the first two constraints. And $\mathcal{G}(P, W)$ can be viewed as a generator which can generate P which can be self-representable and is close to the raw data. By the third constraint, we can get a discriminator $\mathcal{D}(P)$ which can discriminate whether the examples in P of the same class will get together. So, the idea of generative adversarial can be used in Supervised Missing Value learning (SMVL) and achieve the transfer of knowledge from label information by fusing the generator $\mathcal{G}(P, W)$ and the discriminator $\mathcal{D}(P)$.

2.2 Objective Function

According to the feature relationship, the self-represention error of P is,

$$\mathcal{F}(P, W) = \frac{1}{2}\|P - PW\|_F^2, \tag{1}$$

where the diagonal elements of W should be zeros ($diag(W) = 0$). In order to improve generalization and stability, we add a Frobenius norm term of W. Besides, L1-norm will be added to make W sparse.

Considering the upper recovery error bound, we can learn P by,

$$\mathcal{G}(P, W) = \lambda_1\|W\|_1 + \lambda_2\|W\|_F^2 + \mathcal{F}(P, W), \tag{2}$$

with the constraint that,

$$\|(P - R) \odot A\|_F \leq \varepsilon, \tag{3}$$

where λ_1 and λ_2 are the regularization coefficients, and $\odot$ denotes the Hadamard product. A is an indicator matrix with the element being equal to 0 if the corresponding element in R is missing or 1 otherwise.

By the third constraint,

$$\mathcal{D}(P) = \frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\|P(i,:) - \frac{1}{n_k}\sum_{j=1}^{n_k}P(j,:)\|_F^2 = \frac{1}{2}\|P - GP\|_F^2, \tag{4}$$

where K is the number of classes. $G_{ij} = \frac{1}{n_k}$ if both example i and j are in the k^{th} class or 0 otherwise and n_k is the number of examples in the class.

Based on the idea of generative adversarial, the objective function of SMVL is,

$$\begin{aligned} \min \quad & \mathcal{G}(P, W) + \alpha\mathcal{D}(P), \\ \text{s.t.} \quad & \|(P - R) \odot A\|_F \leq \varepsilon, \quad \text{diag}(W) = 0, \end{aligned} \tag{5}$$

where α is a coefficient for controlling the influence of label information. And UMVL can be viewed as a special case of SMVL when $\alpha = 0$.

2.3 Discussion

1) If we use R to train W by $\min \|R - RW\|_F^2$ and predict P by $P = XW$, the performance will be poor because W is just used to fit the known values in R.

2) Although the objective function of UMVL can be written as $\mathcal{Z}(P, W) = \mathcal{G}(P, W) + \lambda\|(P - R) \odot A\|_F$, it's difficult to control the square error between R and P. The R may be quite different from P even if the value of $\mathcal{Z}(P^*, W^*)$ is very small and the prediction will be poor. So we want to find the weight W under the condition that the known elements in R are close to the corresponding elements in P.

3) If we constrain P strictly by $\|(P - R) \odot A\|_F = 0$, the objective function may degenerate to $P^* = R$ and the missing

values can not be learned. Another advantage to relax the upper bound is that it can alleviate the influence of the noisy data in R and prevent overfitting.

4) The coefficient α can not be set too large because it will make all the missing values of a feature tend to take the mean of the feature.

2.4 Optimization

We use the Lagrangian multiplier method [4] and proximal method [10] to optimize the objective function. We rewrite the function in the conic form,

$$\begin{aligned} &\min \quad \mathcal{G}(P,W)+\alpha\mathcal{D}(P), \\ &\text{s.t.} \quad \begin{bmatrix}(P-R)\odot A\\ \varepsilon\end{bmatrix}\in\mathcal{K}, \quad \text{diag}(W)=0, \end{aligned} \tag{6}$$

where $\mathcal{K}=\left\{\begin{bmatrix}X\\ t\end{bmatrix} : \|X\|_F \le t\right\}$ is the second-order cone and self-dual. And it is the traditional conic constraint problem [2], which can be solved by using the projection algorithm on conic. The Lagrangian is given by,

$$\begin{aligned} \mathcal{L} = \lambda_1(\|W\|_1+(\frac{2\lambda_2}{\lambda_1})\frac{1}{2}\|W\|_F^2)+\frac{1}{2}\|P-PW\|_F^2 \\ + < Y,(P-R)\odot A > -s\varepsilon+\frac{\alpha}{2}\|P-GP\|_F^2, \end{aligned} \tag{7}$$

where $\mathcal{H}(W)=\|W\|_1+(\frac{2\lambda_2}{\lambda_1})\frac{1}{2}\|W\|_F^2$ is *elastic net regularization*, Y and s are Lagrange multipliers.

2.4.1 Optimize P. We can update P by gradient descent,

$$P^{t+1}=P^t-\mu\nabla\mathcal{L}(P^t), \tag{8}$$

where μ is the learning rate and the gradient is

$$\begin{aligned} \nabla\mathcal{L}(P^t) = P^t(I_d-W^t)(I_d-W^t)^T \\ +\alpha(I_m-G)^T(I_m-G)P^t+y^t\odot A. \end{aligned} \tag{9}$$

2.4.2 Optimize W. We consider a quadratic model to approximate $\mathcal{F}(P,W)$,

$$\begin{aligned} \mathcal{Q}(W,W^t,P^t)=\mathcal{F}(P^t,W^t)+ < W-W^t,\nabla\mathcal{F}(W^t) > \\ +\frac{1}{2\mu}\|W-W^t\|_F^2, \end{aligned} \tag{10}$$

where the gradient is,

$$\nabla\mathcal{F}(W^t)=(P^t)^T(P^tW^t-P^t). \tag{11}$$

So we can update W by,

$$W^{t+1}=J-diag(J), \tag{12}$$

where

$$\begin{aligned} J &= \arg\min_W\Big(\lambda_1\mathcal{H}(W)+\mathcal{Q}(W,W^t,P^t)\Big) \\ &= \lambda_1\mathcal{H}(W)+\frac{1}{2\mu}\|W-(W^t-\mu\nabla\mathcal{F}(W^t))\|_F^2 \\ &= \mathbf{prox}_{\mu\lambda_1\mathcal{H}(W)}\big(W^t-\mu\nabla\mathcal{F}(W^t)\big) \\ &= \frac{1}{1+2\mu\lambda_1}\mathbf{prox}_{\mu\lambda_1\|\cdot\|_1}\big(W^t-\mu\nabla\mathcal{F}(W^t)\big) \\ &= \frac{1}{1+2\mu\lambda_1}\mathcal{S}_{\mu\lambda_1}\big(W^t-\mu\nabla\mathcal{F}(W^t)\big), \end{aligned} \tag{13}$$

and **prox** is proximal operator and $\mathcal{S}_\phi(x)$ is soft-thresholding (shrinkage) operator,

$$\mathcal{S}_\phi(x)=\begin{cases} x-\phi, & x>\phi \\ 0, & |x|\le\phi \\ x+\phi, & x<-\phi \end{cases}. \tag{14}$$

2.4.3 Optimize y and s. We can update the two lagrange multiplier y and s by,

$$\begin{bmatrix}Y_{k+1}\\ s_{k+1}\end{bmatrix}=\mathcal{P}_\mathcal{K}\left(\begin{bmatrix}Y_k\\ s_k\end{bmatrix}+\delta\begin{bmatrix}(P^t-R)\odot A\\ -\varepsilon\end{bmatrix}\right), \tag{15}$$

where δ is step length and $\mathcal{P}_\mathcal{K}$ is the orthogonal projection onto $\mathcal{K}$ which is given by [5],

$$\mathcal{P}_\mathcal{K}\begin{bmatrix}X\\ t\end{bmatrix}=\begin{cases}\begin{bmatrix}X\\ t\end{bmatrix}, & \|X\|_F\le t \\ \frac{\|X\|_F+t}{2\|X\|_F}\begin{bmatrix}X\\ \|X\|_F\end{bmatrix}, & -\|X\|_F\le t\le\|X\|_F \\ \begin{bmatrix}0\\ 0\end{bmatrix}, & t\le-\|X\|_F\end{cases}. \tag{16}$$

3 EXPERIMENT

3.1 Data set and Evaluation

Nine data sets from UCI Machine Learning Repository[1] will be used: Breast Cancer Wisconsin (BCW), Connectionist Bench (CB), Ecoli, Image Segmentation (IM), Iris, Leaf, Multiple Features Data Set (Mfeat), Parkinson Speech (PS), and Vertebral Column (VC). Because the range of each feature will be quite different, so normalization processing is carried out to better measure the quality of learned missing values. For each data set, we remove 20% data randomly as the test set and leave the rest of 80% as the training set.

For evaluation measures, we use Root Mean Square Error (RMSE) which is widely used in the rating prediction in recommendation algorithms to measure the accuracy of the learned missing values,

$$RMSE=\sqrt{\frac{1}{|\Omega_T|}\sum_{i,j\in\Omega_T}(p_{ij}-r_{ij})^2}, \tag{17}$$

where Ω_T denotes the set of missing values.

3.2 Parameter Analysis

Due to lack of space, we choose three typical data sets (Iris, BCW and IS: The number of features is incremental.) to analysis the parameters. Without loss of generality, we assume $\varepsilon=\gamma|R|$ where γ can be viewed as the average error of each known value.

As shown in the left of Figure 1, λ_1 should be increased to make the weight matrix W more sparse as the number of features increases. This is because each feature may relate to few other features and sparse weight matrix can model this situation. And this situation will be more obvious when

[1] http://archive.ics.uci.edu/ml

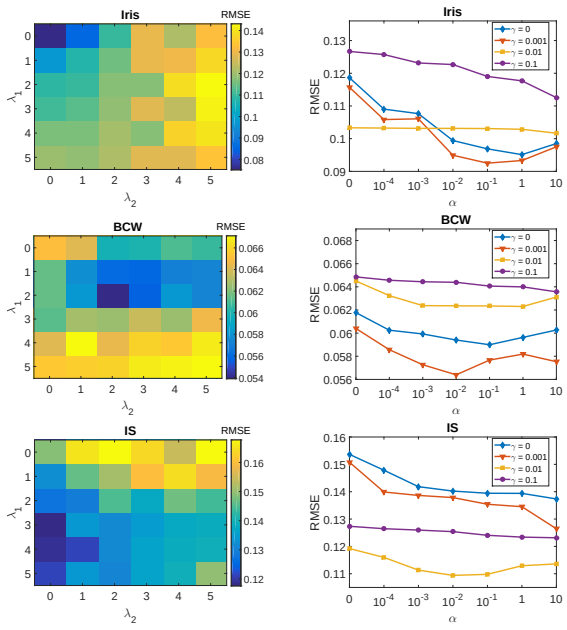

Figure 1: Parameter analysis on the three data sets.

there are a large amount of features. It seems that λ_2 has less influence but it still can improve the quality in some data sets because it can prevent overfitting and make all the values of W small to prevent ill-conditioned.

As shown in the right of Figure 1, as α increases, more knowledge of label information will be transferred to improve the learning quality. So the performance will be enhanced. In some cases, the performance will decrease after arriving the optimal value because a high value α will reduce the self-represented capacity of P. If strictly constraining the error between known values and learned values by setting $\gamma = 0$, we can not get the best result by adjusting α but the relaxed error bound can lead to the best learning result. Because the raw data may contain some noisy data, perfectly fitting of known data may lead to a poor result of recovery. However, if the error bound ε is too large, the result may be worse, because we even can not make sure the recovery error on known values, not to say the missing values. So the error recovery bound should be neither tight nor loose.

3.3 Comparsion Experiments

We compare the quality of the learned missing values by the proposed UMVL and SMVL algorithms with five algorithms, i.e., MEAN (replace the missing values with the mean of the feature), SLIM [9], MC [8], SVD [11], MICE [12]. The comparison results are shown in Table 1. The performance of SLIM is very poor and even worse than MEAN. Although the algorithm is based on regression, it fits the known values by weight matrix and do not consider the missing values. The performance of MC is not good due to the low-rank assumptions which may be wrong. For an improved method, SVD is slightly better than MC. MICE can work well but the proposed UMVL is better because UMVL is an end-to-end model which can better control the learning results according to the raw data. Using the label information, SMVL can further enhance the effect of UMVL.

Table 1: The comparison results on the nine data sets of the seven algorithms.

Data set	MEAN	SLIM	MC	SVD	MICE	UMVL	SMVL
BCW	0.1284	0.1636	0.0959	0.0889	0.0766	0.0604	**0.0564**
CB	0.1748	0.1330	0.1179	0.1171	0.0724	0.0685	**0.0605**
Ecoli	0.1746	0.1905	0.1641	0.1447	0.1385	0.1247	**0.1148**
IM	0.2310	0.2136	0.1413	0.1558	0.1315	0.1192	**0.1094**
Iris	0.1739	0.2681	0.1665	0.1355	0.1357	0.1035	**0.0925**
Leaf	0.1934	0.1907	0.0968	0.0857	0.0838	0.0669	**0.0605**
Mfeat	0.1808	0.3382	0.1716	0.1610	0.1230	0.1115	**0.1062**
PS	0.1545	0.1207	0.1182	0.1169	0.1052	0.1043	**0.0917**
VC	0.1498	0.1817	0.1232	0.1222	0.1150	0.1021	**0.0958**

4 CONCLUSION

Focusing on learning the missing values from the known data as accurately as possible, we have proposed an end-to-end algorithm. We recover the complete data matrix under the three constraints: feature relationship, upper recovery error bound and class relationship. Both unlabeled and labeled data are applicable for our algorithm and the idea of generative adversarial is used when considering the label information. Extensive experiments on nine real-world data sets have confirmed that the proposed algorithm can learn the missing values accurately.

ACKNOWLEDGMENT

This work was supported by Guangdong Natural Science Funds for Distinguished Young Scholar (2016A030306014) and Tip-top Scientific and Technical Innovative Youth Talents of Guangdong special support program (2016TQ03X542).

REFERENCES

[1] Sonia A. Bhaskar. 2015. Probabilistic low-rank matrix recovery from quantized measurements: Application to image denoising. In *ACSSC*. 541–545.
[2] Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen. 2010. A Singular Value Thresholding Algorithm for Matrix Completion. *SIAM Journal on Optimization* 20, 4 (2010), 1956–1982.
[3] Chris H. Q. Ding, Tao Li, Wei Peng, and Haesun Park. 2006. Orthogonal nonnegative matrix t-factorizations for clustering. In *KDD*. 126–135.
[4] Ehsan Elhamifar and René Vidal. 2013. Sparse Subspace Clustering: Algorithm, Theory, and Applications. *IEEE Trans. Pattern Anal. Mach. Intell* 35, 11 (2013), 2765–2781.
[5] Masao Fukushima, Zhi-Quan Luo, and Paul Tseng. 2002. Smoothing Functions for Second-Order-Cone Complementarity Problems. *SIAM Journal on Optimization* 12, 2 (2002), 436–460.
[6] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Networks. In *CoRR*.
[7] Zhao Kang, Chong Peng, and Qiang Cheng. 2016. Top-N Recommender System via Matrix Completion. In *AAAI*. 179–185.
[8] Raghunandan H. Keshavan, Andrea Montanari, and Sewoong Oh. 2010. Matrix completion from a few entries. *IEEE Trans. Information Theory* 56, 6 (2010), 2980–2998.
[9] Xia Ning and George Karypis. 2011. SLIM: Sparse Linear Methods for Top-N Recommender Systems. In *ICDM*. 497–506.
[10] Neal Parikh and Stephen P. Boyd. 2014. Proximal Algorithms. *Foundations and Trends in Optimization* 1, 3 (2014), 127–239.
[11] Arkadiusz Paterek. 2007. Improving regularized singular value decomposition for collaborative filtering. In *Kdd Cup & Workshop*. 39–42.
[12] White I R, Royston P, and Wood AM. 2011. Multiple imputation using chained equations: Issues and guidance for practice. In *Statistics in Medicine*, Vol. 30. 377–99.