Imputing Various Incomplete Attributes via Distance Likelihood Maximization

Shaoxu Song
BNRist, School of Software, Tsinghua University
Beijing, China
sxsong@tsinghua.edu.cn

Yu Sun
BNRist, School of Software, Tsinghua University
Beijing, China
sy17@mails.tsinghua.edu.cn

ABSTRACT

Missing values may appear in various attributes. By "various", we mean (1) different types of values in a tuple, such as numerical or categorical, and (2) different attributes in a tuple, either the dependent or determinant attributes of regression models or dependency rules. Such varieties unfortunately prevent the imputation performing. In this paper, we propose to study the distance models that predict distances between tuples for missing data imputation. The immediate benefits are in two aspects, (1) uniformly processing and collaboratively utilizing the distances on all the attributes with various types of values, and (2) rather than enumerating the combinations of imputation candidates on various attributes, we can directly calculate the most likely distances of missing values to other complete ones and thus infer the corresponding imputations. Our major technical highlights include (1) introducing the imputation statistically explainable by the likelihood on distances, (2) proving NP-hardness of finding the maximum likelihood imputation, and (3) devising the approximation algorithm with performance guarantees. Experiments over datasets with real missing values demonstrate the superiority of the proposed method compared to 11 existing approaches in 5 categories. Our proposal improves not only the imputation accuracy but also the downstream applications such as classification, clustering and record matching.

CCS CONCEPTS

• **Information systems → Data cleaning**.

KEYWORDS

Data Imputation; Incomplete Data; Distance Likelihood

ACM Reference Format:
Shaoxu Song and Yu Sun. 2020. Imputing Various Incomplete Attributes via Distance Likelihood Maximization. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3394486.3403096

1 INTRODUCTION

Missing data are prevalent, for instance, owing to device issues in sensor readings, transmission problems in networks, or privacy concerns in survey questionnaire [18]. These missing values would obviously encumber subsequent applications. The incomplete data affect not only the induced knowledge in the training phase, but also the application to the test data where missing values may also appear. It is not surprising that more accurate missing data imputation generally leads to better performance in downstream applications, such as classification, clustering or record matching. [1]

	Street	No.	Longitude	Latitude
t_1	Hospital Street	③	9	1
t_2	Hospital St.	④	8	3
t_3	Hospital Street	⑤	6	4
t_4	New York Street	6	0	1
t_5	New York St.	5	2	2
t_6	New York St.	4	3	4
t_7	New York Street	3	4	5
t_8	– (New York Street)	2	5	6
t_9	Hospital Street	⑥	– (5)	– (5)

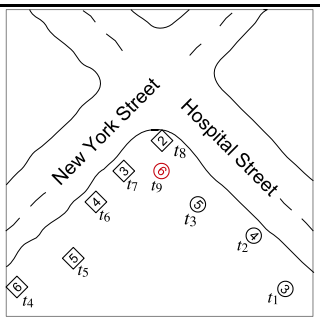

Figure 1: Example POI data with missing values denoted by – and the corresponding truths in (·).

1.1 Challenges

In practice, missing values could appear in various attributes.

(1) The incomplete attribute could either be categorical or numerical, e.g., categorical Street in t_8 or numerical Longitude in t_9 denoted by – in Figure 1. The existing value regression model-based approaches such as [5, 34] impute numerical values but cannot handle the categorical ones.

(2) Missing values could appear in various attributes of tuples, e.g., on Street in t_8 but Longitude and Latitude in t_9 in Figure 1. A regression model or dependency rule [28] may use (Street) No. and Longitude to infer the Latitude value. However, Longitude and

[1] See an empirical study in Section 5.2.

KDD '20, August 23–27, 2020, Virtual Event, CA, USA

ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/3394486.3403096

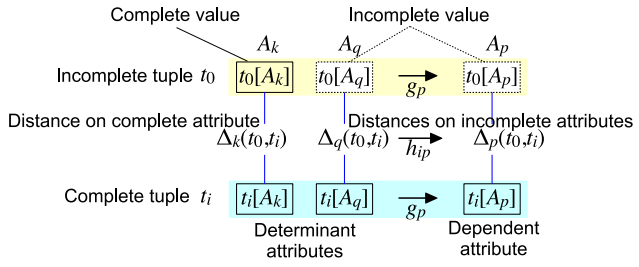

Figure 2: Incomplete values appear in both the determinant attribute A_q and dependent attribute A_p of the value regression model g_p and distance regression model h_{ip}

Latitude values are often missing together in GPS readings. With the missing determinant attribute Longitude, the aforesaid dependency or regression based approaches would not perform. [2]

1.2 Proposal

Rather than directly inferring the missing values by using a dependency rule or regression model (g_p), as illustrated in Figure 2, we propose to predict the distances between missing values and complete ones. The missing values are then imputed according to the inferred distances. Benefits of using distance regression models (h_{ip}) are in two aspects.

(1) Both the categorical and numerical values can be uniformly processed and collaboratively utilized. For instance, both the distances on the categorical attribute Street and the numerical attribute No. are utilized to predict the distance on Latitude. In contrast, a value regression model as aforesaid can infer only among numerical attributes No., Longitude and Latitude, but not the categorical Street.

(2) Both the determinant and dependent attributes could be efficiently imputed, i.e., $t_0[A_q]$ and $t_0[A_p]$ in Figure 2, respectively. Instead of enumerating all the possible $t_0[A_q]$ to determine $t_0[A_p]$, we directly calculate *the most likely distances* of the missing values $t_0[A_q]$ and $t_0[A_p]$ to the complete ones. The imputations are then determined referring to the estimated distances. [3]

Example 1. *Consider a point-of-interest (POI) dataset with four attributes, Street, No., Longitude and Latitude, in Figure 1. Each black circle ◯ with street number indicates the location (longitude and latitude values) of a POI in the Hospital Street. Similarly, the black square □ with street number denotes the location of POI in the New York Street. The POI data are collected by check-in activities, from various service apps, with distinct information formats. Some information may be absent, e.g., the longitude and latitude values of t_9 are missing (we manually label the corresponding truths by red circle ◯).*

To utilize and impute all the attributes, we study the distances on attributes between tuples, e.g., Δ_{Street} or $\Delta_{Latitude}$. Regression models are learned over the distances, such as $h(\Delta_{No.}, \Delta_{Longitude}, \Delta_{Latitude}) \rightarrow \Delta_{Street}$. The distance of t_8[Street] and t_7[Street] can thus be predicted referring to their distances on No., Longitude, Latitude. Based on the predicted distance (e.g., edit distance 1), we determine the most likely imputation t_8[Street] = New York Street.

Moreover, the determinant attributes of the aforesaid distance regression model h could also be missing, such as t_9[Longitude] and t_9[Latitude]. A natural idea is to enumerate the possible GPS locations, and determine the most likely one that can accurately predict the distances between t_9 and other tuples on Street using h. Instead of the costly enumeration, we show in Section 4 how to directly calculate the most likely distances, e.g., $\Delta_{\text{Longitude}}(t_9, t_2) = 2.792$ (without normalization). Similar to the aforesaid imputation of Street, we have t_9[Longitude] = 5 based on the distances.

The example illustrates that with distance models, we can uniformly handle various types of missing values and efficiently impute various attributes of incomplete data.

1.3 Contribution

Our major contributions in this study are as follows.

(1) We formalize the likelihood of a tuple w.r.t. the distance models for predicting distances between tuples in Section 2. An imputation statistically explainable by the likelihood on distances is then derived, which illustrates the rationale of the proposal.

(2) We analyze the hardness of finding the imputation with the maximum likelihood (Theorem 1) in Section 3. The reduction from the 3-SAT problem [15] motivates us to relax the problem by imputing individually the incomplete attributes in a tuple, to eliminate candidate value combinations.

(3) We develop an approximation algorithm by the aforesaid problem relaxation in Section 4. The bound of approximation ratio is studied in general cases (Proposition 4).

(4) We conduct an extensive evaluation, in Section 5, to demonstrate the superiority of our proposal in both imputation accuracy and the improvement of downstream classification, clustering and record matching applications. The experiments run on a number of real datasets with (a) artificial missing values knowing the truth, (b) real missing values having manually labeled truth, and (c) real missing values without labeled truth but having class labels for applications.

2 LIKELIHOOD ON DISTANCES

In this section, we first formalize distance models for predicting distances. Likelihood is then studied w.r.t. the distance predictions.

Consider a relation instance $r = \{t_1, \ldots, t_n\}$ over schema $R = (A_1, \ldots, A_m)$ with complete values. We denote $\text{dom}(A)$ the domain of each attribute $A \in R$. Let t_0' be another tuple over R, e.g., a candidate for imputing the incomplete tuple t_0. [4] We study the likelihood of t_0' by evaluating its distances to tuples in r.

Let Δ_k be a distance metric for each attribute $A_k \in R$, denoted by $\Delta_k(t_0'[A_k], t_i[A_k])$ or simply $\Delta_k(t_0', t_i)$, having $0 \le \Delta_k(t_0', t_i) \le 1$, where t_0', t_i are tuples from R. For instance, it can be a normalization distance [13] for numerical values,

$$\Delta_k(t_0', t_i) = \frac{|t_0'[A_k] - t_i[A_k]|}{\max_{a,b \in \text{dom}(A_k)} |a - b|}, \quad (1)$$

[2] Please refer to Table 2 in Section 5 for a summary of existing representative methods in dealing with various incomplete attributes.
[3] See Section 4 for the approach in detail.
[4] See Section 3.1 for imputation candidates in detail.

or a normalized distance [17] for string values,

$$\Delta_k(t'_0, t_i) = \frac{2 \cdot \text{dist}(t'_0[A_k], t_i[A_k])}{|t'_0[A_k]| + |t_i[A_k]| + \text{dist}(t'_0[A_k], t_i[A_k])}, \quad (2)$$

where $\text{dist}(a, b)$ could be edit distance, cosine similarity, jaccard coefficient or any other string similarity measures [21].

2.1 Distance Models

Instead of the value regression model g that directly predicts the missing value, we study the distance regression model [6, 7] that predicts the distances of t'_0 to tuples $t_i \in r$. Such predictions on distances are then utilized to determine the most likely imputation in Section 3.2.

Note that different tuples may have distinct distance relationships to neighbors. For example, the model for predicting the distances to t_8 at the crossroads in Figure 1 is different from that for t_1. Therefore, for each tuple $t_i \in r, A_p \in R$, we consider a distance regression model h_{ip}. It predicts the distance $\Delta_p(t'_0, t_i)$ on attribute A_p, referring to their distances on the other attributes $R \setminus \{A_p\}$ between t'_0 and t_i,

$$h_{ip}(\{\Delta_k(t'_0, t_i) \mid A_k \in R \setminus \{A_p\}\}) \to \Delta_p(t'_0, t_i).$$

For instance, h_{ip} can be a polynomial regression [2], logistic regression or simply linear regression [20].

We denote $\mathbf{x}_{0i}$ a vector of distances on all the attribute $A_k \in R \setminus \{A_p\}$ between t'_0 and $t_i \in r$, together with a constant term 1,

$$\mathbf{x}_{0i} = (1 \quad x_{0i1} \quad x_{0i2} \quad \dots \quad x_{0ik} \quad \dots \quad x_{0i,n-1})^\top,$$

where $x_{0ik} = \Delta_k(t'_0, t_i)$. Let $y_{0ip} = \Delta_p(t'_0, t_i)$ be the distance to predict between tuples t'_0 and t_i on attribute $A_p \in R$.

The *distance regression model* h_{ip} of the complete tuple $t_i \in r$, for predicting distance $\Delta_p(t'_0, t_i)$ on attribute A_p, is

$$y_{0ip} = h_{ip}(\mathbf{x}_{0i}) + \varepsilon_{ip}, \quad (3)$$

where ε_{ip} is an error term.

Example 2. *Consider the relation $r = \{t_1, \dots, t_7\}$ in Figure 1 with* $\text{dom}(Latitude) = \{0, 1, \dots, 10\}$. *According to Formula 1, the normalized distance between t_2 and t_3 on Latitude is* $\Delta_{Latitude}(t_2, t_3) = \frac{|3-4|}{|10-0|} = 0.1$. *For incomplete attribute $A_4 = Latitude$ of t_9, a linear regression distance model h_{34} of the complete tuple $t_3 \in r$, for predicting distance $\Delta_{Latitude}(t'_9, t_3)$ on attribute Latitude, can be* $y_{934} = 0.022 - 0.302 \cdot \Delta_{Street}(t'_9, t_3) + 0.069 \cdot \Delta_{No.}(t'_9, t_3) + 0.713 \cdot \Delta_{Longitude}(t'_9, t_3)$.

The learning of distance model h_{ip} for each $t_i \in r$ can be performed over a set of the nearest neighbors t_j of t_i. For each t_j, we compute its distances to t_i on all attributes, as the training data.

Unlike the costly discovery of data dependencies by enumerating the combinations of determinant attributes, the learning of distance model naturally utilizes all the other attributes in R excluding A_p. There could be m distance models for each tuple $t_i \in r$.

2.2 Distance Likelihood

We now present the likelihood of t'_0, referring to its distances to various tuples $t_i \in r$ predicted by the corresponding distance regression models h_{ip}.

For the distance regression model h_{ip} of t_i in Formula 3, we consider a normal distribution with zero mean and variance σ_{ip} of the error term [25], i.e., $\varepsilon_{ip} \sim \mathcal{N}(0, \sigma^2_{ip})$. As illustrated in Figure 3, such a normal distribution is widely observed in real datasets.

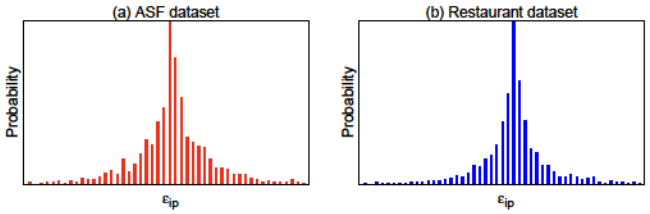

Figure 3: Distributions of error term ε_{ip} in the distance regression models from two different datasets

It follows $y_{0ip} \sim \mathcal{N}(h_{ip}(\mathbf{x}_{0i}), \sigma^2_{ip})$, having

$$f(y_{0ip} \mid \mathbf{x}_{0i}, h_{ip}) = \left(2\pi\sigma^2_{ip}\right)^{-\frac{1}{2}} \exp^{-\frac{(y_{0ip} - h_{ip}(\mathbf{x}_{0i}))^2}{2\sigma^2_{ip}}}.$$

The likelihood of a tuple t'_0 w.r.t. $t_i \in r$ referring to its distance regression model h_{ip} is thus written as log likelihood

$$\mathcal{L}(t'_0 \mid t_i, A_p) = \log f(y_{0ip} \mid \mathbf{x}_{0i}, h_{ip}) \quad (4)$$
$$= -\frac{\log(2\pi\sigma^2_{ip})}{2} - \frac{(y_{0ip} - h_{ip}(\mathbf{x}_{0i}))^2}{2\sigma^2_{ip}}.$$

Considering the m distance models for predicting the distances on all attributes $A_p \in R$.[5] Since the distance models for predicting each attribute $A_p \in R$ are independent, the likelihood of a tuple t'_0 w.r.t. $t_i \in r$ can be written as

$$\mathcal{L}(t'_0 \mid t_i) = \log \prod_{A_p \in R} f(y_{0ip} \mid \mathbf{x}_{0i}, h_{ip}) = \sum_{A_p \in R} \mathcal{L}(t'_0 \mid t_i, A_p) \quad (5)$$
$$= \sum_{A_p \in R} -\frac{\log(2\pi\sigma^2_{ip})}{2} - \frac{(y_{0ip} - h_{ip}(\mathbf{x}_{0i}))^2}{2\sigma^2_{ip}}.$$

In practice, not all the tuples in r may have distance prediction relationships with t'_0. For instance, as aforesaid, the distance to t_8 may be predicted by the distance model of t_7, since both of them are close to the crossroads in Figure 1, but not t_1. Therefore, we consider only the tuples with the largest likelihoods in r for evaluating a tuple. Let $r(t'_0)$ denote the set of κ tuples $t_i \in r$ with the largest likelihoods $\mathcal{L}(t'_0 \mid t_i)$, i.e., top-$\kappa$ likelihoods. Since the tuples are independent, as well as the distance models, we have the *distance likelihood* of a tuple t'_0 w.r.t. r

$$\mathcal{L}(t'_0 \mid r) = \log \prod_{t_i \in r} \prod_{A_p \in R} f(y_{0ip} \mid \mathbf{x}_{0i}, h_{ip}) \quad (6)$$
$$= \sum_{t_i \in r(t'_0)} \mathcal{L}(t'_0 \mid t_i) = \sum_{t_i \in r(t'_0)} \sum_{A_p \in R} \mathcal{L}(t'_0 \mid t_i, A_p)$$
$$= \sum_{t_i \in r(t'_0)} \sum_{A_p \in R} -\frac{\log(2\pi\sigma^2_{ip})}{2} - \frac{(y_{0ip} - h_{ip}(\mathbf{x}_{0i}))^2}{2\sigma^2_{ip}}.$$

[5] Of course, in practice, one may consider only the distance models predicting the distances on a subset of attributes $S \subset R$.

Example 3 (Example 2 continued). *Consider the relation $r = \{t_1, \ldots, t_7\}$ in Figure 1. Let $t'_9 = (Hospital\ Street, 6, 6, 4)$ be an imputation candidate for t_9. Given the distance model h_{34} in Example 2 and the corresponding error term with variance $\sigma^2_{34} = 0.02$, referring to Formula 4, the likelihood of t'_9 w.r.t. h_{34} of t_3 is $\mathcal{L}(t'_9 \mid t_3, Latitude) = -\frac{\log(2\pi*0.002)}{2} - \frac{(0-(0.022-0.302*0+0.069*0.25+0.713*0))^2}{2*0.002} = 2.754$. Considering all the 4 distance models, we have $\mathcal{L}(t'_9 \mid t_3) = 1.195 + 1.152 + 2.712 + 2.754 = 7.813$. Given $\kappa = 5$, it follows $r(t'_9) = \{t_3, t_5, t_7, t_1, t_4\}$. The likelihood is then computed by Formula 6, $\mathcal{L}(t'_9 \mid r) = 7.813 + 4.724 + 4.098 - 0.136 - 9.234 = 7.265$.*

3 IMPUTATION VIA DISTANCE LIKELIHOOD MAXIMIZATION

In this section, we propose to determine the most likely imputation w.r.t. the aforesaid distance models. Hardness of finding the imputation with the maximum likelihood is analyzed in Theorem 1.

3.1 Imputation Candidates

We consider a set of *value candidates*

$$\mathsf{can}(t_0[A_q]) \subseteq \mathsf{dom}(A_q)$$

for imputing $t_0[A_q]$ in a tuple t_0. It can be simply the entire domain of attribute A_q, $\mathsf{dom}(A_q)$, or narrowed down by the existing imputation methods (see Section 6 for a survey), for instance, filtered by distance constraints [28], suggested by k-nearest-neighbors on complete attributes [1], etc. For the complete attributes in t_0, we denote $\mathsf{can}(t_0[A_q]) = \{t_0[A_q]\}$.

By considering all the attributes in R, we define the *tuple candidates* for imputing t_0,

$$\mathsf{can}(t_0) = \prod_{A_q \in R} \mathsf{can}(t_0[A_q]). \tag{7}$$

Example 4 (Example 2 continued). *For incomplete attributes Longitude and Latitude in t_9, with $\mathsf{dom}(Longitude) = \{0, 1, \ldots, 10\}$ and $\mathsf{dom}(Latitude) = \{0, 1, \ldots, 10\}$, we can simply utilize the domain of each incomplete attribute to be imputation candidates. It leads to a number of 121 tuple candidates, i.e., $\mathsf{can}(t_0)$ ={(Hospital Street,6,0,0), (Hospital Street,6,0,1), …, (Hospital Street,6,10,10)}.*

3.2 Problem Statement and Analysis

Among all the possible imputation candidates $t'_0 \in \mathsf{can}(t_0)$, we propose to find the one with the maximum distance likelihood $\mathcal{L}(t'_0 \mid r)$ as defined in Formula 6.

Problem 1. *Given a tuple t_0 with imputation candidates on each incomplete attribute, a set of distance models on complete tuples in r, and a number κ of tuples considered in likelihood evaluation, the optimum imputation problem is to find an imputation t'_0 such that the likelihood $\mathcal{L}(t'_0 \mid r)$ is maximized.*

The corresponding decision problem is thus:

Problem 2. *Given a tuple t_0 with imputation candidates on each incomplete attribute, a set of distance models on complete tuples in r, a number κ of tuples considered in likelihood evaluation, and a constant ℓ, the imputation checking problem is to determine whether exists an imputation t'_0 with likelihood $\mathcal{L}(t'_0 \mid r) \geq \ell$.*

The optimal imputation t'_0 is statistically explainable. Referring to Formula 6, the imputation with the maximum likelihood $\mathcal{L}(t'_0 \mid r)$ is indeed the one with the minimum deviation from the predictions $(y_{0ip} - h_{ip}(\mathbf{x}_{0i}))^2$. In this sense, an imputation with distances to tuples in r most coincide with the underlying distance models.

Example 5 (Example 3 continued). *Consider another imputation candidate $t''_9 = (Hospital\ Street, 6, 5, 5)$. Similar to Example 3, we compute the likelihood $\mathcal{L}(t''_9 \mid r)$ with $r(t''_9) = \{t_2, t_3, t_6, t_5, t_7\}$, having $\mathcal{L}(t''_9 \mid r) = 9.658 + 8.501 + 7.431 + 3.101 + 3.051 = 31.742$. The likelihood is higher than that of t'_9 in Example 3. Indeed, t''_9 is the optimal imputation with the maximum likelihood.*

Hardness Analysis. We show that if the number κ is considered as an input, even for the simple linear regression models h_{ip}, the optimum imputation problem is NP-hard.

Theorem 1. *If the number κ of largest likelihood in $\mathcal{L}(t'_0 \mid r)$ is considered as an input, the imputation checking problem is NP-complete. (Please see Section A.1 for the proof sketch.)*

4 APPROXIMATION WITH INDIVIDUAL ATTRIBUTES

In this section, rather than enumerating all the tuple candidates $t'_0 \in \mathsf{can}(t_0)$ and computing their tuple likelihoods, we relax the setting by computing the individual likelihood of each value candidate $t'_0[A_q] \in \mathsf{can}(t_0[A_q])$. That is, all the incomplete attributes are imputed separately by maximizing the value likelihood, given linear regression distance models h_{ip}. We show that with maximizing the value likelihood over individual attributes, the approximation performance is guaranteed (Proposition 4).

4.1 Likelihood over Individual Attributes

Let us first introduce the individual likelihood of each value candidate $t'_0[A_q] \in \mathsf{can}(t_0[A_q])$ w.r.t. r, denoted by $\mathcal{L}(t'_0[A_q] \mid r)$. The imputation via maximizing the value likelihood is then presented in Algorithm 1 in Section 4.2.

4.1.1 Value Likelihood Definition. For each attribute $A_k \in R$ and tuple $t_i \in r$, we define

$$\delta^{\min}_{0ik} = \min_{t'_0[A_k] \in \mathsf{can}(t_0[A_k])} \Delta_k(t'_0, t_i), \tag{8}$$

$$\delta^{\max}_{0ik} = \max_{t'_0[A_k] \in \mathsf{can}(t_0[A_k])} \Delta_k(t'_0, t_i), \tag{9}$$

which denote the lower and upper bounds of distances on attribute A_k between t_i and the possible imputation t'_0, respectively. For complete attributes A_l and the currently considered value candidate $t'_0[A_q]$, we have

$$\delta^{\min}_{0il} = \delta^{\max}_{0il} = \Delta_l(t_0, t_i) = \Delta_l(t'_0, t_i),$$
$$\delta^{\min}_{0iq} = \delta^{\max}_{0iq} = \Delta_q(t'_0, t_i) = \Delta_q(t'_0[A_q], t_i[A_q]).$$

Any imputation t'_0 with value $t'_0[A_q]$ must have distances $\delta^{\min}_{0ik} \leq \Delta_k(t'_0, t_i) \leq \delta^{\max}_{0ik}$ for all attributes $A_k \in R$.

Consider a linear regression distance model h_{ip} with parameter Φ_{ip}, i.e.,

$$y_{0ip} = h_{ip}(\mathbf{x}_{0i}) + \varepsilon_{ip} = \mathbf{x}^\top_{0i}\Phi_{ip} + \varepsilon_{ip},$$

where

$$\Phi_{ip} = (\phi_{ip0} \quad \phi_{ip1} \quad \dots \quad \phi_{ipk} \quad \dots \quad \phi_{ip,m-1})^\top. \tag{10}$$

Instead of enumerating $t'_0 \in \mathsf{can}(t_0)$ and computing the likelihood $\mathcal{L}(t'_0 \mid t_i, A_p)$ of each tuple candidate in Formula 4, we study directly the most likely distance values in the range of $[\delta^{\min}_{0ik}, \delta^{\max}_{0ik}]$ on all attributes, to estimate the value likelihood of $t'_0[A_q]$ w.r.t. t_i and distance model Φ_{ip}

$$\mathcal{L}(t'_0[A_q] \mid t_i, A_p) = \max_{\substack{\delta^{\min}_{0ip} \le y_{0ip} \le \delta^{\max}_{0ip}, \\ \delta^{\min}_{0ik} \le x_{0ik} \le \delta^{\max}_{0ik}}} \log f(y_{0ip} \mid \mathbf{x}_{0i}, \Phi_{ip}) \tag{11}$$

$$= -\frac{\log(2\pi\sigma^2_{ip})}{2} - \frac{1}{2\sigma^2_{ip}} \min_{\substack{\delta^{\min}_{0ip} \le y_{0ip} \le \delta^{\max}_{0ip}, \\ \delta^{\min}_{0ik} \le x_{0ik} \le \delta^{\max}_{0ik}}} (y_{0ip} - \mathbf{x}_{0i}^\top \Phi_{ip})^2.$$

Similar to tuple likelihood in Formula 6, we define the *value likelihood* of $t'_0[A_q]$ w.r.t. r

$$\mathcal{L}(t'_0[A_q] \mid r) = \sum_{t_i \in r(t'_0[A_q])} \mathcal{L}(t'_0[A_q] \mid t_i) \tag{12}$$

$$= \sum_{t_i \in r(t'_0[A_q])} \sum_{A_p \in R} \mathcal{L}(t'_0[A_q] \mid t_i, A_p),$$

where $\mathcal{L}(t'_0[A_q] \mid t_i)$ is the value likelihood w.r.t. all the distance models of t_i, and $r(t'_0[A_q])$ is the set of κ tuples in r with the largest value likelihoods $\mathcal{L}(t'_0[A_q] \mid t_i)$.

Finally, as illustrated in Algorithm 1 in Section 4.2, rather than considering tuple candidates $t'_0 \in \mathsf{can}(t_0)$, we find the imputation $t'_0[A_q] \in \mathsf{can}(t_0[A_q])$ with the maximum value likelihood $\mathcal{L}(t'_0[A_q] \mid r)$ individually for each incomplete attribute A_q.

Example 6 (Example 2 continued). *Consider the incomplete tuple t_9 in Figure 1 with* $\mathsf{can}(t_9[Longitude]) = \{0, 1, \dots, 10\}$ *and* $\mathsf{can}(t_9[Latitude]) = \{0, 1, \dots, 10\}$. *For complete attributes of t_9, e.g., A_2 = No., we have the bounds of distances between t'_9 and t_3, $\delta^{\min}_{932} = \delta^{\max}_{932} = \Delta_{No.}(t_9, t_3) = 0.25$. Similarly, for the currently considered value candidate, e.g. $t'_9[Longitude] = 5$, we have $\delta^{\min}_{933} = \delta^{\max}_{933} = \Delta_{Longitude}(t'_9, t_3) = 0.1$. For the other incomplete attribute, $t_9[Latitude]$, we have $\delta^{\min}_{934} = 0$ and $\delta^{\max}_{934} = 1$. To compute the value likelihood $\mathcal{L}(t'_9[Longitude] \mid t_3, Longitude)$, the distance of each attribute between t'_9 and t_3 can be any value in the range $[\delta^{\min}_{93k}, \delta^{\max}_{93k}]$, $A_k \in R$, in order to maximize the likelihood. (See Example 7 below for calculation in detail.)*

4.1.2 Value Likelihood Calculation. To calculate the value likelihood $\mathcal{L}(t'_0[A_q] \mid r)$ in Formula 12, it is indeed to solve

$$\min_{\substack{\delta^{\min}_{0ip} \le y_{0ip} \le \delta^{\max}_{0ip}, \\ \delta^{\min}_{0ik} \le x_{0ik} \le \delta^{\max}_{0ik}}} (y_{0ip} - \mathbf{x}_{0i}^\top \Phi_{ip})^2$$

in Formula 11 for $\mathcal{L}(t'_0[A_q] \mid t_i, A_p)$.

To minimize the aforesaid squared value, we consider the maximum and minimum values of error term ε_{ip} in the distance model in Formula 3, i.e.,

$$\epsilon^{\max}_{0ip}(t'_0[A_q]) = \max_{\substack{\delta^{\min}_{0ip} \le y_{0ip} \le \delta^{\max}_{0ip}, \\ \delta^{\min}_{0ik} \le x_{0ik} \le \delta^{\max}_{0ik}}} y_{0ip} - \mathbf{x}_{0i}^\top \Phi_{ip} \tag{13}$$

$$= \delta^{\max}_{0ip} - \sum_{\phi_{ipk} > 0} \delta^{\min}_{0ik} \phi_{ipk} - \sum_{\phi_{ipk} < 0} \delta^{\max}_{0ik} \phi_{ipk},$$

$$\epsilon^{\min}_{0ip}(t'_0[A_q]) = \min_{\substack{\delta^{\min}_{0ip} \le y_{0ip} \le \delta^{\max}_{0ip}, \\ \delta^{\min}_{0ik} \le x_{0ik} \le \delta^{\max}_{0ik}}} y_{0ip} - \mathbf{x}_{0i}^\top \Phi_{ip} \tag{14}$$

$$= \delta^{\min}_{0ip} - \sum_{\phi_{ipk} > 0} \delta^{\max}_{0ik} \phi_{ipk} - \sum_{\phi_{ipk} < 0} \delta^{\min}_{0ik} \phi_{ipk}.$$

The derivations on both formulas are natural, given that $\delta^{\max}_{0ip} \ge \delta^{\min}_{0ip} \ge 0$ and $\delta^{\max}_{0ik} \ge \delta^{\min}_{0ik} \ge 0$ are non-negative distance values.

Lemma 2. *If $\epsilon^{\min}_{0ip}(t'_0[A_q]) \le 0 \le \epsilon^{\max}_{0ip}(t'_0[A_q])$, there must exist an assignment of y_{0ip} and $\mathbf{x}_{0i}$ having $y_{0ip} - \mathbf{x}_{0i}^\top \Phi_{ip} = 0$.*

We consider all three possible cases, (1) both $\epsilon^{\min}_{0ip}(t'_0[A_q])$ and $\epsilon^{\max}_{0ip}(t'_0[A_q])$ are positive, (2) both of them are negative, and (3) otherwise $\epsilon^{\min}_{0ip}(t'_0[A_q]) \le 0 \le \epsilon^{\max}_{0ip}(t'_0[A_q])$. Referring to Lemma 2, the value likelihood in Formula 11 can be directly calculated.

$$\mathcal{L}(t'_0[A_q] \mid t_i, A_p) = \tag{15}$$

$$\begin{cases} -\frac{\log(2\pi\sigma^2_{ip})}{2} - \frac{1}{2\sigma^2_{ip}}(\epsilon^{\min}_{0ip}(t_0[A_q]))^2, & 0 < \epsilon^{\min}_{0ip}(t_0[A_q]) \\ -\frac{\log(2\pi\sigma^2_{ip})}{2} - \frac{1}{2\sigma^2_{ip}}(\epsilon^{\max}_{0ip}(t_0[A_q]))^2, & \epsilon^{\max}_{0ip}(t_0[A_q]) < 0 \\ -\frac{\log(2\pi\sigma^2_{ip})}{2}, & \text{otherwise} \end{cases}$$

The bounds of distances $\delta^{\min}_{0ik}$ and $\delta^{\max}_{0ik}$ can be determined by $\mathsf{can}(t_0[A_k])$ in $O(c)$ time, referring to Formulas 8 and 9. Considering all the m attributes, it costs $O(cm)$ to compute $\delta^{\min}_{0ik}$ and $\delta^{\max}_{0ik}$ for all the attributes $A_k \in R$. Together with the cost $O(m)$ for computing $\epsilon^{\min}_{0ip}(t'_0[A_q])$ and $\epsilon^{\max}_{0ip}(t'_0[A_q])$, the time complexity of computing $\mathcal{L}(t'_0[A_q] \mid t_i, A_p)$ is $O(cm)$.

Example 7 (Example 6 continued). *Given the incomplete tuple t_9 with $\delta^{\min}_{93k}$ and $\delta^{\max}_{93k}$ w.r.t. tuple $t_3 \in r$ for each attribute $A_k \in R$. Consider again the value candidate $t'_9[Longitude] = 5$, we have $\epsilon^{\max}_{933}(t'_9[Longitude]) = 0.1 - (-0.065 + 0 * 0.483 + 0.25 * 0.010 + 0 * 1.241) = 0.162$ and $\epsilon^{\min}_{933}(t'_9[Longitude]) = 0.1 - (-0.065 + 0 * 0.483 + 0.25 * 0.010 + 1 * 1.241) = -1.079$. Since $\epsilon^{\min}_{933}(t'_9[Longitude]) < 0 < \epsilon^{\max}_{933}(t'_9[Longitude])$, according to Formula 15, the value likelihood has $\mathcal{L}(t'_9[Longitude] \mid t_3, Longitude) = -\frac{\log(2\pi * 0.004)}{2} = 2.683$.*

4.2 Approximation Algorithm

Algorithm 1 in Section A presents the imputation via maximizing the value likelihood on each incomplete attribute individually. Instead of enumerating tuple candidates $t'_0 \in \mathsf{can}(t_0)$, DLM investigates each value candidate $t'_0[A_q] \in \mathsf{can}(t_0[A_q])$ separately, in Line 3. With Formulas 12 and 15, the value likelihoods $\mathcal{L}(t'_0[A_q] \mid t_i, A_p)$, $\mathcal{L}(t'_0[A_q] \mid t_i)$ and $\mathcal{L}(t'_0[A_q] \mid r)$ are calculated. Line 10 finds the imputation with the maximum $\mathcal{L}(t'_0[A_q] \mid r)$, for each incomplete attribute $A_q \in U$ in t_0 in Line 2.

As illustrated in Section 4.1.2, Line 6 in Algorithm 1 costs $O(cm)$ to calculate the likelihood $\mathcal{L}(t_0'[A_q] \mid t_i, A_p)$. Similar to the cost of computing the tuple likelihood, a value likelihood $\mathcal{L}(t_0'[A_q] \mid r)$ can thus be calculated in $O(cm^2n + n\log\kappa)$ time. By considering c value candidates for incomplete attribute A_q, the time complexity of Algorithm 1 is $O(cm(cm^2n + n\log\kappa))$, i.e., runs in polynomial time.

Example 8. *Rather than considering and evaluating all the possible 121 tuple candidates in* $\mathsf{can}(t_9)$ *of* t_9 *as in Examples 4, Line 10 in Algorithm 1 finds an imputation with the maximum value likelihood for each incomplete attribute, i.e.,* $\mathcal{L}(t_9'[Longitude] = 5 \mid r) = 52.496$ *and* $\mathcal{L}(t_9'[Latitude] = 5 \mid r) = 50.135$.

4.3 Performance Analysis

Consider the tuple likelihood $\mathcal{L}(t_0' \mid r)$ defined in Formula 6 of the imputation t_0' returned by approximation Algorithm 1, which maximizes the value likelihood $\mathcal{L}(t_0'[A_q] \mid r)$ in Formula 12 instead. Let $\mathcal{L}(t_0^* \mid r)$ be the tuple likelihood of the optimum imputation t_0^*.

Lemma 3. *For any* $A_q \in R$, *the approximate imputation* t_0' *and the optimal solution* t_0^* *have*

$$\mathcal{L}(t_0' \mid r) \le \mathcal{L}(t_0^* \mid r) \le \mathcal{L}(t_0^*[A_q] \mid r) \le \mathcal{L}(t_0'[A_q] \mid r).$$

By Lemma 3, we obtain an upper bound of the maximum tuple likelihood. To capture more precisely the relationships between $\mathcal{L}(t_0' \mid r)$ and $\mathcal{L}(t_0^* \mid r)$, we further investigate the bounds of $\epsilon_{0ip}^{\max}(t_0'[A_q])$ and $\epsilon_{0ip}^{\min}(t_0'[A_q])$ for error terms in Formulas 13 and 14, respectively, which are used to calculate the value likelihood $\mathcal{L}(t_0'[A_q] \mid t_i, A_p)$ in Formula 15. For each Φ_{ip}, the parameter of distance model h_{ip} in Formula 10, let $\varphi_{ip}^{\max} = 1 - \sum_{\phi_{ipk}<0} \phi_{ipk}$ and $\varphi_{ip}^{\min} = -\sum_{\phi_{ipk}>0} \phi_{ipk}$, having $\varphi_{ip}^{\min} \le \epsilon_{0ip}^{\min}(t_0[A_q]), \epsilon_{0ip}^{\max}(t_0[A_q]) \le \varphi_{ip}^{\max}$. We show that the approximation performance is bounded.

Proposition 4. *Approximation Algorithm 1 returns an approximate imputation* t_0', *having*

$$\mathcal{L}(t_0' \mid r) \ge \frac{\alpha}{\beta}\mathcal{L}(t_0^* \mid r),$$

where

$$\alpha = \min_{t_i \in r, A_p \in R}\left(-\frac{\log(2\pi\sigma_{ip}^2)}{2} - \frac{\max((\varphi_{ip}^{\min})^2, (\varphi_{ip}^{\max})^2)}{2\sigma_{ip}^2}\right),$$

$$\beta = \max_{t_i \in r, A_p \in R}\left(-\frac{\log(2\pi\sigma_{ip}^2)}{2}\right),$$

are the minimum and maximum log likelihoods of a tuple referring to the distance regression model in Formula 4.

5 EXPERIMENT

In this section, we compare our proposal DLM with the existing approaches, in terms of both imputation accuracy and improvement of downstream applications. The experiments run on a machine with 3.1GHz CPU and 16GB memory. Table 1 summarizes the datasets used in the experiments. Table 2 lists the major competitors. Please see Section A.5 for detailed experimental settings.

Table 1: Dataset summary

Dataset	$\lvert r\rvert$	$\lvert R\rvert$	Data Type	Missing Value
Restaurant	864	4	categorical	artificial
Solar-Flare	1.4k	10	categorical	artificial
Mushroom	5.6k	22	categorical	artificial
ASF	1.5k	6	numerical	artificial
Letter	20k	16	numerical	artificial
GPS	328	3	numerical	real, labeled truth
MAM	1k	5	numerical	real, no truth
Adult	49k	14	both	real, no truth

Table 2: Representative imputation methods on various incomplete attributes with various value types

Method	Category	Data Type	Incomplete Type
kNNE [8]	neighbor	both	various
MIBOS [31]	neighbor	categorical	various
GMM [33]	cluster	numerical	various
CMI [35]	cluster	categorical	various
LOESS [5]	regression	numerical	fixed
IIM [34]	regression	numerical	fixed
ERACER [19]	statistical	both	fixed
MC [4]	statistical	numerical	various
ER [12]	constraint	categorical	fixed
HoloClean [24]	constraint	categorical	fixed
DD [28]	constraint	both	fixed

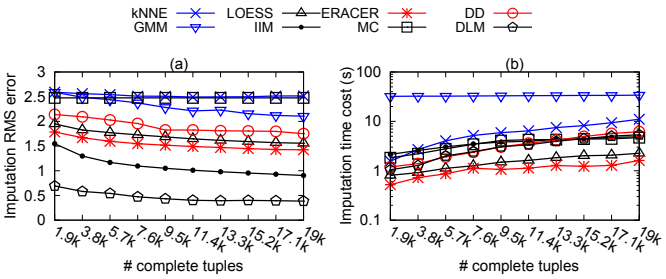

Figure 4: Varying the number of complete tuples in r, over Letter with 1k incomplete tuples, 1 incomplete attribute and $\kappa = 10$

5.1 Comparison with Existing Imputation

Tables 3 and 4 report the performance of imputing numerical and categorical values, respectively, over the datasets with known truths as listed in Table 1. For numerical values, we report the RMS error between the imputation and the corresponding truth, while categorical values use the accuracy measure as introduced in Section A.5.4. Approaches supporting the corresponding data types are considered in Table 2. Please see Section 6 for explanations on categorizing these 11 existing imputation approaches, and Section A.5.5 for the corresponding implementation details.

Table 3: Imputation RMS error of DLM compared to the existing approaches summarized in Table 2, over various numerical datasets with known truth as listed in Table 1

Dataset	kNNE	GMM	LOESS	IIM	ERACER	MC	DD	DLM
GPS	$1.37*10^{-5}$	$6.05*10^{-5}$	$1.91*10^{-5}$	$1.74*10^{-5}$	$3.62*10^{-5}$	$2.57*10^{-4}$	$1.15*10^{-5}$	$\mathbf{6.42}*10^{-6}$
Letter	2.515	2.104	1.556	0.903	1.424	2.475	1.750	**0.385**
ASF	19.804	39.336	9.102	7.731	36.835	101.587	10.057	**1.054**

Table 4: Imputation Accuracy of DLM compared to the existing approaches summarized in Table 2, over various categorical datasets with known truth as listed in Table 1

Dataset	kNNE	MIBOS	CMI	ERACER	ER	HoloClean	DD	DLM
Solar-Flare	0.539	0.396	0.380	0.601	0.330	0.287	0.428	**0.780**
Mushroom	0.785	0.477	0.614	0.701	0.364	0.228	0.486	**0.910**
Restaurant	0.388	0.212	0.124	0.368	0.076	0.016	0.408	**0.504**

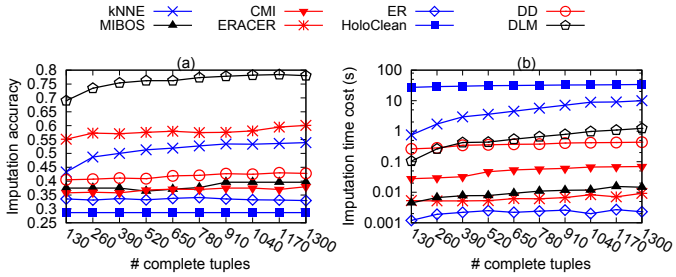

Figure 5: Varying the number of complete tuples in r, over Solar-Flare with 100 incomplete tuples, 1 incomplete attribute and $\kappa = 10$

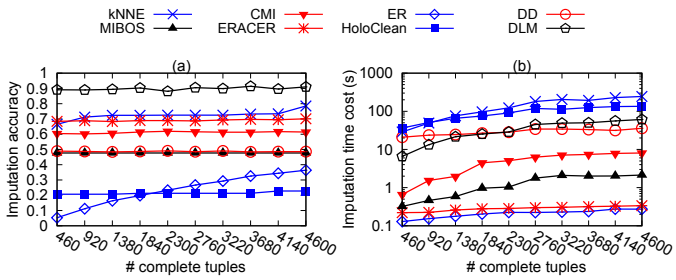

Figure 6: Varying the number of complete tuples in r, over Mushroom with 1k incomplete tuples, 1 incomplete attribute and $\kappa = 5$

Figures 4, 5 and 6 present the results over the relatively larger data sizes. With the increase of complete tuples available in r, the imputation performance of all approaches improves. The result is not surprising, since the imputation techniques more or less rely on the complete tuples, either on their values or their models. When the size of complete tuples is larger, e.g., Letter in Figure 4, the result becomes stable. The corresponding time costs of imputation increase.

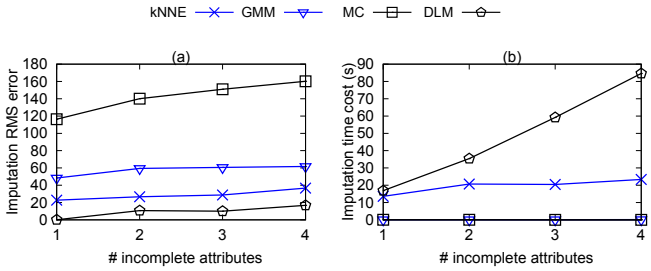

Figure 7: Varying the number of incomplete attributes $|U|$, over ASF with 100 incomplete tuples, 1.4k complete tuples in r and $\kappa = 5$

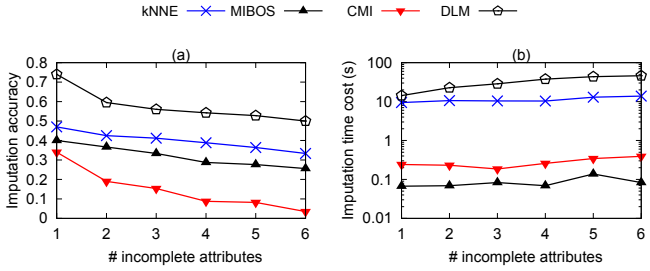

Figure 8: Varying the number of incomplete attributes $|U|$, over Solar-Flare with 100 incomplete tuples, 1.3k complete tuples in r and $\kappa = 10$

Figures 7 and 8 report the results on various numbers of incomplete attributes. Some approaches are omitted, since they can only be applied to impute a fixed set of incomplete attributes, as summarized in Table 2. It is not surprising that the imputation performance drops when more attributes are incomplete.

Our proposed DLM algorithm consistently shows the best imputation accuracy over various numerical and categorical datasets, as well as various sets of incomplete attributes. Similar results are observed over the GPS data with real missing values in Table 3.

Table 5: Application accuracy without/with imputation over various numerical datasets

Application	Dataset	Missing	kNNE	GMM	LOESS	IIM	ERACER	MC	DD	DLM
Classification	MAM	0.818	0.824	0.827	0.827	0.828	0.828	0.821	0.822	**0.838**
Classification	Letter	0.736	0.804	0.805	0.817	0.850	0.819	0.801	0.808	**0.871**
Clustering	ASF	0.880	0.958	0.933	0.971	0.977	0.935	0.892	0.970	**0.995**

Table 6: Application accuracy without/with imputation over various categorical datasets

Application	Dataset	Missing	kNNE	MIBOS	CMI	ERACER	ER	HoloClean	DD	DLM
Classification	Adult	0.791	0.799	0.799	0.803	0.805	0.798	0.796	0.798	**0.813**
Classification	Mushroom	0.647	0.729	0.692	0.710	0.717	0.687	0.673	0.695	**0.752**
Matching	Restaurant	0.765	0.811	0.809	0.793	0.809	0.776	0.769	0.814	**0.824**

5.2 Application Case Study

To validate the effectiveness of applying imputation in real applications, we consider three case studies on clustering, classification and record matching.

5.2.1 Clustering with/without Imputation. The clustering experiment is performed over the ASF data. We use the k-means clustering implementation [30]. Since the ASF dataset is originally complete, we consider the clustering results over the original data as the truth of clusters. A total number of 100 clusters are obtained, which could be determined according to [23]. Artificial missing values are then introduced as mentioned in Section A.5.1, and imputed by using various approaches. The clustering algorithm is conducted again over the data with missing values and the imputed dataset. The purity [13] measure is employed, which counts for each cluster the number of data points from the most common class (truth cluster). The higher the purity is, the better the imputation improves clustering. The accuracies of clustering results without/with imputation are reported in Table 5.

It is not surprising that approaches with better imputation performance in Table 3 generally have higher clustering accuracy as well in Table 5. Our proposal DLM again shows the best clustering performance.

5.2.2 Classification with/without Imputation. The classification application is first performed over the MAM and Adult datasets with real-world missing values. Rather than labeling the truth of missing values, the datasets provide the class labels of tuples instead. Thereby, we use f1-score of classification to evaluate the imputation performance. Again, the classification is performed over the data without/with various imputations. We employ the kNN classifier implementation [30], and conduct 5-fold cross validation. Moreover, the Letter and Mushroom datasets also include class labels. Tables 5 and 6 report the classification accuracy over Letter and Mushroom with 1k incomplete tuples (artificially introduced as mentioned in Section A.5.1).

The proposed DLM approach indeed leads to the best classification accuracy. The results demonstrate again the superiority of our proposal in imputing real missing values, as well as in improving the downstream applications.

5.2.3 Record Matching with/without Imputation. The record matching [9] is performed over the Restaurant data. Missing values are artificially introduced as mentioned in Section A.5.1. We use the existing rule-based implementation [11] and perform over the data without/with imputation.

As shown in Table 6, the record matching accuracy (f-measure) is generally related to the imputation accuracy in Table 4. With imputation, some missing data could be repaired and the matching accuracy is improved compared to Missing without imputation. Our proposed DLM, having the most accurate imputation in Table 4, leads to the highest matching f-measure as well.

6 RELATED WORK

Table 2 summarizes the typical data imputation methods, together with well supported data types (categorical/numerical). Some approaches can impute the missing values on various attributes, while the others support only a fixed set of incomplete attributes.

Nearest Neighbor-based. The nearest neighbor-based imputation (kNN) [1] finds a set of neighbors t_i for the tuple t_0 with missing values $t_0[U]$, referring to its complete attributes $t_0[R \setminus U]$. The values on attribute $A \in U$ of the neighbors are then aggregated as the imputation of the missing value $t_0[A]$. MIBOS [31] computes a tuple-similarity which is indeed defined on value equality, i.e., by counting the number of attributes with equal values. The kNN Ensemble [8] explores more neighbors on various subsets of the complete attributes $R \setminus U$. The major problem of the kNN-based imputation is that the candidates suggested by neighbors may distant with each other, and thus the aggregated value is not accurate.

Clustering-based. Instead of finding nearest neighbors on various subsets of the complete attributes $R \setminus U$, clustering methods are also employed to explore different groups of neighbors, e.g., by kernel function imputation strategy [35], fuzzy k-means [16] and its iterative manner [22], or the advanced Gaussian mixture model (GMM) [33]. Again, values of neighbors from various clusters are aggregated. It still suffers from the aforesaid problem of distant values from various neighbors.

Regression-based. Rather than simply aggregating the values suggested by neighbors, LOESS [5] learns a regression model from

nearest neighbors, for predicting the missing value $t_0[A]$ referring to the complete attributes $t_0[R \setminus U]$. Unfortunately, as aforesaid in the kNN imputation [1], the neighbors found by the complete attributes $R \setminus U$ may be distant with each other, and thus do not share the same/similar value regression model. Thereby, IIM [34] learns an individual regression model for each complete tuple, referring to its nearest neighbors. It is notable that the regression-based method cannot handle categorical data.

Statistics-based. The methods based on statistical models [19, 32] capture the probabilistic correlations between reliable attributes with correct values and flexible attributes with dirty values. The imputation is thus to find the values that can maximize the likelihood w.r.t. the probabilistic correlations. A complex relational dependency network is considered in [19] to model the probabilistic relationships among attributes. As reported in [32], ERACER [19] shows better performance than [32]. MC [4] formulates the imputation problem as a task of filling the missing entries of a partially observed matrix. Given an incomplete matrix with missing values, the matrix completion problem is to find the corresponding lowest rank matrix. Again, it imputes only numerical values.

Constraint-based. To determine certain fixes, editing rules (ER) and certain regions [12] are considered, based on the equality relationships between the incomplete tuples and reference data. Fixing rules [29] and Sherlock rules [14] can also determine certain imputations for missing values, relying on experts to specify evidence and negative patterns or seed rules. Unfortunately, the heterogeneous values with various information formats often make the null cells barely imputed, owing to the strict value equality relationships considered in editing/fixing rules. For the same reason, the imputation is not accurate by HoloClean [24] in Section 5.1. To deal with the similarity relationships between heterogeneous values, differential dependencies (DD) are employed for imputation [28]. However, for those values satisfying the constraints, the DD rules treat them equally and cannot further distinguish which one should be the true imputation.

7 CONCLUSIONS

In this paper, we study the imputation of incomplete data in (1) various types of values and (2) various attributes of tuples. To address both variety issues, the likelihood on distances is investigated. The distance likelihood can be uniformly defined over various types of values, and efficiently calculated without enumerating the combinations of imputation candidates on various incomplete attributes. To find the optimum imputation with the maximum distance likelihood, we (1) analyze NP-hardness of the problem in Theorem 1; (2) propose an approximation algorithm with performance guarantee in Proposition 4. Extensive experiments on datasets (with real missing values) show that our proposal DLM improves the imputation accuracy compared to 11 existing approaches in 5 categories, and improves the downstream applications such as classification, clustering and record matching.

Acknowledgement. This work is supported in part by the National Key Research and Development Plan (2019YFB1705301) and the National Natural Science Foundation of China (61572272, 71690231).

REFERENCES

[1] N. S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.
[2] C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006.
[3] P. Bohannon, W. Fan, M. Flaster, and R. Rastogi. A cost-based model and effective heuristic for repairing constraints by value modification. In *SIGMOD*, pages 143–154. ACM, 2005.
[4] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.
[5] W. S. Cleveland and C. Loader. *Smoothing by Local Regression: Principles and Methods*, pages 10–49. Physica-Verlag HD, Heidelberg, 1996.
[6] C. Cuadras and C. Arenas. A distance based regression model for prediction with mixed data. *Communications in Statistics-Theory and Methods*, 19(6):2261–2279, 1990.
[7] A. H. de Souza Júnior, F. Corona, G. D. A. Barreto, Y. Miché, and A. Lendasse. Minimal learning machine: A novel supervised distance-based approach for regression and classification. *Neurocomputing*, 164:34–44, 2015.
[8] C. Domeniconi and B. Yan. Nearest neighbor ensemble. In *ICPR*, pages 228–231, 2004.
[9] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate record detection: A survey. *TKDE*, 19(1):1–16, 2007.
[10] W. Fan, F. Geerts, L. V. S. Lakshmanan, and M. Xiong. Discovering conditional functional dependencies. In *ICDE*, pages 1231–1234, 2009.
[11] W. Fan, X. Jia, J. Li, and S. Ma. Reasoning about record matching rules. *PVLDB*, 2(1):407–418, 2009.
[12] W. Fan, J. Li, S. Ma, N. Tang, and W. Yu. Towards certain fixes with editing rules and master data. *PVLDB*, 3(1):173–184, 2010.
[13] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques, 3rd edition*. Morgan Kaufmann, 2011.
[14] M. Interlandi and N. Tang. Proof positive and negative in data cleaning. In *ICDE*, pages 18–29, 2015.
[15] R. M. Karp. Reducibility among combinatorial problems. In *Proceedings of a symposium on the Complexity of Computer Computations*, pages 85–103, 1972.
[16] D. Li, J. Deogun, W. Spaulding, and B. Shuart. Towards missing data imputation: a study of fuzzy k-means clustering method. In *Rough sets and current trends in computing*, volume 3066, pages 573–579. Springer, 2004.
[17] Y. Li and B. Liu. A normalized levenshtein distance metric. *TPAMI*, 29(6):1091–1095, 2007.
[18] R. J. Little and D. B. Rubin. *Statistical analysis with missing data*. John Wiley & Sons, 2014.
[19] C. Mayfield, J. Neville, and S. Prabhakar. ERACER: a database approach for statistical inference and data cleaning. In *SIGMOD*, pages 75–86, 2010.
[20] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. MIT press, 2018.
[21] G. Navarro. A guided tour to approximate string matching. *ACM Comput. Surv.*, 33(1):31–88, 2001.
[22] S. Nikfalazar, C. Yeh, S. E. Bedingfield, and H. A. Khorshidi. A new iterative fuzzy clustering algorithm for multiple imputation of missing data. In *FUZZ-IEEE*, pages 1–6, 2017.
[23] C. Patil and I. Baidari. Estimating the optimal number of clusters k in a dataset using data depth. *Data Science and Engineering*, 4(2):132–140, 2019.
[24] T. Rekatsinas, X. Chu, I. F. Ilyas, and C. Ré. Holoclean: Holistic data repairs with probabilistic inference. *PVLDB*, 10(11):1190–1201, 2017.
[25] J. T. Rgd Steel. Principales and pricedures of statistics. 1960.
[26] W. Rudin et al. *Principles of mathematical analysis*, volume 3. McGraw-hill New York, 1964.
[27] S. Song and L. Chen. Differential dependencies: Reasoning and discovery. *ACM Trans. Database Syst.*, 36(3):16:1–16:41, 2011.
[28] S. Song, A. Zhang, L. Chen, and J. Wang. Enriching data imputation with extensive similarity neighbors. *PVLDB*, 8(11):1286–1297, 2015.
[29] J. Wang and N. Tang. Towards dependable data repairing with fixing rules. In *ICDE*, pages 457–468, 2014.
[30] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.
[31] S. Wu, X. Feng, Y. Han, and Q. Wang. Missing categorical data imputation approach based on similarity. In *SMC*, pages 2827–2832, 2012.
[32] M. Yakout, L. Berti-Équille, and A. K. Elmagarmid. Don't be scared: use scalable automatic repairing with maximal likelihood and bounded changes. In *SIGMOD*, pages 553–564, 2013.
[33] X. Yan, W. Xiong, L. Hu, F. Wang, and K. Zhao. Missing value imputation based on gaussian mixture model for the internet of things. *Mathematical Problems in Engineering*, 2015, 2015.
[34] A. Zhang, S. Song, Y. Sun, and J. Wang. Learning individual models for imputation. In *ICDE*, pages 160–171, 2019.
[35] S. Zhang, J. Zhang, X. Zhu, Y. Qin, and C. Zhang. Missing value imputation based on data clustering. *Trans. Computational Science*, 1:128–138, 2008.

A REPRODUCIBILITY

The code implementation of our method is publicly available.[6]

A.1 Proof Sketch of Theorem 1

The problem is clearly in NP. Given an imputation t'_0, it can be verified in polynomial time whether its likelihood has $\mathcal{L}(t'_0 \mid r) \geq \ell$.

To prove the NP-hardness, we show a reduction from the 3-SAT problem, which is one of Karp's 21 NP-complete problems [15].

Let $C = C_1 \wedge C_2 \wedge \cdots \wedge C_v$ be a Boolean formula in 3-CNF with v clauses, where $C_i = l_1^i \vee l_2^i \vee l_3^i$ for each clause $C_i, i = 1, \ldots, v$. Each literal $l_j^i, j = 1, 2, 3$, is either z or $\neg z$ for some variable z. We create a relation schema $R = (A, Z_1, \ldots, Z_w, B_1, \ldots, B_v)$, where each attribute Z_k corresponds to variable $z_k, k = 1, \ldots, w$. For each clause C_i, we place three complete tuples t_{i1}, t_{i2} and t_{i3} in r ($|r| = 3v$). For the literal l_j^i in the form of z_k in C_i, we assign $t_{ij}[Z_k] = 1$, while the literal in the form of $\neg z_k$ in C_i leads to $t_{ij}[Z_k] = -1$. For the other z_k that does not appear in C_i, it has $t_{ij}[Z_k] = 0$. We show that C has a satisfying assignment if and only if t_0 has an imputation t'_0 with likelihood $\mathcal{L}(t'_0 \mid r) \geq \ell$.

A.2 Proof of Lemma 2

Continuous variables y_{0ip} and x_{0ik} denote the distance values in the ranges of $[\delta_{0ip}^{\min}, \delta_{0ip}^{\max}]$ and $[\delta_{0ik}^{\min}, \delta_{0ik}^{\max}]$, respectively. Note that $y_{0ip} - \mathbf{x}_{0i}^\top \Phi_{ip}$ is a linear function. Given the maximum and minimum values of $y_{0ip} - \mathbf{x}_{0i}^\top \Phi_{ip}$ having $\epsilon_{0ip}^{\min}(t'_0[A_q]) \leq 0 \leq \epsilon_{0ip}^{\max}(t'_0[A_q])$, according to the intermediate value theorem [26], there must exist a solution of variables y_{0ip} and x_{0ik} such that $y_{0ip} - \mathbf{x}_{0i}^\top \Phi_{ip} = 0$.

A.3 Proof of Lemma 3

According to Formula 6, the optimal imputation $t_0^* \in \mathsf{can}(t_0)$ has

$$\mathcal{L}(t_0^* \mid r) = \sum_{t_i \in r(t_0^*)} \sum_{A_p \in R} -\frac{\log(2\pi\sigma_{ip}^2)}{2} - \frac{(y_{0ip}^* - (\mathbf{x}_{0i}^*)^\top \Phi_{ip}))^2}{2\sigma_{ip}^2}.$$

Combining Formulas 11 and 12, we obtain

$$\mathcal{L}(t_0^*[A_q] \mid r) = \sum_{t_i \in r(t_0^*[A_q])} \sum_{A_p \in R} \mathcal{L}(t_0^*[A_q] \mid t_i, A_p)$$

$$= \sum_{t_i \in r(t_0^*[A_q])} \sum_{A_p \in R} -\frac{\log(2\pi\sigma_{ip}^2)}{2} - \frac{1}{2\sigma_{ip}^2} \min_{\substack{\delta_{0ip}^{\min} \leq y_{0ip} \leq \delta_{0ip}^{\max}, \\ \delta_{0ik}^{\min} \leq x_{0ik} \leq \delta_{0ik}^{\max}}} (y_{0ip} - \mathbf{x}_{0i}^\top \Phi_{ip})^2.$$

Moreover, for any tuple $t_i \in r$, it always has

$$(y_{0ip}^* - (\mathbf{x}_{0i}^*)^\top \Phi_{ip}))^2 \geq \min_{\substack{\delta_{0ip}^{\min} \leq y_{0ip} \leq \delta_{0ip}^{\max}, \\ \delta_{0ik}^{\min} \leq x_{0ik} \leq \delta_{0ik}^{\max}}} (y_{0ip} - \mathbf{x}_{0i}^\top \Phi_{ip})^2.$$

It follows

$$\mathcal{L}(t_0^* \mid t_i) \leq \mathcal{L}(t_0^*[A_q] \mid t_i).$$

[6] https://github.com/DLMImputation/DLM

Referring to the aforesaid definitions, we have

$$\mathcal{L}(t_0^* \mid r) = \sum_{t_i \in r(t_0^*)} \mathcal{L}(t_0^* \mid t_i) \leq \sum_{t_i \in r(t_0^*)} \mathcal{L}(t_0^*[A_q] \mid t_i)$$
$$\leq \sum_{t_i \in r(t_0^*[A_q])} \mathcal{L}(t_0^*[A_q] \mid t_i) = \mathcal{L}(t_0^*[A_q] \mid r).$$

Since t_0^* is the optimum imputation with the maximum tuple likelihood $\mathcal{L}(t_0^* \mid r)$, for any approximate imputation $t'_0 \in \mathsf{can}(t_0)$, it always has

$$\mathcal{L}(t'_0 \mid r) \leq \mathcal{L}(t_0^* \mid r).$$

Finally, Algorithm 1 finds the imputation $t'_0[A_q]$ with the maximum value likelihood $\mathcal{L}(t'_0[A_q] \mid r)$ for each incomplete attribute A_q of t_0, i.e.,

$$\mathcal{L}(t_0^*[A_q] \mid r) \leq \mathcal{L}(t'_0[A_q] \mid r).$$

The conclusion is proved.

A.4 Proof of Proposition 4

According to the definition of $\mathcal{L}(t'_0 \mid r)$ in Formula 6 and the definitions of $\delta_{0ik}^{\min}$ and $\delta_{0ik}^{\max}$ in Formulas 8-9, we have

$$\mathcal{L}(t'_0 \mid r) = \sum_{t_i \in r(t'_0)} \sum_{A_p \in R} -\frac{\log(2\pi\sigma_{ip}^2)}{2} - \frac{(y_{0ip} - h_{ip}(\mathbf{x}_{0i}))^2}{2\sigma_{ip}^2}$$

$$\geq \sum_{t_i \in r(t'_0)} \sum_{A_p \in R} -\frac{\log(2\pi\sigma_{ip}^2)}{2} - \frac{1}{2\sigma_{ip}^2} \max_{\substack{\delta_{0ip}^{\min} \leq y_{0ip} \leq \delta_{0ip}^{\max}, \\ \delta_{0ik}^{\min} \leq x_{0ik} \leq \delta_{0ik}^{\max}}} (y_{0ip} - \mathbf{x}_{0i}^\top \Phi_{ip})^2$$

$$\geq \sum_{t_i \in r(t'_0)} \sum_{A_p \in R} \min_{t_i \in r, A_p \in R} \left(-\frac{\log(2\pi\sigma_{ip}^2)}{2} - \frac{\max((\varphi_{ip}^{\min})^2, (\varphi_{ip}^{\max})^2)}{2\sigma_{ip}^2} \right)$$

$$= \kappa\alpha m.$$

According to Lemma 3, we have

$$\mathcal{L}(t_0^* \mid r) \leq \mathcal{L}(t'_0[A_q] \mid r) = \sum_{t_i \in r(t'_0[A_q])} \sum_{A_p \in R} -\frac{\log(2\pi\sigma_{ip}^2)}{2} -$$
$$\frac{1}{2\sigma_{ip}^2} \min_{\substack{\delta_{0ip}^{\min} \leq y_{0ip} \leq \delta_{0ip}^{\max}, \\ \delta_{0ik}^{\min} \leq x_{0ik} \leq \delta_{0ik}^{\max}}} (y_{0ip} - \mathbf{x}_{0i}^\top \Phi_{ip})^2$$

$$\leq \sum_{t_i \in r(t'_0[A_q])} \sum_{A_p \in R} \max_{t_i \in r, A_p \in R} \left(-\frac{\log(2\pi\sigma_{ip}^2)}{2} \right) = \kappa\beta m.$$

Combining the aforesaid derivations, it concludes

$$\mathcal{L}(t'_0 \mid r) \geq \frac{\kappa\alpha m}{\kappa\beta m} \mathcal{L}(t_0^* \mid r) = \frac{\alpha}{\beta} \mathcal{L}(t_0^* \mid r).$$

A.5 Experimental Settings

A.5.1 Artificial Missing Values. Datasets Restaurant[7], Mushroom[8], Solar-Flare[9], ASF[10] and Letter[11] are datasets originally complete. Following the same line of evaluating data repairing techniques by

[7] http://www.cs.utexas.edu/users/ml/riddle/data.html
[8] https://sci2s.ugr.es/keel/dataset.php?cod=178
[9] http://archive.ics.uci.edu/ml/datasets/solar+flare
[10] http://archive.ics.uci.edu/ml/datasets/Airfoil+Self-Noise
[11] https://archive.ics.uci.edu/ml/datasets/letter+recognition

Algorithm 1: DLM(r, t_0, κ)

Input: r a set of complete tuples,
t_0 a tuple with incomplete attributes U,
κ the number of tuples with the largest likelihood considered for evaluating a candidate
Output: an imputation t_0' of t_0 with the maximum value likelihood

$t_0' \leftarrow t_0$;
for *each incomplete attribute* $A_q \in U$ *in* t_0 **do**
  for *each* $t_0'[A_q] \in \text{can}(t_0[A_q])$ **do**
    for *each* $t_i \in r$ **do**
      for *each* $A_p \in R$ **do**
        $\mathcal{L}(t_0'[A_q] \mid t_i, A_p) \leftarrow$ likelihood computed by Formula 15;
      $\mathcal{L}(t_0'[A_q] \mid t_i) \leftarrow$ likelihood computed by aggregating $\mathcal{L}(t_0'[A_q] \mid t_i, A_p)$;
    $r(t_0'[A_q]) \leftarrow$ set of tuples $t_i \in r$ with κ-largest likelihoods $\mathcal{L}(t_0'[A_q] \mid t_i)$;
    $\mathcal{L}(t_0'[A_q] \mid r) \leftarrow$ likelihood computed by aggregating $\mathcal{L}(t_0'[A_q] \mid t_i)$ in Formula 12;
  $t_0'[A_q] = \arg\max_{t_0'[A_q] \in \text{can}(t_0[A_q])} \mathcal{L}(t_0'[A_q] \mid r)$;
return t_0'

artificially injecting errors [3], we randomly remove values from various attributes in tuples as incomplete data.

A.5.2 Real Missing Values and Labeled Truth. GPS is a dataset manually collected by carrying a GPS device and walking around the campus. The GPS readings are often temporally unavailable, owing to various reasons such as low battery, energy saving or weak signals. There are 54 incomplete tuples with real missing values on Longitude and Latitude naturally embedded in the dataset. The corresponding timestamp is available as the complete attribute, since the collecting program intends to record the GPS readings in every second. Moreover, as we know exactly the trajectory, truths of the missing values are manually labeled.

A.5.3 Real Missing Values without Ground Truth. Datasets MAM[12] and Adult[13] are two datasets with 13.63% and 7.41% real-world missing values. MAM contains a BI-RADS assessment, which can be used to predict the severity (benign or malignant) of a mammographic mass lesion from BI-RADS attributes and the patient's age. Adult is extracted from census data of the United States in 1994, describing some social information about the citizens. Rather than comparing the imputation accuracy, we study the classification over the datasets without/with various imputation methods. A higher classification accuracy [13] generally indicates the better effectiveness of imputation.

A.5.4 Criteria. For numerical values, to evaluate the imputation accuracy, we compare the imputation t_0' to the corresponding truth $t_0^{\#}$, using the root-mean-square error, $\text{RMS} = \sqrt{\sum_{A \in U} \frac{(t_0'[A] - t_0^{\#}[A])^2}{|U|}}$.

[12] https://sci2s.ugr.es/keel/dataset.php?cod=86
[13] https://sci2s.ugr.es/keel/dataset.php?cod=192

For categorical datasets, e.g., Restaurant, we measure the accuracy on whether an imputation equals exactly the ground truth. Let truth be the set of truth values for incomplete data and found be the set of results returned by imputation algorithms. The accuracy is given by $\text{accuracy} = \frac{|\text{truth} \cap \text{found}|}{|\text{truth}|}$, i.e., the proportion of incomplete values accurately imputed.

A.5.5 Implementation Details. As listed in Table 2, the major competitors of our proposal are as follows.

(1) Neighbor-based kNN Ensemble (kNNE) [8] employs different subsets of complete attributes in the incomplete tuple, to generate a diverse set of NN classifiers for ensemble learning.

(2) Tuple similarity-based MIBOS [31] finds the complete tuples having the maximum number of same values on complete attributes with the incomplete tuple, to impute the missing values.

(3) Gaussian mixture model-based GMM [33] clusters the complete data based on EM algorithm. The incomplete data are classified according to the result of clustering. The complete tuples closest to the incomplete tuple in the same cluster are utilized to impute the missing values.

(4) Clustering-based CMI [35] employs k-Means clustering to divide the dataset (including the instances with missing values) into clusters. The complete tuples in the same cluster are used to impute the incomplete tuples.

(5) Value regression model-based LOESS [5] learns regression models among nearest neighbors to impute missing values, where the incomplete attributes are used as the dependent attributes and all the other complete attributes are regarded as the determinant attributes of the regression models.

(6) Individual regression-based IIM [34] learns individual regression models for each complete tuple, whose determinant and dependent attributes are the complete and incomplete attributes of incomplete tuples, respectively.

(7) Statistics-based ERACER [19] iteratively learns a global relational dependency model, i.e., linear regression for numerical data and relational dependency network for categorical data in our experiments, to infer the probabilistic relationships among attributes.

(8) Matrix completion via convex optimization MC [4] forms a convex relaxation of the matrix completion problem and minimizes the nuclear norm. The convex relaxation is solved using semidefinite programming.

(9) ER [12] finds certain fixes with editing rules defined by value equality. We consider all the complete tuples as master data and discover CFDs from r (by [10]) as editing rules.

(10) HoloClean [24] employs not only the aforesaid discovered CFDs as constraints but also the statistical learning and probabilistic inference to impute missing values.

(11) Similarity constraint-based DD [28] imputes missing values with differential dependencies, discovered by [27].

The imputation candidates of DLM are given by the kNN neighbors as introduced in Section 3.1. Similar to distance models in our DLM, the models in the GMM approach could be offline learned. In order to mitigate the risk of a sub-optimal configuration and insufficient rules in the case of rule-based tools, each approach in comparison has been configured in a best-effort fashion (e.g., by iteratively choosing good parameters or by defining a reasonable set of quality rules).