10-1-2021

Data Imputation Using Differential Dependency and Fuzzy Multi-Objective Linear Programming

Mohammadreza Safi
University of Windsor

Data Imputation Using Differential Dependency and Fuzzy Multi-Objective Linear Programming

By

Mohammadreza Safi

A Thesis
Submitted to the Faculty of Graduate Studies through
the Department of Electrical and Computer Engineering
in Partial Fulfillment of the Requirements for
the Degree of Master of Applied Science
at the University of Windsor

Windsor, Ontario, Canada

2021

Data Imputation Using Differential Dependency and Fuzzy Multi-Objective Linear Programming

by

Mohammadreza Safi

APPROVED BY:

_____

J. Ahamed

Department of Mechanical, Automotive & Materials Engineering

_____

M. Hassanzadeh

Department of Electrical and Computer Engineering

_____

S. Alirezaee, Co-Advisor

Department of Electrical and Computer Engineering

_____

M. Ahmadi, Co-Advisor

Department of Electrical and Computer Engineering

September 20, 2021

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

Missing or incomplete data is a serious problem when it comes to collecting and analyzing data for forecasting, estimating, and decision making. Since data quality is so important in machine learning and its results, in most cases data imputation is much more appropriate than ignoring them. Missing data imputation is often based on considering equality, similarity, or distance of neighbors. Researchers use different approaches for neighbors' equalities or similarities. Every approach has its advantages and limitations. Instead of equality, some researchers use inequalities together with a few relationships or similarity rules. In this thesis, after recalling some basic imputation methods, we discus about data imputation based on differential dependencies (DDs). DDs are conditional rules in which the closeness of the values of each pair of tuples in some attribute indicates the closeness of the values of those tuples in another attribute. Considering these rules, a few rows are created for each incomplete row and placed in the set of candidates for that row. Then from each set one row is selected such that they are not incompatible with each other. These selections are made by an integer linear programming (ILP) model. In this thesis, first, we propose an algorithm to generate DDs. Then in order to improve the previous approaches to increase the percentage of imputation, we suggest fuzzy relaxation that allows a little violation from DDs. Finally, we propose a multi-objective fuzzy linear programming to reach an imputation with more percentage of imputation in addition to decrease the summation of violations. A variety of datasets from “Kaggle” is used to support our approach.

DEDICATION

Dedicate to

My lovely wife, Effat,

who I could never be successful without her support, patience and encouragement

and to

My dear daughter, Yasamin,

who makes my life meaningful and I'm proud of her. It was a memorable event that my daughter and I were students at the University of Windsor, simultaneously.

ACKNOWLEDGEMENTS

I would like to sincerely thank my co-supervisors, Dr. Shahpour Alirezaee and Dr. Majid Ahmadi, for their guidance and support in successfully completing my thesis. I am deeply grateful for their involvement, guiding, mentoring, and providing any help that I needed to complete my degree. It is my honor to have worked under their supervision. I am grateful to thank my committee members, Dr. Mohammad Hassanzadeh and Dr. Jalal Ahamed for their encouragement, constructive and valuable comments in addition to positive criticism which in fact, improved my ideas and solutions.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS/SYMBOLS

AE	Absolute Error
CC	Correlation Coefficient
DI	Data Imputation
DD	Differential Dependency
DM	Decision Maker
FLP	Fuzzy Linear Programming
FDD	Fuzzy DD
FD	Functional Dependencies
HR	High Rate
ILP	Integer Linear Programming
KNN	K-Nearest Neighbor
LHS	Left-Hand-Side
LP	Linear Programming
LR	Low Rate
MR	Medium Rate
MCAR	Missing Completely at Random
MAR	Missing at Random
MNAR	Missing not at Random
RMS	Root Mean Square
NRMS	Normalized Root Mean Square
RHS	Right-Hand-Side
YE	Years of Experience
WS	Weekly Salaries
DE	Degree of Education
MOP	Multi Objective Programming
AOS	Alternative Optimal Solution
ZM	Zimmermann Method
IZM	Improved Zimmermann Method
w.r.t	With respect to

$\mathfrak{R}$	The set of real numbers
T_p	True positive
F_p	False positive
T_N	True negative
F_N	False negative
IM	Number of imputed cells
Δ	Number of missed data
χ_A	Characteristic function of a set A
$\tilde{A}$	Fuzzy Set
$\mu_{\tilde{A}}$	Membership function of the fuzzy set $\tilde{A}$
$lev(c,d)$	Levenshtein distance of two categorical data
$\boldsymbol{\Sigma}$	The set of all Differential dependencies
$(X_i, X_k) \asymp DD_t$	Two tuples X_i and X_k are compatible w.r.t. DD_t
$(X_i, X_k) \vDash \boldsymbol{\Sigma}$	Two tuples X_i and X_k are compatible w.r.t. all DDs in $\boldsymbol{\Sigma}$
$U_w^i \succ U_t^i$	The candidate U_t^i is dominated by the candidate U_w^i
X^C	The rows of matrix X that are complete
X^I	The incomplete rows of matrix X
$\tilde{\to}_\alpha$	A fuzzy deduction with the degree of satisfactory α
$(U_t^i, U_k^l)_{\alpha_{itlk}^w} \asymp DD_w$	Two tuples X_i and X_k are compatible w.r.t. DD_t with the degree of satisfactory
$\widetilde{Max}$	Fuzzy maximization
$\tilde{\leq}$	Fuzzy inequality

CHAPTER 1
INTRODUCTION TO DATA IMPUTATION

1.1 Missing Data

Nowadays, data collection, storage, and analysis have become vital for various processes in estimating, forecasting and decision making. From business, economy, marketing, agriculture, engineering, industry and technology to healthcare, medical and social sciences, and in politics and military, all are involved in a vast processing amount of data. With increase in importance and complexity of data analysis, data quality has become one of the fundamental challenges. Also, for machine learning applications, high level of data quality are crucial to ensure strong prediction and decision making.

In addition to outlier data, one of the most common issues is missing data. The datasets, unusually, have some hidden, incomplete, or missing data for various reasons including imperfect procedures of manual data entry, incorrect measurements, equipment errors, sensor failures, omitted entries in datasets, and ignored responses in questionnaires. In many cases, incomplete or missing data can have a significant effect on statistical analysis and its results. It reduces the power of analysis, forecasting, estimating, and decision making. To be more precise, missing data poses a threat to the validity of scientific research [1].

According to Little and Rubin [2], missing data mechanism can be divided into three parts:

- Missing Completely at Random (MCAR),
- Missing at Random (MAR),
- Missing not at Random (MNAR).

When the probability of being missing is the same for all cases and the causes are not related to the type of data and the value of other data it referred to as MCAR. In this case, missing is independent of the observed and unobserved data. MCAR means there is no relationship between the absence of the data and any values; be it observed or missing. It's just missing and there is no logic for it. When data are MAR, the fact that the data are missing is systematically related to the observed but not the unobserved data. In this case, the lack of data may be predicted by other features in the dataset. In MNAR missing is systematically related to the unobserved data. That is, the missing is related to events or factors which are not measured by the researcher.

In cases MCAR and MAN, if the number of missed data is less than 10% of whole dataset, sometimes we can ignore or delete the missed parts [3]. However, there are situations in which, small amounts of missing data may contain important information that may not be ignored. For instance, let us consider the case where there are more than 10 attributes for each customer in a dataset of a big store. Suppose that a few number of customers have high amount of money spent in the store, while their age or sexuality are missed. Ignoring these customers in the data analysis causes a reduction in the validity of the results.

1.2 Missing Data Imputation

One of the most important tasks of data cleaning is to account for missing data imputation or for short data imputation (DI). DI is the process of filling missing data with estimated values. In general, missing values can be replaced by the values of others in the sample which may have a value (hot-deck) sampling or uses values from a different dataset (cold-deck sampling).

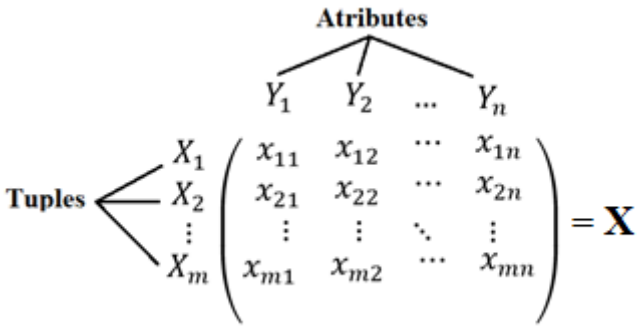
Using mean, mode and median are some common, simple, and of course, naive techniques to estimate missing data.

As a simple example of a dataset with missing data, consider table 1.1 in which gray cells show missing data. The rows are called variables or tuples and the columns are called features or attributes. In column Y6, Education Degree, the numbers 1,2,3,4 and 5 means Diploma or lower, College, BSc, MSc, and PhD, respectively.

Table 1.1 a dataset with some missing data

	Y1	Y2	Y3	Y4	Y5	Y6	Y7
	Name	eage	Street	House Rent	Years of Experience	Education Degree	Weekly Salary
X1	J. Adams	42	Jordan. Rd	3500	2	1	1050
X2	E. Smit.	38	Steels Av.		15	2	
X3	B. Jones	32	Toms St.	1800	12	1	1650
X4	E. Johnny	45	Finch St.	2000	25	2	2100
X5	R. Sadri	44	Jordan. Rd		14	2	1950
X6	M. Ahmad		Finch St.	2200	9	1	
X7	C. Jones	26	Toms St.	1750	6	2	1750
X8	W. Acord	43	Steels Av.	2100	16	3	2200
X9	B. Cooper	38	Jordan. Rd	3200	11	3	2100
X10	S. Brown		Toms St.		28	5	4850
X11	H. David	35	Jordan. Rd	3400	15	3	2200
X12	K. Shaker	46	Steels Av.	1800	18	4	2500

We can show this data sets as the following matrix in which the entries x_{ij}, rows X_i and columns Y_j denote data, tuples and attributes, respectively.



In general, we have two kinds of imputation. They are single imputation and multiple imputation.

1.2.1 Single Imputation

In a single imputation, a single value is estimated for each null cell and, therefore, a single completed or estimated row is produced. Consider the incomplete row X2 in Table 1.1.

The house rent, x_{24}, and the weekly salary, x_{27}, are missed for E. Smit. If one were to use the mean of all weekly salaries, the estimated value for x_{27} would yield \$2235. On the other hand, if one were to consider only those people whose years of experience or degree of education are the same as E. Smit, we have four candidates \$2,100, \$1,950, \$1,750 and \$2,200 with an average of \$2,000. In order to estimate E. Smit's house rent, it is possible to consider the global average or the local (neighborhood) average in his or her living area, as a criterion. Since \$2,100 and \$1800 are the amounts in Steels Avenue, the average \$1,950 in this area would be a feasible estimation for x_{24}! In this example, local (neighborhood) or global average were used as a way for imputation. Indeed, there are several other methods to substitute alternative values for x_{24} and x_{27}.

When the number of missing data is less than 5% of the entire dataset, single imputation is recommended because the results of data analysis will not change much. If the amount of missing data is significant, deletion and single imputation may be problematic because it cannot reflect uncertainty about the estimated values [4].

1.2.2 ***Multiple Imputation***

Multiple imputation can be used to address the shortcomings of single imputation. Instead of filling in a single value for each missing value, multiple imputation procedures replace each missing value with a set of possible values that represent the uncertainty in the missing value [5]. The goal of multiple imputation is not to come up with "the right value" for each missing value. Rather, multiple imputation attempts to produce datasets that provide statistically valid inferences of parameters, such as confidence intervals, based on the incomplete data. By chance, single imputation may yield lower or higher values than would be expected based on the raw mean and standard deviation. By doing multiple imputation those noise can be averaged out. So, we recapture the mean and variance as well to make multiple datasets to mitigate the effect of any bad guesses [4].

Consider again Table 1.1 and the components of X2 as the attributes of E. Smit. Let us assume that we have two candidates \$2,000 and \$2,235 for x_{27} and two candidates \$2,100 and \$1,800 for x_{24}. So the following four candidates may be generated for X2:

E. Smit, 38, Steels Av. 1800, 15, 2, 2000

E. Smit, 38, Steels Av. 2100, 15, 2, 2000

E. Smit, 38, Steels Av. 1800, 15, 2, 2235

E. Smit, 38, Steels Av. 2100, 15, 2, 2235

1.3 Data Imputation based on Differential Dependencies

Depending on the type of data and missingness, different methods have been developed and have been gradually improved in disjoint categories. Some methods pay attention to the relations between variables (tuples) and/or attributes. These relations may be functional or logical. Among them, the method based on differential dependencies (DDs), is a combination of functional and logical relations between attributes considering the differences between tuples. In this approach, it is assumed that the distance between the values of an attribute is affected by the distances between the same values in one or more other attributes. As an example in Table 1.1, the distance (difference) of *weekly salaries* (WS) is affected by the distance between *years of experience* (*YE*) and the distance between *degree of education* (*DE*), meaning for every two persons whose YE and *DE* have a distance less than or equal to *3* and *1*, respectively, WS difference must be less than or equal to *$300*, weekly.

DDs are conditional rules containing a few propositions with conjunction operators as the antecedent (*if statement*) and another proposition as the consequence (*then statement*). The propositions are made by inequalities for the absolute differences between data.

Song et. al. [6, 31] use DDs to nominate one or more candidates for each missing data. Then they generate a set of rows U^i as candidates for each incomplete tuple X_i. Considering DDs, those candidates which are incompatible with the current complete tuples, are then removed from each U^i. Then, one of the candidates in each updated U^i is selected such that all selected candidates from all U^i are pairwise compatible w.r.t. DDs. These selections are made using an integer linear programming (ILP) model. In the rest of this dissertation, we say the Song's method, instead of the method proposed by Song et al. [6, 31], for simplification.

1.4 Challenges

Two most challenges in this method are finding DDs and solving the related optimization problem. In some datasets, a few DDs may already be known due to the nature and the

content of the data. Sometimes mathematical, physical, or in general scientific relations can determine DD rules. Certain organizational rules, administrative regulations, or policies may be available in some datasets and we can use them to identify some DDs. Conversely, high quantities of reliable previous experiences may be able to determine some DDs. However, without pre-determined rules, identifying DDs is a complex task that requires an in-depth analysis of the entire dataset. Sometimes we may not be able to reach absolutely certain rules.

The number of constraints and variables in the relevant optimization model is highly sensitive to changes in the number of missing data and their estimated candidates. A small set of candidates would reduce the likelihood of finding completed rows that conform to DDs. Therefore, the model is forced to select candidates' row that may have more null cells. Accordingly, a large set of candidates for incomplete rows would yield a higher number of constraints and variables, hence disrupting the model.

1.5 Objectives and Novelties

The main objective of our method in this dissertation is to improve the Song's method in order to increase the number of imputations. During the Song procedure of selecting a row from each updated U^i, consider the case where there are rows in U^i that have more imputed cells, but they have a slight violation from DDs w.r.t the previously selected rows. So, because of the policy, the model has to ignore them and select another row without violation even though it has fewer filled cells. So, in order to do more filling, we use fuzzy flexibility in DDs with small violations. We propose a fuzzy bi-objective ILP model in which one of the objective functions is going to increase the number of imputations, and the other seeks to reduce the sum of violations. To solve this model, we use the Improved Zimmermann Method (IZM) proposed by Safi et al. [7].

In order to confirm with the Song's method, we need the same DDs for both methods. For this reason, in Chapter 4, we propose a heuristic method to generate DD rules.

In our proposed method, in addition to apply relationships and rules that are already known, we use the correlation coefficient between the tuples' differences in each attribute with tuples' differences in the other attributes. Also, in order to prevent a dramatic increase in

the number of constraints and variables, we use the k-nearest neighbors' method to limit the number of candidates for each incomplete row.

1.6 Research Questions

The main questions of this research are as follows

1. Will our method increase the percentage of imputed cell?
2. Will our method increase the percentage of completed rows?
3. What is the impact of our method on the value of F-Measure?
4. What is the impact of our method on the value of NRMS?
5. Is there any relation between possible increasing in imputations and possible increasing in violations?

1.7 Datasets

In this thesis we use some datasets from Kaggle that is a public site containing so many datasets with quantitative (numerical) data and qualitative (categorical, string) data with both nominal and ordinal types. Some datasets have just one kind of these data and some have a combination of all types. Here we have chosen those datasets containing numerical and/or ordinal categorical data. Some datasets contains only integer numbers, some have only decimal numbers and some of them are the combination of both.

1.8 The Structure of Thesis

The organization of this thesis is as follows.

Chapter 2 deals with the background. It starts with different definitions of distances and continues with the DI and optimization background in two separate sections. In Section 2.3 after some general explanations about different approaches for DI, the famous KNN method is discussed. Kinds of reliability measures such as RMS, NRMS, AE, Precision, Recall and F-measure are the other subjects in this section. In the optimization background section, single-objective and multi-objective models, linear programming (LP) and integer linear programming (ILP), Pareto optimality, fuzzy sets and fuzzy linear programming (FLP) and the IZM algorithm for solving FLP problems are recalled. The final section of this chapter has a review of the related literature.

In Chapter 3, DI based on DDs using ILP is discussed. At the beginning of this chapter, some definitions and notations are recalled. The method of finding candidates for null cells,

generating candidates for incomplete rows and the refinements of candidates based on DDs are explained in the next section of this chapter. The fourth and fifth sections of this chapter deal with imputation using an ILP model and the ROUND algorithm [6,31] based on DDs. The pros and cons of the method proposed in this chapter are discussed in the final section.
Our proposed method is explained in Chapter 4. After an introduction, we explain about some challenges about DDs discovering and then we present our heuristic method to create DDs in the third section of this chapter. In the next three sections, in addition to some discussion about the $\alpha - satisfactory$ and creating candidates for incomplete rows based on Fuzzy DDs (FDDs) we use fuzzy relaxation to convert DDs to FDDs. In the last section of this chapter, we propose an FLP model and the FROUND Algorithm to achieve the maximum imputation with a specific average satisfaction of DDs.
The experimental results of our proposed method are illustrated in Chapter 5. We have used our method in several datasets selected from the Kaggle site and have compared our results with the output of the ROUND Algorithm and the KNN method.
Chapter 6 contains the conclusion and future works.

CHAPTER 2
STATE of THE ART

2.1 Introduction

In this chapter we are going to provide some necessary backgrounds that are needed to this thesis. Also in the last section we have a brief review on some related articles. The necessary backgrounds are divided in two parts DI Backgrounds and Optimization Backgrounds.

2.2 Distance Functions

In order to compute the distances between data, we can use different distance functions or simply distances. Let $\mathfrak{D}$ be a set of data all from the same kind. A distance function d over $\mathfrak{D}$ is a real valued function, $d: \mathfrak{D} \times \mathfrak{D} \to \mathfrak{R}$ with the following properties:

1- $d(A,B) \geq 0$, for all $A, B \in \mathfrak{D}$
2- $d(A,B) = 0$ iff $A = B$
3- $d(A,B) = d(B,A)$ for all $A, B \in \mathfrak{D}$
4- $d(A,B) \leq d(A,C) + d(C,B)$ for all $A, B, C \in \mathfrak{D}$

Depends on the members of $\mathfrak{D}$, we can use different distances. These are some examples:

- If the members of $\mathfrak{D}$ are real numbers, the absolute value is the most common distance, i.e. $d(x,y) = |x - y|$, for all $x, y \in \mathfrak{D}$.
- If the members of $\mathfrak{D}$ are real valued vectors with the same dimensions, the LP norms are most common, i.e.

$$d(X,Y) = \|X - Y\|_p = (\sum_{i=1}^{n} (x_i - y_i)^p)^{1/p}, \qquad (2,1)$$

For all vectors $X = (x_1, \dots, x_n)$ and $Y = (y_1, \dots, y_n)$ in $\mathfrak{D}$. The LP norm for $p = 2$ is the Euclidian norm.

- If the members of $\mathfrak{D}$ are $m \times n$ real matrices, the following matrix norm is used

$$d(X, Y) = \|X - Y\|_p = (\sum_{j=1}^{m} \sum_{j=1}^{n} (x_{ij} - y_{ij})^p)^{1/p} \qquad (2,2)$$

- Sometimes the members of $\mathfrak{D}$ are vectors or matrices of ordinal data (qualitative or categorical data for which the values can be sorted). In this case, regarding the order, we can assign suitable real numbers to each data and use one of the above distances. As an instance let $\mathfrak{D}$ be a set of education degrees, *Diploma*, *College*, *BSc*, *MSc*, *PhD*, we can consider the real vector (1,2,3,4,5) instead, or every other five dimensional real vector with increasing components.
- One of the most common distances for nominal data (qualitative or categorical data for which the values cannot be sorted) is the Levenshtein distance [14]. The Levenshtein distance that may also be referred to as *edit distance*, is defined between two string or nominal data X and Y, denoted by $lev(X, Y)$. In the case length of X and Y are different, it is equal to the number of position that X and Y have symbol(s) but they are different plus to the difference of their length. For example $lev(Martin, Martina) = 1$, $lev(Marvin, Martina) = 2$, $lev(Martin, Catrina) = 4$. Levenshtein distance for two equal length string X and Y is called the Hamming distance [15]. In this case, it is equal to the number of positions that X and Y are different.

2.3 Data imputation background

In this section, we recall only those parts of DI that apply in this thesis. However, in some parts, we may discuss it a little more in detail. After a review on some basic method, the KNN method, which is one of the most common method on DI, is explained. Finally we discus about some common formula as criteria for measuring reliability of imputation methods.

2.3.1 Imputation methods

As mentioned in Chapter I, one of the standard approaches to missing data is still to delete missing values, especially if values are MCAR and MAN, and the percentage of them is

less than %10 of whole dataset. Although, methods as Mean, Median and Mode imputation are among the simplest and fastest methods, however, they do not recommended as reliable methods. In these kinds of imputation, a missed data in an attribute replace with the mean, median or mode of the other data in the same attribute, respectively. Obviously, mean only is used for numerical data.

Depends on missingness and datasets, different approaches of DI are developed. Choosing suitable approach for each dataset need a deep analysis on data, finding possible relation between attributes, diversity and type of data, possible categories in dataset and using the experience of experts in the related field or subject.

Table 2.1 a dataset to examine the relations between attributes

	Y1	Y2	Y3	Y4	Y5
X1	1	7.333333	7	6	3.94
X2	1.35	8.152625	7.7	5	6.115
X3	1.7	8.567667	12.4	7	8.94
X4	2.05	8.664208	13.1	4	10.41
X5	2.4	8.528	13.8	8	7.705
X6	2.75	7.244792	12.5	3	4.645
X7	3.1	6.900333	11.2	6	5.645
X8	3.45	7.580375	16.9	4	8.47
X9	3.8	6.370667	13.6	6	4.175
X10	4.15	7.356958	15.3	6	6.41
X11	4.5	6.625	15	8	9.47
X12	4.85	8.260542	15.7	6	10.645
X13	5.2	8.349333	19.4	5	3.47
X14	5.55	9.977125	16.1	7	8.705
X15	5.9	12.22967	16.8	3	5.115
X16	6.25	16.19271	18.5	5	6.115
X17	6.6	19.952	23.2	3	7.175
X18	6.95	24.59329	18.9	9	3.47
X19	7.3	30.20233	23.6	3	6.175
X20	7.65	35.86487	20.3	3	4.94

As an example, consider the dataset in Table 2.1. The scatter plot of attributes Y1 and Y2, and the scatter plot of attributes Y1 and Y3 are graphed in Figures 2.1 and 2.2, respectively.

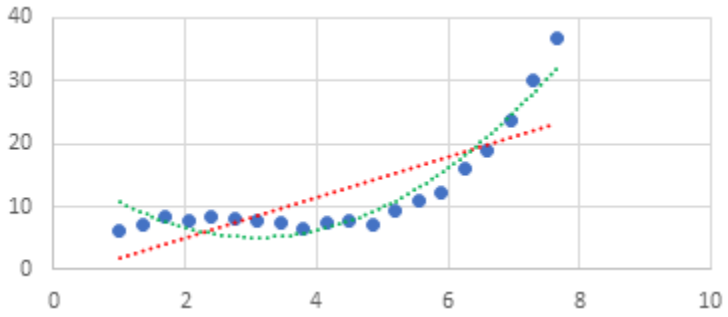


Figure 2.1 the relation between attributes Y1 and Y2 in Table 2.1

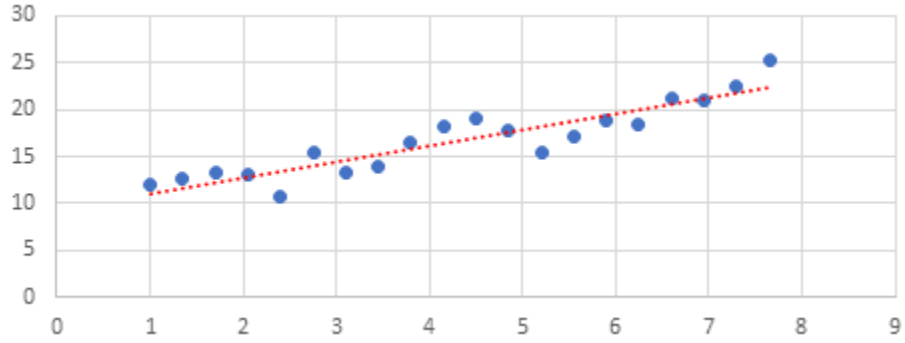


Figure 2.2. The relation between attributes Y1 and Y3 in Table 2.1

Assume that the value of $x_{12,2}$ is missed. By the average on Y2 we have 12.4473, while the regression line related to columns Y1 and Y2 give us 13.975. Both of these values are far from the correct value 8.260542. In this case a polynomial interpolation lead to a value 8.5 which is so better estimation than the average and the linear regression value. Figure 2.1 illustrates the relation between attributes Y1 and Y2 in Table 2.1.

Now assume that the value of $x_{15,3}$ is not available. The average on Y3 gives us 15.55, while the value of regression line of Y3 and Y1 leads to the exact value 16.8.

A real dataset that is related to the amount of CO_2, daily, weekly, monthly, in Toronto 2020 is illustrated in Figure 2.3, [8]. If we consider weekly or monthly average amounts a third degree approximation is an appropriate way to estimate missing data. On the other hand, as illustrated in Figure 4.2, for the daily amount of CO_2 only in August 2020 [8], estimating regression line is more appropriate.

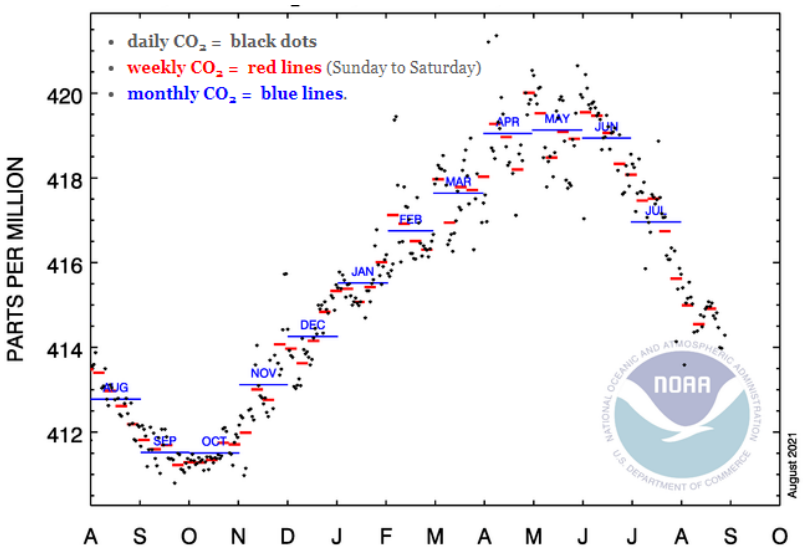


Figure 2.3 The daily, weekly, monthly amount of CO_2 in Toronto, Aug. 2019- Oct. 2020, [8].

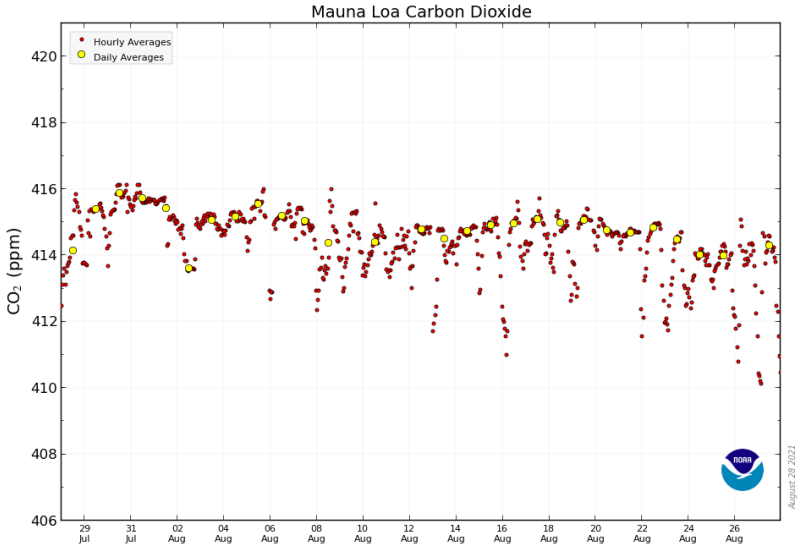


Figure 2.4 The daily amount of CO_2 in Toronto, Aug. 2020, [8].

2.3.2 KNN Method

In this subsection we are going to explain one of the most famous methods in DI that is K nearest neighbor (KNN). This method first proposed by Evelyn Fix and Joseph Hodges in 1951 [9] and later developed by some other researchers in various cases. The first development proposed by (Altman, Naomi S. (1992). [10]. Then expanded by several other

researchers such as H. Schwender [11] for categorical data, and a weighted nearest neighbor imputation method based on Lq distances by Tutz, G., Ramzan, S., 2015 [12]. A review on KNN method is done in [13].

KNN uses for both classification and regression cases. In classification case we consider the most frequency in the neighborhood of missed data. Consider the following example.

Example 2.1 KNN for a classification case.

Figure 2.5 illustrates a simple example of KNN with k=3 and k=5 in classification case. The question mark "?" denotes a missed data. It should be classified either to blue dots or to red stars. By k=3, the neighborhood with 3 objects, the interior circle induces that the question mark,"?", is a blue ball, while k=5 says it is a red star.

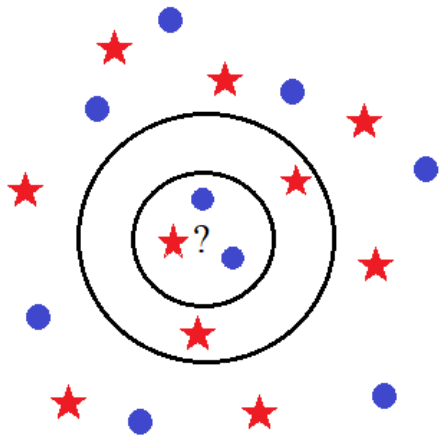


Figure 2.5 KNN for a classification case with k=3 and k=5

Since in this thesis we are dealing with numerical datasets, let us explain KNN method for the dataset in the following example.

Example 2.2 KNN for a numerical case.

Consider the dataset that is shown by Table 2.2 in which the values of x_{71} and x_{24} is missed and we are going to estimate the value of x_{71} by KNN method, with k=2. The row X7 is called the pilot row.

- First ignore all rows with missed data, except the pilot row X7.

- Compute the distance between all remained rows with X7, except X7, by Euclidian norm as follows:

$$d_i = \|\boldsymbol{X_i} - \boldsymbol{X_7}\| = \sqrt{\sum_{j=2}^{5}(x_{ij} - x_{7j})^2}\ , i = 1, \dots, 11, i \neq 7,2$$

- Since k=2, then find 2 smallest d_i.

Table 2.2 the data set to examine KNN for a numerical case

	Y1	Y2	Y3	Y4	Y5
X1	0.067351	3.5337	0.46959	0.67908	3.4344
X2	2.0628	2.9216	0.42472		1.6802
X3	2.4709	3.065	1.5676	1.4824	0.9216
X4	0.58809	1.9563	2.787	2.5957	2.8933
X5	0.4297	1.6312	3.5541	2.2711	0.37425
X6	3.8302	3.4968	0.18482	2.0489	0.61292
X7		0.99051	1.1687	0.24167	3.6203
X8	3.2064	0.06788	3.815	3.1113	2.2949
X9	0.35379	0.38978	3.0439	0.12003	2.7294
X10	0.10072	3.2336	0.22308	2.7215	0.31298
X11	3.3391	3.9263	0.1137	3.571	1.4083

We have d_1 =7.182456, d_3 =13.28502, d_4 =9.621631, d_5 =20.75604, d_6 =19.55992, d_8 =17.84561, d_9 =4.685751, d_{10} =23.01357 and d_{11} =25.70927. Since d_1 and d_9 are two smallest then X1 and X9 are nearest neighbor of X7. Therefore, the values of $x_{11} = 0.067351$ and $x_{91} = 0.35379$ are candidates for x_{17}. We can consider their average for x_{17} that is 0.210571.

Although, the Euclidian norm is most common for identifying the distances, there are several other measures for this purpose, such as other LP norms, absolute value etc.

In the cases that the data are text strings instead of numbers, other measures distances must be used that are called string metrics, in general. These measures use between two text strings in order to compute matching or differences between strings. Among them, Levenshtein distance [14] and Hamming distance [15] are most common.

2.3.3 Measures of Reliability,

As mentioned above, the performance and efficiency of each imputation technique may vary according to the types of missingness and datasets. To test a method, we can consider a complete table of data as a matrix X. Then we randomly delete some elements, and complete the new table by our method as the estimated matrix X_{es}. Then we can compare the estimated table by the original one, via some measures. There are several criteria that are used to measure the reliability of different imputation techniques. In continue some of these criteria are explained.

A. *RMS, NRMS and AE*

The following measures are most common for comparison, particularly when all missing data are imputed.

- The Root Mean Square (RMS) Error, $RMS = \|X_{es} - X\|$, (2.3)
- The Normalized RMS (NRMS), $NRMS = \frac{\|X_{es}-X\|}{\|X\|}$, (2.4)
- The Absolute Error, $AE = \frac{1}{m.n}\sum_{j=1}^{n}\sum_{i=1}^{m} I(x_{ij}^{es} = x_{ij})$ (2.5)

Where x_{ij}^{es} is the estimated values of x_{ij} if it is missed and $x_{ij}^{es} = x_{ij}$, for else. In addition, $I(x_{ij}^{es} = x_{ij}) = 1$, if $x_{ij}^{es} = x_{ij}$ and $I(x_{ij}^{es} = x_{ij}) = 0$, if $x_{ij}^{es} = x_{ij}$

The AE uses for both categorical and numerical data, directly. In terms of RMS and $NRMS$, we can use any appropriate norms. However, LP and Levenshtein norms are most common for numerical and categorical data, respectively.

In this thesis we use $NRMS$ with L2 or Euclidian norm, i.e.

- $NRMS = \frac{\|X_{es}-X\|}{\|X\|} = \frac{\sqrt{\sum_{i,j}(x_{ij}^{es}-x_{ij})^2}}{\sqrt{\sum_{i,j}(x_{ij})^2}}$, $\|X\| = \sqrt{\sum_{i,j}(x_{ij})^2}$. (2.6)

Obviously, NRMS≥0, however, can reach values more than 1 that of course, will be bad news about the imputation technique. Conversely, the lower NRMS amount indicates greater reliability.

B. *Precision, Recall, and F-measure*

In the case that all missing data are imputed by our method, these measures can describe the reliability of the method, perfectly. Now consider two different method such that the first impute %90 of null cells with $NRMS = 0.15$ and the second impute %15 of null cells

with $NRMS = 0.1$. Which one of them is preferred? Although, in terms of $NRMS$, the second is better, the first impute much more than the first. Therefore we need another measure that consider the percentage of imputation beside the $NRMS$.

One of the most common criteria, which is useful when the method cannot impute all missed data, is *F-measure*. This measure consider the accuracy rate and the filling rate, simultaneously and compute as follows

$$F - measure = \frac{2 * P * R}{P + R} \qquad (2.7)$$

where

$P =$ precision $=$ the proportion of filled cells that are correct,

$R =$ recall $=$ the proportion of null cells that are accurately filled.

In the other words, precision is equal to $(\frac{correctly\ filled}{total\ filled})$ and recall is $(\frac{correctly\ filled}{total\ missed})$ and hence they could calculated as follows

$$P = precision = \frac{T_p}{T_p + F_p} = \frac{T_p}{IM}, \qquad (2.8)$$

$$R = recal = \frac{T_p}{T_p + F_N} = \frac{T_p}{\Delta}, \qquad (2.9)$$

where

$T_p = True\ Positive =$ the number of correctly imputed,

$F_p = False\ Positive =$ the number of incorrectly imputed,

$F_N = False\ Negative = \Delta - T_p =$ the number of incorrectly imputed or not imputed,

$IM = T_p + F_p =$ the number of imputed cells,

$\Delta = total\ number\ of\ missed\ data.$

Another simplification leads to

$$F - measure = \frac{2 * T_p}{IM + \Delta}, \qquad (2.10)$$

To better understanding, note to the following example. Example 2.3 contains a dataset with integer data, while the data in the next example are decimal.

Example 2.3 Consider a dataset contains some information about 1000 cameras which are installed at 1000 crosses in Ontario streets. These cameras record the following items, continuously and send them every 2 hours.

- The number of crossing.
- The number of violations crossing the red light.
- The number of crossing when the traffic light is yellow.
- The number of cars that change their lanes, when crossing the intersection.

Because of some noise in recording and/or transmitting the information, some data may be lost. However, a complete 24 hours dataset without any missed data is available. In order to check the efficiency and reliability of some DI methods, some of these data are missed randomly, the methods are implemented to imputation, separately and then the results are compared with the original dataset.

Suppose that we have deleted 1200 data from 12000 existing data and one of these methods has imputed 1100 values such that 900 of them are exactly equal to the original one. Therefore,

$T_p = 900,$

$F_p = 200,$

$\Delta = 1200,$

$F_N = \Delta - T_p = 300,$

$IM = 1100,$

$$P = precision = \frac{900}{900 + 200} = \frac{900}{1100} = 0.8182,$$

$$R = recal = \frac{900}{900 + 300} = \frac{900}{1200} = 0.6923,$$

$$F - measure = \frac{2 * 900}{1100 + 1200} = 0.7826$$

When the data are dealing with integer numbers with relatively small variance, numerating True Positive values is simple. If the imputed value is equal to the original one, it is acceptable and it must numerate as a True Positive, else it must numerate as a False Positive. But, for decimal numbers, the first question to compute the F-Measure is which of the estimations (imputation) should be accepted as a True Positive value. As an instance,

assume that the original data is 5.2368. A DI method has estimated it by 5.2362 and the other by 6.4587. Obviously, the acceptance of the first estimation is too easier than the second. Of course, for the datasets with integer values that have high value variances, the problem is similar. In these cases, determining the acceptable range depends on the decision-maker. The next example shows one of these cases.

Example 2.4 Consider the data in Table 2.3. In order to examine the reliability of some DI methods, some of these data are deleted randomly. Suppose that these deleted data are the highlighted ones in the table. A method imputed the following values instead of the missed data, in order.

0.30, -0.44, 0.3586, $'-'$, 2.6243, 1.97, $'-'$, -0.6381, 0.21854, -0.3543.

The notation $'-'$ shows the method was not able to impute a value instead the missed data. The value of NRMS by this imputation is equal to 0.02379.

Now if we accept the imputation with the maximum violation $\rho = 0.2\sigma_t, 1 \leq t \leq 5$, that means 20% violation from the standard deviation in each attributes, then the third, the sixth and the ninth are not acceptable as the corrected imputation or the *True Positive*. So

$T_p = 5,$

$F_p = 2,$

$\Delta = 10,$

$F_N = \Delta - T_p = 5,$

$IM = 8,$

$$P = precision = \frac{3}{8} = 0.375,$$

$$R = recal = \frac{5}{10} = 0.5,$$

$$F - measure = \frac{2 * 5}{8 + 10} = 0.556$$

Choosing the appropriate value for ρ depends on the decision maker.

Table 2.3 The dataset to compute F-Measure in the continues case, example 2.4

	Y1	Y2	Y3	Y4	Y5
X1	0.369833	2.631172	1.725549	0.583857	1.08
X2	0.506954	2.169096	1.160197	-0.04073	2.61
X3	0.262	2.094885	1.172521	-0.4442	1.68
X4	0.613101	2.178033	0.347419	0.257863	2.92
X5	0.501534	2.598686	1.441513	-0.02511	1.85
X6	0.933681	2.769828	0.012714	0.611315	2.85
X7	0.954985	2.62775	0.189216	-0.20452	2.69
X8	0.121658	2.353946	0.43678	0.27633	1.86
X9	0.406166	2.945584	-0.14247	-0.21711	1.28
X10	0.074583	2.660565	1.171158	0.053895	2.33
X11	0.443355	2.361247	-0.649	0.270604	1.68
X12	0.78744	2.728602	-0.24643	-0.24179	2.54
X13	0.185651	2.896706	0.661956	-0.57098	1.25
X14	0.03077	2.334142	1.5692	-0.29402	2.03
X15	0.137684	2.434395	-1.99405	-0.6349	1.98
X16	0.498083	2.27807	-0.47869	0.545596	2.59
X17	0.823552	2.283397	0.723843	-0.07731	1.67
X18	0.513918	2.654911	-1.53133	0.479326	2.37
X19	0.883171	2.522575	1.02274	-0.11523	2.38
X20	0.816728	2.57832	-1.67419	-0.30039	2.98
Stdv	0.291689	0.237276	1.063292	0.369642	0.563408

Obviously, $0 \leq F-measure \leq 1$, moreover, if the method can fill all blanks, then $\Delta = IM$ and hence $F-measure = \frac{T_p}{IM} = p$.

2.4 Optimization Background

In general, an optimization problem is the problem of finding the best solution between all feasible solutions. These problems can divided to discrete or continuous, linear or nonlinear, single-objective or multi-objective, constraint or unconstraint etc. those parts of optimization problems that will discuss in this thesis are briefly explained in this section.

2.4.1 Single-objective and multi-objective programming

In a single-objective model, the objective function $f(X)$ to be minimized or maximized on a set $S \subset \mathbb{R}^n$. The decision vector $X = (x_1, \dots, x_n)$ is a vector in $\mathbb{R}^n$ and $f: \mathbb{R}^n \to \mathbb{R}$, where $\mathbb{R}$ is the set of real numbers. In this case we write

$$\begin{aligned} & min(max)\ f(X) \\ & s.t. \quad X \in S \end{aligned} \tag{2.11}$$

If $f: \mathbb{R}^n \to \mathbb{R}^k$, the problem is called a multi-objective programming problem. In this case we have $f(X) = (f_1(X), \dots, f_k(X))$ and we can write

$$\begin{aligned} & min(max)\ f_1(X) \\ & min(max)\ f_2(X) \\ & \vdots \\ & min(max)\ f_k(X) \\ & s.t. \quad X \in S \end{aligned} \tag{2.12}$$

2.4.2 Linear and Integer Programming

Consider problem (2.12). If $f(X) = C^T X$, $C^T = (c_1, \dots, c_n) \in \mathbb{R}^n$, $S = \{X \in \mathbb{R}^n : AX(\leq = \geq b), X \geq 0\}$ the problem is called a linear programming (LP) problem. Where $A = (a_{ij}) \in \mathbb{R}^{m \times n}$, $b = (b_1, \dots, b_m)^T \in \mathbb{R}^m$. Note that only the elements of X are unknown. In this case we can rewrite the problem as follows:

$$\begin{aligned} & min(max) \sum_{j=1}^{n} c_j x_j \\ & s.t. \quad \sum_{j=1}^{n} a_{ij} x_j (\leq = \geq) b_i; i = 1, \dots, m \\ & \qquad x_j \geq 0. \end{aligned} \tag{2.13}$$

If we have $x_j \in \mathbb{Z}$, the set of integer numbers, instead of $x_j \geq 0$ the problem is called an integer linear programming (ILP) problem. Also, if we have $x_j \in \{0,1\}$, the problem is called a binary or zero-one linear programming problem.

If the constraints in problem (2.13) create a bounded space, the optimum value is unique and there is at least one $X^* = (x_1^*, \dots, x_n^*) \in S$ as the optimal solution i.e. $\sum_{j=1}^{n} c_j x_1^* \geq \sum_{j=1}^{n} c_j x_j$, for all $X = (x_1, \dots, x_n) \in S$ in the minimum case and the similar result in maximum case with $\leq$.

2.4.3 Pareto optimality in multi objective programming

Since the objective functions, in a multi objective programming (MOP) problem are usually conflict with each other's, then a vector that optimize all functions rarely exist. Then we seek a Pareto optimal solution which is defined as follows.

Definition 2.1 (Pareto optimal solution):

A solution is called Pareto optimal, if none of the objective functions can be improved without degrading some of the other objective values.

Assume that all objectives in Problem (2.12) are maximization. A vector $X^* \in S$ is called a Pareto optimal solution if there is not another $\bar{X} \in S$ such that

$$f_j(\bar{X}) \geq f_j(X^*) \text{ for all } j = 1, \dots, k$$

and

$$f_l(\bar{X}) > f_l(X^*) \text{ for some } 1 \leq l \leq k$$

There are several methods to find Pareto optimal solutions. Among them, weighted sum, *minimax* (*maxmin*), weighted *minimax* (*maxmin*), ε-constraint and interactive method are most common. We explain the weighted sum and ε-constraint when all objective function are in maximization case. The other approaches can find in the wonderful reference [16] that is written by Ralph E. Steuer.

In weighted sum method the decision maker (DM) assign a weight $w_j > 0$ to each objective function $f_j(X), j = 1, \dots, k$ and solve the following problem

$$max \sum_{j=1}^{n} w_j f_j(X)$$

$$s.t. \qquad X \in S. \qquad (2.14)$$

Theorem 2.1 The optimal solution of problem (2.14) is Pareto optimal for all arbitrary $w_j > 0$. On the other hand, for every Pareto optimal solution X^* there exist a non-negative vector weight $W^* = (w_1^*, \dots, w_k^*)$ such that X^* is the optimal solution of the related weighted problem.

In ε-constraint method, the DM select one of the objective function, say$f_l(X)$, as the main objective and assign some lower bounds ε_j for the other objectives $f_j(X), j = 1, \dots, k, j \neq l$ and solve the following problem.

$max\ f_l(X)$

$f_j(X) \geq \varepsilon_j, j = 1, \dots, k, j \neq l$

$s.t. \qquad X \in S.$ (2.15)

Theorem 2.2 The optimal solution of problem (2.15) is Pareto optimal for all arbitrary l and $\varepsilon_j > 0$. On the other hand, for every Pareto optimal solution X^* there exist a non-negative vector weight $\varepsilon^* = (\varepsilon_1^*, \dots, \varepsilon_k^*)$ and $1 \leq l \leq k$ such that X^* is the optimal solution of the related ε-constraint problem.

2.4.4 Fuzzy Sets and Fuzzy Linear Programming (FLP)

- **Fuzzy sets and fuzzy inequality**

Fuzzy sets first were introduced by Lotfi A. Zadeh at 1965 [17]. In the classic case, when A is a subset of the universe U, we have a characteristic function $\chi_A: U \to \{0,1\}$, such that for all $x \in U$; $\chi_A(x) = 1$, if $x \in A$ and $\chi_A(x) = 0$, if $x \notin A$. According to Zadeh's definition, a fuzzy subset $\tilde{A}$ of the universe U is a set of pairs $(x, \mu_{\tilde{A}}(x))$ in which $x \in U$ and $\mu_{\tilde{A}}: U \to [0,1]$ is a function that its value in x, $\mu_{\tilde{A}}(x)$, shows the degree of belonging x in $\tilde{A}$. In the other words, by $\mu_{\tilde{A}}(x)$ we mean the degree of satisfaction from the expression "x is belong to $\tilde{A}$". In fact, $\mu_{\tilde{A}}$ in fuzzy sets plays the rule of χ_A in crisp sets. $\mu_{\tilde{A}}$ is called the membership function of $\tilde{A}$.

Now consider a real value function f in a crisp inequality $f(x) \leq b$ with the solution set $A = \{x: f(x) \leq b\}$. It can be extended to the fuzzy case as $\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) : f(x) \lesssim b\}$ in which the membership function $\mu_{\tilde{A}}$ is a decreasing continues function with $\mu_{\tilde{A}}(x) = 1$ for $x \in \{t: f(t) \leq b\}$, $\mu_{\tilde{A}}(x) = 0$ for $x \in \{t: f(t) > b + \vartheta\}$ and $0 < \mu_{\tilde{A}}(x) < 1$ for $x \in \{t: b < f(t) \leq b + \vartheta\}$, where ϑ is the maximum admissible violation. In the linear case:

$$\mu_{\tilde{A}}(x) = \begin{cases} 1 & f(x) \leq b \\ 1 - \frac{f(x)-b}{\vartheta} & b < f(x) \leq b + \vartheta, \\ 0 & f(x) > b + \vartheta \end{cases} \qquad (2.16)$$

- **Fuzzy linear programming**

In the fuzzy case of the model (2.13), the feasibility constraints $x_j \geq 0, j = 1, \dots, n$ are still crisp, while the other constraints are fuzzy. Also, for the objective function, the goal is to

reach an aspiration level, more than an optimal value. The general form of an FLP is as follows

$$\widetilde{max} \quad z = \sum_{j=1}^{n} c_j x_j$$

$$s.t. \quad \sum_{j=1}^{n} a_{ij} x_j \mathrel{\widetilde{\leq}} b_i; i = 1, \dots, m \qquad (2.17)$$

$$x_j \geq 0.$$

where $\widetilde{max}$ and $\mathrel{\widetilde{\leq}}$ denote the relaxed or fuzzy version of the ordinary max and $\leq$ respectively. In this problem, we are going to find a vector $X = (x_1, \dots, x_n)$ for which the value of the objective function reaches a goal and the constraints satisfy as far as possible. For representing the fuzzy goal, let us stipulate that the objective function cx be essentially greater than or equal to an aspiration level b_0, chosen by the DM. Then we consider the following problem:

$$Find \quad X = (x_1, \dots, x_n)$$

$$s.t \quad \sum_{j=1}^{n} c_j x_j \mathrel{\widetilde{\geq}} b_0$$

$$\sum_{j=1}^{n} a_{ij} x_j \mathrel{\widetilde{\leq}} b_i; i = 1, \dots, m \qquad (2.18)$$

$$x_j \geq 0.$$

For treating fuzzy inequalities, Zimmermann [18] proposed linear membership function as follows:

$$\mu_{\tilde{A}_0}(X) = \begin{cases} 1 & \sum_{j=1}^{n} c_j x_j \geq b_0 \\ 1 - \frac{b_0 - \sum_{j=1}^{n} c_j x_j}{\vartheta_0} & b_0 - \vartheta_0 < \sum_{j=1}^{n} c_j x_j \leq b_0 \\ 0 & \sum_{j=1}^{n} c_j x_j < b_0 - \vartheta_0 \end{cases} \qquad (2.19)$$

$$\mu_{\tilde{A}_i}(X) = \begin{cases} 1 & \sum_{j=1}^{n} a_{ij} x_j \leq b_i \\ 1 - \frac{\sum_{j=1}^{n} a_{ij} x_j - b_i}{\vartheta_i} & b_i < \sum_{j=1}^{n} a_{ij} x_j \leq b_i + \vartheta_i; i = 1, \dots, m \\ 0 & \sum_{j=1}^{n} a_{ij} x_j > b_i + \vartheta_i \end{cases} \qquad (2.20)$$

So we must find a vector $X = (x_1, \dots, x_n)$ such that increase the value of all membership function, as much as possible. By introducing the auxiliary variable λ, Problem (2.18) can be transformed as follows:

$$\max \lambda$$

$$s.t. \boldsymbol{\lambda} \leq \mu_{\tilde{A}_i}(X); i = 0,1, \dots, m \qquad (2.21)$$

$$X \geq 0$$

After some simplification, we have the following equivalent LP problem: (in the sense that the optimal solution for (2.21) is also optimal for (2.18))

$$\begin{aligned} &\max \quad \lambda \\ &s.t. \quad \textstyle\sum_{j=1}^{n} c_j x_j \geq b_0 - (1-\lambda)\vartheta_0 \\ &\qquad \textstyle\sum_{j=1}^{n} a_{ij} x_j \leq b_i + (1-\lambda)\vartheta_i, i = 1, \dots, m \\ &\qquad X \geq 0, 0 \leq \boldsymbol{\lambda} \leq \mathbf{1} \end{aligned} \tag{2.22}$$

2.4.5 Improved Zimmerman Method for Solving FLP

Although, the solution of Problem (2.21) by the *Zimmermann method* (ZM) guarantees the maximum value for λ, yet in the case of existing alternative optimal solution (AOS), it does not guarantee the maximum value for $\sum_{j=1}^{n} c_j x_j$. Safi et al [7] proposed an algorithm to improve ZM that is called IZM. They proved that their algorithm gives the maximum value for both $\boldsymbol{\lambda}$ and $\sum_{j=1}^{n} c_j x_j$. Following their paper on the geometry of ZM [19] they illustrated the difficulties of ZM and the efficiency of IZM, geometrically [20].

Since the first 4 steps in the IZM algorithm is similar to ZM, here we discus steps 5 and 6 which are about AOS case and the fuzzy efficiency.

Let (X^*, λ^*) be the optimal solution of problem (2.22) and the problem has AOS, then solve the following LP problem [7]

$$\begin{aligned} &\max \quad \mathrm{z} = \sum_{j=1}^{n} c_j x_j \\ &s.t. \quad \textstyle\sum_{j=1}^{n} c_j x_j \geq b_0 - (1-\lambda^*)\vartheta_0 \\ &\qquad \textstyle\sum_{j=1}^{n} a_{ij} x_j \leq b_i + (1-\lambda^*)\vartheta_i, i = 1, \dots, m \\ &\qquad X \geq 0, \end{aligned} \tag{2.23}$$

Note that in Problem (2.23) λ^* is not a variable. It is a constant.

If Problem (2.23) is unbounded, then Problem (2.17) does not have any bounded optimal solution. Else, let X^{**} be the optimal solution of (2.23), then $z^{**} = \sum_{j=1}^{n} c_j x_j^{**}$ is the best value for z with the degrees of satisfaction $1 - \frac{b_0 - \sum_{j=1}^{n} c_j x_j^{**}}{\vartheta_0}$ and $1 - \frac{\sum_{j=1}^{n} a_{ij} x_j^{**} - b_i}{\vartheta_i}, i = 1, \dots, m$ for the objective function and the constraints, respectively.

If Problem (2.23) has AOS and we are interested in a fuzzy efficient solution, we can solve the following problem that give us a fuzzy efficient solution as well as it has the best value for $\sum_{j=1}^{n} c_j x_j$. [7]

$$\begin{aligned} \max \quad & \sum_{i=0}^{m} \lambda_i \\ s.t. \quad & \mu_{\tilde{A}_i}(X^{**}) \leq \lambda_i \leq \mu_{\tilde{A}_i}(X); i = 0,1,\dots,m \qquad (2.24) \\ & \sum_{j=1}^{n} c_j x_j = \sum_{j=1}^{n} c_j x_j^{**} \\ & X \geq 0 \end{aligned}$$

2.5 Literature review

Following "A method of estimating the yield of a missing plot in field experimental work" proposed by F. E. Allan and J. Wishart in 1930 [21], as one of the first papers in missing data, this subject extended in various fields with many applications. Due to the importance and wide applications of data analysis, missing data and DI have become one of the hottest research topics. The publication of thousands of articles in this field confirms its importance.

Single imputation methods consider a unique value for each missed data, a single row for each row in database containing missing data and so one completed dataset. Most common single imputation methods are based on mean and mode [22], least square and interpolation [23]. Although, these methods allow us to estimate parameter values, yet they ignore the variety of estimates, which leads to minimizing standard errors and confidence intervals for estimating parameters. It means the single value assigned cannot reflect the sampling variation around the actual value. Multiple imputation overcomes this weakness and generates several values, say M, for each missing value. Therefore, we have M complete dataset that we can estimate their preferred parameters using standard statistical techniques [24]. Multiple imputation was first started with Donald B. Rubin [25], where he considered more than one candidate for each missed data. Rubin represents how to combine both sources to obtain confidence intervals for the estimated parameters.

Two main categories in DI are statistical techniques and machine learning approaches. In statistical methods, we can find numerous articles that use Mean, Mode, Expectation Maximization, Gaussian Mixture Model, Least Squares, Markov Chain Monte Carlo, etc. Artificial Neural Networks, Decision Trees, Clustering, Genetic Algorithm, K-Nearest Neighbor are among the machine learning approaches that have engaged many researchers. W. C. Lin and C.F. Tsai in [26] have reviewed numerous papers in these approaches that have been published from 2006 to 2017.

Some researchers have addressed types of dependencies between attributes and using them in imputation. These relations which, in general, are called functional dependencies (FD) consider kinds of functional relations between the values in an attribute with the values in one or more other attributes. In their famous paper, "*An efficient algorithm for discovering functional and approximate dependencies*", Y. Huhtala et al have pointed to 8 papers on FD and then have proposed their efficient algorithm, TANE [27], to find FDs from large databases. In [28] the authors have reviewed 16 kinds of FD and several related papers including Metric FD, Neighborhood Dependencies, Fuzzy FD, Similarity FD, Matching Dependencies.

Another kind of dependencies are related to the differences between tuples in different attributed which is called differential dependencies (DDs). Song and Chen in [29] first address several theoretical issues of DDs, including formal definitions, differential keys and minimal cover for DDs. Then, they investigate how to discover DDs from a given dataset. Identifying distance thresholds for metric distance constraints is studied by Song et al. [30]. They have proposed an algorithms to determine the distance thresholds having the maximum expected desirability.

In [6] and [31] the authors have suggested three algorithm for imputation based on DDs. They introduce an integer linear programming (ILP) model to achieve a maximum filling regarding compatibility w.r.t. DDs. Since solving ILP models with numerous constraints and variables are difficult, they convert ILP to an LP model and use their algorithms to obtain the final imputation using the optimal solution of the LP model. Paper [31] works for imputing single incomplete attribute, which is the right-hand-side (RHS) attribute of the given DDs. The filling uses the complete left-hand-side (LHS) values to find neighbors and obtain the missing RHS values regarding the DDs. In [6] the authors have extended

the methods proposed in [31] in general cases to fill multiple incomplete attributes including LHS attributes of the DDs.

CHAPTER 3

ENRICHING DATA IMPUTATION BASED ON DIFFERENTIAL DEPENDENCIES

3.1 Introduction

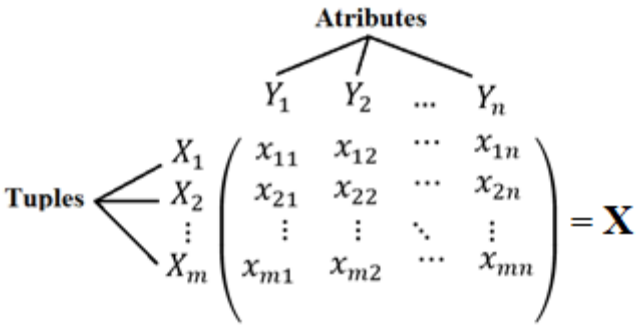
In addition to categorizing or clustering datasets, notice to the possible relations or dependencies between attributes usually leads to more reliable results in DI. Functional dependencies (FDs) are usually defined in terms of equalities. It means for every two tuples, *equality* in the values of one or more attributes leads to the *equality* in the values of another attribute. However, in many datasets, DDs can better describe the relationship between attributes. By DDs, we study the relation between attributes based on the differences between the values of tuples in those attributes. It means for every two tuples, *closeness* in the values of one or more attributes leads to the *closeness* in the values of another attribute. The following examples contain some FD or DD examples.

Example 3.1 Assume that a dataset contains some information about students of a specific university. The students are variables or tuples and student number, name of the department, entering year, name of the academic advisor and passed courses are some of the attributes in this dataset. Based on the information, for every two students if the third and fourth digit of their student ID are the same, then they are studying in the same department. In addition, if they have entered in the same year and are studying in the same department, then they have the same academic advisor. This relations are FDs.

The main purpose of this chapter is to explain the method proposed by Song et al. [6, 31] which is the DI based on DDs. In the second section, the definition of DD, some required notations and incompatibility with respect to (w.r.t) DDs are recalled. In Section 3, the method of generating some candidates for each incomplete row is discussed. These candidates are not incompatible with current complete rows. In the next stage, one and only one of the candidates for each incomplete row must be selected such that they are not incompatible with each other. This is done by an integer linear programming model (ILP) in Section 4. Since solving ILP model with numerous variables and constraints needs much time and memory, Song et al. have relaxed it to an LP model and suggested 3 algorithms to find final imputation. Final section of this chapter is deal with their algorithms.

3.2 Differential Dependencies (DDs)

Before the formal definition of DDs, we need to introduce some notations. Let us show data sets as matrix **X** of the following form in which the components, rows and columns are data, tuples and attributes, respectively.



The matrix $\boldsymbol{X}$ is decomposed to $\boldsymbol{X^I}$ and $\boldsymbol{X^C}$, where the rows of $\boldsymbol{X^I}$ are the incomplete rows of $\boldsymbol{X}$ and $\boldsymbol{X^C}$ is its rest. We denote rows of $\boldsymbol{X^I}$ by $X_1^I, X_2^I, \dots, X_\theta^I$ and rows of $\boldsymbol{X^C}$ by $X_1^C, X_2^C, \dots, X_\mu^C$. In addition, x_{ij}^I and x_{rj}^C, $i = 1, \dots, \theta, r = 1, \dots, \mu, j = 1, \dots, n$ denote the entries of $\boldsymbol{X^I}$ and $\boldsymbol{X^C}$, respectively.

$$X_{m\times n} = \begin{pmatrix} X^I_{\theta\times n} \\ X^C_{\mu\times n} \end{pmatrix}, \theta + \mu = m$$

$$\boldsymbol{X^I} = \begin{matrix} X_1^I \\ X_2^I \\ \vdots \\ X_\theta^I \end{matrix} \begin{pmatrix} x_{11}^I & x_{12}^I & \cdots & x_{1n}^I \\ x_{21}^I & x_{22}^I & \cdots & x_{2n}^I \\ \vdots & \vdots & \ddots & \vdots \\ x_{\theta 1}^I & x_{\theta 2}^I & \cdots & x_{\theta n}^I \end{pmatrix}, \ \mathbf{X^C} = \begin{matrix} X_1^C \\ X_2^C \\ \vdots \\ X_\mu^C \end{matrix} \begin{pmatrix} x_{11}^C & x_{12}^C & \cdots & x_{1n}^C \\ x_{21}^C & x_{22}^C & \cdots & x_{2n}^C \\ \vdots & \vdots & \ddots & \vdots \\ x_{\mu 1}^C & x_{\mu 2}^C & \cdots & x_{\mu n}^C \end{pmatrix}$$

DD is kind of dependency between attributes in which closeness of data in some attributes lead to closeness of data in another attributes. These dependencies are written by conditional rules.

Example 3.2 Considering Table 1.1, we can say: "if two persons have the close years of experience and close degrees of education, their weekly salary should be close". Closeness in these attributes can be interpreted by distances less than or equal to 3, 1 and 300, respectively. On the other hand, if two persons live in the same street, their house rent has a maximum difference $350 monthly. We can write these rules as follows

DD1: $d(x_{i5}, x_{k5}) \leq 3 \wedge d(x_{i6}, x_{k6}) \leq 1 \Rightarrow d(x_{i7}, x_{k7}) \leq 300$

DD2: $d'(x_{i3}, x_{k3}) \leq 0 \ \Rightarrow d(x_{i4}, x_{k4}) \leq 350.$

Where $d(a, b) = |a - b|$ and $d'(c, d) = lev(c, d)$ that is the Levenshtein distance of two strings by its definition in Section 2.2. These two DDs can be written as follows

$$DD1\text{: } (Y5, Y6 \rightarrow Y7 < 3{,}1{,}300 >)$$

$$DD2\text{: } (Y3 \rightarrow Y4 < 0{,}350 >)$$

For two street names, the Levenshtein distance less than or equal to zero means two streets are the same. It can be easily seen that these DDs are true for all complete rows of Table 1.1.

The next example contains some DDs with more than two antecedents.

Example 3.3 In a smart cities project, 90 boxes of equipment are installed in 90 specific points of Ontario province biggest cities, Toronto, Ottawa, Mississauga, Brampton, Hamilton, London, Markham, Vaughan, Kitchener and Windsor. Each box contains a CO_2 Sensor, a Humidity Tester, a Thermometer and a UV Meter. The position of each box is known in terms of latitude and longitude. These electronic devices measure the related amounts every 2 hours and the data are telecommunicated to the main center gradually by a transmitter and then they are collected in a table like as Table 3.1, gradually. It is necessary to mention that measuring the daily UV index is started from 6 am and finished to 6 pm and so the related amounts at the other hours in Table 3.1 is considered zero.

Sometimes during the measurements, because of some problems in electronic devices, the related parameters are not calculated and so we have some missing data. Also, because of noise and other possible issues in data transmission, some data are lost and then we have again some missing data. DDs are determined every 24 hours using the information of the complete rows.

Studies show that at close spatial and temporal distances, there is not much difference in table parameters. Moreover, due to the effect of temperature and humidity on the amount of carbon dioxide [32, 35], in cases where temperature and humidity are not far from each other, the difference between the CO2 indexes is less than a certain limit.

Table 3.1 the data set related to the Smart City in Example 3.3

	Box Name	Latitude	Longitude	Time Hour	Temp. ℃	Humidity %	CO2 Ppm	UV Index mv/cm^2
	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8
X_1	S1	43.856098	-79.337021	12 pm	26	72	412.11	5.5
X_2	S1	43.856098	-79.337021	2 pm	27	75	412.31	6.00
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
X_{12}	S1	43.856098	-79.337021	12 am	21	63	412.46	0.00
X_{13}	S2	43.887501	-79.428406	12 pm	27	71	412.49	6.4
X_{14}	S2	43.887501	-79.428406	2 pm	28	71	412.55	6.8
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
X_{24}	S2	43.887501	-79.428406	12 am	20	65	412.49	0.000
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
X_{1065}	S90	42.31785,	-83.03387	12 pm	31	85	415.96	7.1
X_{1065}	S90	42.31785,	-83.03387	2 pm	32	88	416.00	7.5
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
X_{1080}	S90	42.31785,	-83.03387	12 am	25	81	415.23	0.000

Using the full rows of Table 3.1, the relationships for these differences are set as follows

DD1: $d(x_{i2}, x_{k2}) \leq 0.035 \wedge d(x_{i3}, x_{k3}) \leq 0.15 \wedge d(x_{i4}, x_{k4}) \leq 2 \Rightarrow d(x_{i5}, x_{k5}) \leq 0.83$

DD2: $d(x_{i2}, x_{k2}) \leq .035 \wedge d(x_{i3}, x_{k3}) \leq 0.15 \wedge d(x_{i4}, x_{k4}) \leq 4 \Rightarrow d(x_{i6}, x_{k6}) \leq 0.5$

DD3: $d(x_{i2}, x_{k2}) \leq .040 \wedge d(x_{i3}, x_{k3}) \leq 0.95 \wedge d(x_{i4}, x_{k4}) \leq 6 \Rightarrow d(x_{i7}, x_{k7}) \leq 0.35$

DD4: $d(x_{i2}, x_{k2}) \leq 0.048 \wedge d(x_{i3}, x_{k3}) \leq 0.15 \wedge d(x_{i4}, x_{k4}) \leq 2 \Rightarrow d(x_{i8}, x_{k8}) \leq 0.9$

DD5: $d(x_{i5}, x_{k5}) \leq 4 \wedge d(x_{i6}, x_{k6}) \leq 5 \Rightarrow d(x_{i8}, x_{k8}) \leq 1.4.$

Equivalently

DD_1: $(Y_2, Y_3, Y_4, \rightarrow Y_5 < 0.035, 0.15, 2, 0.83 >)$;

DD_2: $(Y_2, Y_3, Y_4, \rightarrow Y_6 < 0.035, 0.15, 4, 0.5 >)$;

DD_3: $(Y_2, Y_3, Y_4, \rightarrow Y_7 < 0.04, 0.95, 6, 0.35 >)$;

DD_4: $(Y_2, Y_3, Y_4, \rightarrow Y_8 < 0.048, 0.15, 2, 0.9 >)$;

DD_5: $(Y_5, Y_6, \rightarrow Y_8 < 4, 5, 1.4 >)$;

Although some general information are taken from the sites *Canadian Energy Issues* [33], *Canadian Daily UV Index Forecast* [34], CO*2.earth* [8], the number in Table 3.1 are not from none of these resources and are not related to a real world project. We use these sites only for the range of data, some general information and making examples of DDs.

As an instance, the fact that the UV index, ultraviolet radiation intensity, has the yearly rate 0.00-8.50 mw/cm^2 is taken from the site *Canadian Daily UV Index Forecast.* Of course it can be increased to 11.

Now we are ready to define the concept of differential dependency.

Definition 3.1 Let $j_1, \dots, j_p$, t be $p + 1$ distinct members of the set $\{1, \dots, n\}$. A differential dependency DD_t between attributes $Y_{j_1}, Y_{j_2}, \dots, Y_{j_p}$ as bases and Y_t as the target has the following form

$$DD_t: (Y_{j_1}, Y_{j_2}, \dots, Y_{j_p} \rightarrow Y_t < d_{j_1}, d_{j_2}, \dots, d_{j_p}, d_t >);\ \ j_1, \dots, j_p\ , t \in \{1, \dots, n\}, \qquad (3.1)$$

that means

$$DD_t \equiv ([d(x_{ij_1}, x_{kj_1}) \leq d_{j_1}] \wedge \dots \wedge [d\left(x_{ij_p}, x_{kj_p}\right) \leq d_{j_p}] \Rightarrow [d(x_{it}, x_{kt}) \leq d_t]). \quad (3.2)$$

The set of all DDs is denoted by **Σ**.

Definition 3.2

a. If x_{ij} is missed we say that the cell (i, j) is a null cell.

b. Let $DD_t \in \boldsymbol{\Sigma}$ be for an arbitrary $1 \leq t \leq n$. Tuples X_i and $X_k, 1 \leq i, k \leq m$ are called compatible w.r.t DD_t if they meet the relation induced by DD_t. In this case we write $(X_i, X_k) \asymp DD_t$. Conversely, they are incompatible w.r.t DD_t, $(X_i, X_k) \not\asymp DD_t$, if they meet its negation i.e.

$$[(X_i, X_k) \not\asymp DD_t] \equiv ([d(x_{ij_1}, x_{kj_1}) \leq d_{j_1}] \wedge \ldots \wedge [d(x_{ij_p}, x_{kj_p}) \leq d_{j_p}] \wedge [d(x_{it}, x_{kt}) > d_t]) \qquad (3.3)$$

c. X_i and X_k are called compatible, in general, if they are not incompatible w.r.t. none of DDs in $\boldsymbol{\Sigma}$. In this case we write $(X_i, X_k) \models \boldsymbol{\Sigma}$.

During the procedure of generating candidates for rows we have to consider null cells ' $-$ ' as a candidate for missed data to avoid infeasibility in the related optimization model, presented in Section 3.4. So some row candidates contain null cells.

Definition 3.3 A candidate is said to be *fully filled* if none of its cells is null cell.

Remark 3.4 Consider DD_t in relation (3.2). If one of the x_{ij} in the right hand side, antecedent, is a null cell then (3.1) is obviously true and hence $(X_i, X_k) \asymp DD_t$. This is the *false antecedent* case in conditional relations that is always true. However, if x_{it} *or* x_{kt} are missed data the last proposition of (3.3) is false and hence $(X_i, X_k) \not\asymp DD_t$ is not true. In the other words, in this case we do not say X_i and X_k are incompatible. In general, we say that a null cell always agrees a distance restriction.

4.1 Candidates Generation

3.2.1 Candidates for each null cells

Suppose that every attribute containing missing data is the target of a DD. We discuss about the generation of these DDs in Chapter 4. However, assume that related to each Y_t we have a DD_t as demonstrated in relation (3.2). Let x_{kt} be a missed data in Y_t. Then x_{it} is a candidate for x_{kt} if none of $x_{ij_1}, \ldots, x_{ij_p}, x_{it}$ are missed and all propositions in the right

hand side of (3.2) are true. As an instance, consider x_{27} in Table 1.1 and the related rule DD1 that is

$$d(x_{i5}, x_{k5}) \leq 3 \wedge d(x_{i6}, x_{k6}) \leq 1 \Rightarrow d(x_{i7}, x_{k7}) \leq 300.$$

According to this DD, the weekly salaries of Jones, Sadri, Acord and David are candidates for x_{27}, because their years of experience and degree of education satisfy the distance restrictions. In addition, the null cell "-" must be considered as a candidate to prevent the infeasibility in the related optimization model, presented in Section 3.4. So the set of candidates for x_{27} is

$$\text{Can}(x_{2,7})=\{1650,1950, 2200,-\}$$

Similarly, the set of candidates for x_{24} is

$$\text{Can}(x_{24})=\{1800,2100,-\}.$$

3.2.2 Candidates for each incomplete row

After identifying all candidates for each null cell of an incomplete row, we can generate all candidates' row by all combination (cross product) of cell candidates. In the previous example, all candidates for the second row, X2, Smit, are

C_1^2: E. Smit, Steels Av. 1800, 15, 2, 1650
C_2^2: E. Smit, Steels Av. 1800, 15, 2, 1950
C_3^2: E. Smit, Steels Av. 1800, 15, 2, 2200
C_4^2: E. Smit, Steels Av. 1800, 15, 2, -
C_5^2: E. Smit, Steels Av. 2100, 15, 2, 1650
C_6^2: E. Smit, Steels Av. 2100, 15, 2, 1950
C_7^2: E. Smit, Steels Av. 2100, 15, 2, 2200
C_8^2: E. Smit, Steels Av. 2100, 15, 2, -
C_9^2: E. Smit, Steels Av. - , 15, 2, 1650
C_{10}^2: E. Smit, Steels Av. - , 15, 2, 1950
C_{11}^2: E. Smit, Steels Av. - , 15, 2, 2200
C_{12}^2: E. Smit, Steels Av. - , 15, 2, -

3.2.3 Candidates Refinement

In this step, all candidates' row that are incompatible with the current complete tuples must be ignored. In our example C_1^2 and C_5^2 must be ignored due to incompatibility with X8 and X11, due to DD1. So we have 10 candidates for row X2 and denote them by $U_1^2, \dots, U_{10}^2$.

4.2 Imputation Using an Integer Linear Programmin Model

Now we have a set of rows U^i as candidates for each incomplete row X_i^I, i.e.

$$X_i^I \to U^i = \begin{matrix} U_1^i \\ U_2^i \\ \vdots \\ U_{r_i}^i \end{matrix} \begin{pmatrix} u_{11}^i & u_{12}^i & \cdots & u_{1n}^i \\ u_{21}^i & u_{22}^i & \cdots & u_{2n}^i \\ \vdots & \vdots & \ddots & \vdots \\ u_{r_i1}^i & u_{r_i2}^i & \cdots & u_{r_in}^i \end{pmatrix}; i = 1, \dots, \theta$$

In this step we need a mechanism to select one row from each U^i such that the selected rows are compatible w.r.t **Σ**. For this purpose an ILP model is constructed as follows:

- Assign zero-one variable y_{it} to each $U_t^i, i = 1, \dots, \theta; t = 1, \dots, r_i$. $y_{it} = 1$ in the optimal solution means U_t^i is selected for the incomplete row X_i^I.
- Compute parameters $h_{it}, i = 1, \dots, \theta,\ t = 1, \dots, r_i$ as the weights of each U_t^i. h_{it} is the difference between the number of null cells in X_i^I and U_t^i.
- The following constraints lead to select one and only one row from each U^i

$$\sum_{t=1}^{r_i} y_{it} = 1 \qquad 1 \le i \le \theta. \qquad (3.4)$$

- For each U_t^i and U_k^l assign the parameter v_{itlk} as the compatibility parameter such that $v_{itlk} = 1$, if U_t^i and U_k^l are incompatible w.r.t. one of the DDs and $v_{itlk} = 0$ for else.
- Consider the following constraints to prevent incompatibility of the selected candidates

$$v_{itlk}(y_{it} + y_{lk}) \le 1 \qquad (3.5)$$

$$1 \le i < l \le \theta, \quad 1 \le \text{t} \le r_i, \quad 1 \le \text{k} \le r_l$$

Now the optimal solution of the following model, $\mathcal{P}$I, give us a maximum filling without incompatibility.

$\mathcal{P}$I: $Max\ \sum_{i=1}^{\theta}\sum_{t=1}^{r_i} y_{it}\, h_{it}$,

$$s.t.\quad \sum_{t=1}^{r_i} y_{it} = 1 \qquad 1 \le i \le \theta$$

$$v_{itlk}(y_{it} + y_{lk}) \le 1 \quad 1 \le i < l \le \theta, 1 \le t \le r_i, \quad 1 \le k \le r_l$$

$$y_{it} \in \{0,1\} \qquad 1 \le i \le \theta, \quad 1 \le t \le r_i$$

Since the number of variables and constraints are usually too large, solving the ILP model might be impossible and hence the ILP model $\mathcal{P}$I is converted to the similar LP in which $y_{it} \in \{0,1\}$ changes to $0 \le y_{it} \le 1$. In this case, constraints (3.4) could not prevent selecting only one row from each U^i. Moreover, constraints (3.5) could not prevent selecting incompatible candidates from different U^i.

As mentioned in subsections 3.3.1and 3.3.2, every incomplete row is a candidate for itself. Of course the weight h_{it} for this candidate is zero. Assigning $y_{it} := 1$ to this rows and zero to the other rows lead to the value zero for all v_{itlk} and hence we have a feasible solution for the LP model. Since the feasible space is closed and bounded, then the problem has an optimal solution, definitely. Using the optimal solution of LP model, Song et al. have suggested 3 algorithms, ROUND, RANDOM and DERAND, to find an imputation in which all pairs of tuples are compatible w.r.t. **Σ**.

3.3 The ROUND Algorithm

Before starting the algorithm let us first define the concept of dominance.

Definition 3.5 Let U_t^i and U_w^i be two tuple candidates for X^i. U_t^i is said to be dominated by U_w^i and denoted by $U_w^i \succ U_t^i$ if $u_{tj}^i \neq' -'$ then $u_{wj}^i = u_{tj}^i$, and there exist $1 \le z \le n$ such that $u_{wz}^i \neq' -'$ and $u_{tz}^i =' -'$. Otherwise we said that U_t^i is not dominated by U_w^i and denote $U_w^i \not\succ U_t^i$.

In fact, $U_w^i \succ U_t^i$ means U_w^i fills more than U_t^i. The following algorithm, select a row from the set of candidates for each incomplete tuple and constructs a matrix $\boldsymbol{L}$ instead of X^I to complet **X**. However, $\boldsymbol{L}$ might have still some null cells.

ROUND Algorithm

Input : X^I as the matrix of incomplete rows, $\sum$ as the set of DDs, $\boldsymbol{Y} = (y_{it})$ as the vector of optimal solution of PI, the matrix U^i for i^{th} row of X^I.
Output : $\boldsymbol{L}$ as the completed X^I.
For each $i = 1, \dots, \theta$;
    For each $t = 1, \dots, r_i$;
        If $U_t^i \not\succ X_i^I$ then set y_{it} to negative
$\boldsymbol{L} \coloneqq X^I$,
Sort y_{it} in descending order of $y_{it}h_{it}$.
For each $y_{it} > 0$
    If $(U_t^i, \boldsymbol{L}) \vDash \boldsymbol{\Sigma}$, then
        $L^i \coloneqq U_t^i$,
        Set y_{iw} to negative for all $U_w^i \not\succ U_t^i$.
Return $\boldsymbol{L}$

The algorithm first eliminates those tuple candidates U_t^i that have no additional contribution to the current X^I, i.e. $U_t^i \not\succ X_i^I$ cannot fill more over X_i^I. In each iteration, a tuple candidate U_t^i with the maximum $y_{it}h_{it}$ and no violation to the other tuples w.r.t. $\boldsymbol{\Sigma}$ is assigned as L^i. Then, all the other candidates U_w^i that cannot fill more than U_t^i, i.e. $U_w^i \not\succ U_t^i$, could not further contribute to the filling and thus can be pruned by setting y_{iw} to negative.

The other two algorithms, RANDOM and DERAND, first try to make a number of compatible imputations and then use ROUND for the remaining rows. RANDOM makes initial imputations randomly, while DERAND first makes those imputations that satisfy a lower bound condition for a conditional expectation. According to their experimental results, DERAND is much faster than the two others.

Since in this thesis we are going to compare our method with their method only in terms of imputation percentage, so we consider only the ROUND Algorithm. Because all ROUND, RANDOM and DERAND have the same results in this criterion.

3.4 Summary

If the DDs are accurately determined, the Song's method can make a high-precision replacement. But determining the DDs is the biggest challenge of this method. In addition to in-depth analysis of existing data, it requires an expert team that is well acquainted with the data set space and its features. The team must be familiar with all sensitivities to data changes and have extensive experience working with the environment to which the data set belongs.

DDs are very sensitive to data sets. Any change in the amount of data, the number of features and the number of tuples can change the DDs completely. Unlike methods such as KNN, which can be used for any type of data set, the Song's method must go through all the steps from the beginning for each new data set, and all DDs need most likely to be changed.

Determining DDs is time-consuming and complicated. It is also not easy to solve the relevant linear programming problem. Because with a slight increase in the number of candidates, the number of variables and the number of constraints increases dramatically.

When Song et al. talk about computational complexity in their paper, they do not consider the steps of determining DDs and solving the LP problem, but only the ROUND algorithm. While most of the complexity is related to generating DDs and solving the LP problem.

On the other hand, the slightest deviation from the DDs causes their algorithm to select rows with fewer filled cells. As a result, the value of F-Measure will decrease significantly.

In the next chapter, we will explain an innovative way to generate DDs, and then we will try to increase the amount of F-Measure with fuzzy flexibilities. A fuzzy two-objective model helps us to find the maximum imputation and minimize the total deviation from the DDs.

CHAPTER 4

The PROPOSED METHOD

4.1 Introduction

In an ideal missing DI method, we are looking for imputation with a low NRMS and a high F-measure values. These two main objectives are usually in conflict with each other. Moreover, we are interested in methods with less complexity, more speed, and low memory footprint. There is no known approach that would satisfy these criteria for a wide range of datasets.

The main objective of this thesis is to improve the Song's method to increase the number of imputations. In each of their 3 algorithms, the criteria to select a row from each U^i is based on the values y_{ij} and h_{ij}, related to the solution of the Model $\mathcal{P}$I. According to the definition of h_{ij}, bigger h_{ij} means the related row has fewer null cells. Their algorithms prefer a candidate with smaller h_{ij}, even if the candidates with more imputed cells have a very small violation with one of the previously selected rows.

In order to do more filling, we suggest the fuzzy flexibility in DDs with small violations. We propose a fuzzy bi-objective ILP model in which one of the objective functions is going to increase the number of imputations, and the other seeks to reduce the sum of violations. To solve this model, we use the IZM proposed by Safi et al. [7].

In cases where the DDs are completely accurate and the violation of them is not acceptable at all, the method mentioned in Chapter 3 gives a reliable imputation and our proposed method will not give better NRMS and F-measure. Of course, this is rare, and DDs are usually not completely accurate and inflexible. For this reason, a slight deviation from the rules will not necessarily worsen the NRMS and F-Measure values.
In order to a comparison with the Song's method, we need the same DDs for both methods. After some explanation about challenges about DDs in Section 4.2, we propose a heuristic method to generate DD rules in Section 4.3.

4.2 DDs' Challenges

One of the serious challenges in DI based on DDs is identifying DDs. Consider again the general form of DDs in relations (3.1) and (3.2). The number of DDs, the number of attributes in the LHS of each DD, i.e. $Y_{j_1}^d, Y_{j_2}^d, \dots, Y_{j_p}^d$ and the amount of the upper bounds of distances in the LHS and RHS, i.e. $d_{j_1}, d_{j_2}, \dots, d_{j_p}, d_t$ are of most concerns in DDs.
Sometimes, due to the importance of the results in data analysis, the Decision Maker (DM) prefers to have higher accuracy and reliability in imputation than the number of imputed cells, and sometimes he or she needs a bigger population and so more imputation to have better analyzing.

All reliability measures such as NRMS, Precision, Recall, F-measure, and Accuracy have their own importance, however, in different situations, some of these criteria are more important than others. Before starting the explanation about avoidance cases in generating DDs, let us define "the covering by DDs".

Definition 4.1 We say that a DD *covers* ω rows, if all statements in the LHS of the DD is true for each pairs of these ω rows. In this case, the coverage percentage of the DD is $cp(DD) = \frac{\omega}{\mu} * 100$, where μ is the number of rows in X^C.
It should be noted that every DD is true for all pairs of rows in X^C, however some of the propositions in the LHS of some DDs might not be satisfied. As mentioned in Chapter 2, these DDs are true because of the *false antecedent* case in conditional rules.

We should avoid the following cases.

- **DDs with less covering rows:** Consider a dataset with 1000 rows and 10 columns containing 100 incomplete rows. If a DD has namely 8 antecedents, i.e. 8 attributes involved in the LHS, such that only 5 rows of those 900 complete rows satisfy all propositions in the LHS, then it covers only these 5 rows and so it is not a significant DD, definitely. Although it is true for all other 895 rows, the reason is the existence of incorrect propositions in the LHS. So, it is not a valuable DD. Therefore, the coverage percentage of a DD, $cp(DD)$ in Definition 4.1, should be considered in the procedure of DD generation. Considering the missing rate (the percentage of incomplete rows) the reasonable lower bound for $cp(DD)$ should be identified by the decision maker. Those DD with the $cp(DD)$ less than this lower bound will be ignored.
- **Useless or redundant DDs:** A DD for which all pairs of candidates' rows with all possible values for null cells are compatible is a useless or redundant DD. As an instance, if d_t in relation (3.2) is greater than or equal to the range of values in the attribute Y_t, then DD_t is useless. In this case ignoring this DD has no effect on the procedure of imputation.
- **Several DDs:** Existing a number of DDs can cause much time-consuming. Sometimes we can find several DDs with the same target Y_t. In this case we can ignore some DDs which have fewer covering.

4.3 Proposed DD Generation Algorithm

As mentioned in Section 3, the DDs express conditional relations between the differences of tuples' values in some attributes. These relations say that the *closeness* of tuples' pairs in one or more attributes induces the *closeness* of those tuples in another attribute.

Therefore, if we want to discover a rule like Equation (3.2) for the target attribute Y_t, we must focus on the difference of the values in each column and their dependency with the difference of the values in the other columns. So for each column $Y_j;\ j = 1, \dots, n$, we make a dependent column Y_j^d, where their elements are $\left|x_{ij}^c - x_{kj}^c\right|; 1 \le i, k \le \mu$. Now, if Y_t^d and Y_r^d have a correlation coefficient (CC) close to 1, it means that the smaller values in

Y_t^d are related to the smaller values in Y_r^d, and the larger the Y_t^d values, the larger the Y_r^d values. Consequently, the closeness of tuple values in Y_t has a direct relation with the closeness of tuple values in Y_r.

According to the above statement, we are going to generate a conditional rule DD_t for each attribute $Y_t, t = 1, \dots, n$ like (3.2) and (3.3) such that each DD_t must be true for all row pairs of X^C. The DDs have the following form

$$DD_t: (Y_{j_1}^d, Y_{j_2}^d, \dots, Y_{j_p}^d \rightarrow Y_t^d < d_{j_1}, d_{j_2}, \dots, d_{j_p}, d_t >);\ \ j_1, \dots, j_p\ , t \in \{1, \dots, n\},$$

$$DD_t \equiv ([d(x_{ij_1}, x_{kj_1}) \leq d_{j_1}] \wedge \dots \wedge [d\left(x_{ij_p}, x_{kj_p}\right) \leq d_{j_p}] \Rightarrow [d(x_{it}, x_{kt}) \leq d_t]).$$

Then we need to identify $j_1, \dots, j_p$, $d_{j_1}, d_{j_2}, \dots, d_{j_p}$, and d_t for each $t \in \{1, \dots, n\}$.

The following algorithm, DDGEN is our proposed algorithm to create DDs.

In this algorithm, we generate DDs with at most two antecedents for which the bounds of inequalities are a coefficient ρ of the standard deviation of each related attribute. The value of ρ identifies the amount of closeness. In addition, the parameter δ specifies the lower bound of the admissible CC or the lower bound rate of dependency. These ρ and δ are identified by the DM and it can be varied depend on him or her desirability. Moreover, the lower bound of inequality in the RHS, i.e. d_t is chosen from the $minmaxD_{rs}^t$ in line 20 of the algorithm. Although, this criterion can be changed to the amount of the maximum percentage covering by different choices of the pairs r and s, yet we prefer the current criterion because of complexity avoiding.

DDGEN Algorithm

Input: Matrices $\boldsymbol{X}$ and $\boldsymbol{X^C}$.

Output: DD_t for each attribute Y_t in $\boldsymbol{X}$.

1. Take ρ and δ from the DM.
2. For each attribute $Y_t, 1 \leq t \leq n$,
3. For $i = 1, \dots, \mu$
4. For $k = i + 1, \dots, \mu$
5. Set $(Y_t^d)_\gamma \coloneqq d(x_{it}^c, x_{kt}^c)$
6. $\gamma \coloneqq \gamma + 1$
7. For $t = 1, \dots, n$
8. Set $\Omega_t \coloneqq \{u | 1 \leq u \leq n, u \neq t, corr(Y_t^d, Y_u^d) \geq \delta\}$
9. If $|\Omega_t| = 0$, Y_t does not have a strong DD with the other attributes, i.e. $DD_t \equiv NA$. **STOP**.
10. If $|\Omega_t| = 1$, let $\Omega_t = \{r\}$,
11. For all $1 \leq i, k \leq m$
12. If none of $x_{ir}, x_{kr}, x_{it}, x_{kt}$ are blank and $d(x_{ir}, x_{kr}) \leq \rho\sigma_r$, put $d(x_{it}, x_{kt})$ in the set D_r^t, where σ_r denotes the standard deviation of Y_r.
13. Set $d_t \coloneqq maxD_r^t$.
14. $DD_t \equiv (Y_r \rightarrow Y_t < \rho\sigma_s, d_t >)$, **STOP.**
15. If $|\Omega_t| \geq 2$
16. For each pair r and s in Ω_t
17. For all $1 \leq i, k \leq m$ if none of $x_{ir}, x_{kr}, x_{is}, x_{ks}, x_{it}, x_{kt}$ are not blank, $d(x_{ir}, x_{kr}) \leq \rho\sigma_r$, and $d(x_{is}, x_{ks}) \leq \rho\sigma_s$ put $d(x_{it}, x_{kt})$ in the set D_{rs}^t.
18. Set $d_{rs} \coloneqq maxD_{rs}^t$.
19. Let (r^*, s^*) is the pair for which $d_{r^*s^*}$ has the minimum value d_{rs}.
20. Then set $d_t \coloneqq d_{r^*s^*}$ and
21. $DD_t \equiv [Y_{r^*}, Y_{s^*} \rightarrow Y_t < \rho\sigma_{r^*}, \rho\sigma_{s^*}, d_t >]$, **STOP.**
22. For each $Y_t, 1 \leq t \leq n$ **Print** DD_t.

Lines 2 to 6 compute all pair differences in attributes and make the differences column Y_t^d for each attribute Y_t.

Line 7 to 21 generate DDs. There are three cases:

a. The correlations between Y_t^d and the other Y_s^d are not considerable. It means there is not any significant dependency and hence we do not have suitable DD with Y_t^d as the target (line 9).

b. Only one of the other attributes has a considerable impact on attribute Y_t and we have a DD with only one attribute as antecedent (lines 10 to 14). Using a max procedure, we generate DDs with only one antecedents as follows

$$DD_t \equiv ([d(x_{ir}, x_{kr}) \leq \rho\sigma_r] \Rightarrow [d(x_{it}, x_{kt}) \leq d_t]).$$

c. There are more than one Y_j^d that have considerable impact on Y_t^d (lines 15 to 17). Using a $minmax$ procedure in lines 18 to 21, we generate DDs with only two antecedents as follows

$$DD_t \equiv ([d(x_{ir}, x_{kr}) \leq \rho\sigma_r \wedge d(x_{is}, x_{ks}) \leq \rho\sigma_s] \Rightarrow [d(x_{it}, x_{kt}) \leq d_t]).$$

Obviously, by this d_t, $1 \leq t \leq n$, all generated DD_t are true for every row pairs of X^C. The next examples shows how we generate DDs in a small data set.

Example 4.1: Assume that we have a dataset with four attributes and some tuples. Let Table 4.1 represents all complete rows of the dataset. We are going to generate one DD for each attributes as a target.

Table 4.1 Example for DD generation

	Y_1	Y_2	Y_3	Y_4
X_1	2.06735	3.5337	0.46959	0.67908
X_2	2.0628	2.9216	0.42472	1.1785
X_3	2.4709	3.065	1.5676	1.4824
X_4	0.58809	1.9563	2.787	2.5957
X_5	0.4297	1.6312	3.5541	3.2711
X_6	3.8302	3.4968	0.18482	2.0489

First we must make the differences column for each attribute. It is illustrated in Table 4.2.

Moreover, we need the CC of the differences columns. The results of calculation is shown in Table 4.3.

Table 4.2 The differences column for each attributes of Table 4.1

	Y_1^d	Y_2^d	Y_3^d	Y_4^d
$\lvert X_1 - X_2 \rvert$	0.004551	0.6121	0.04487	0.49942
$\lvert X_1 - X_3 \rvert$	0.403549	0.4687	1.09801	0.80332
$\lvert X_1 - X_4 \rvert$	1.479261	1.5774	2.31741	1.91662
$\lvert X_1 - X_5 \rvert$	1.637651	1.9025	3.08451	2.59202
$\lvert X_1 - X_6 \rvert$	1.762849	0.0369	0.28477	1.36982
$\lvert X_2 - X_3 \rvert$	0.4081	0.1434	1.14288	0.3039
$\lvert X_2 - X_4 \rvert$	1.47471	0.9653	2.36228	1.4172
$\lvert X_2 - X_5 \rvert$	1.6331	1.2904	3.12938	2.0926
$\lvert X_2 - X_6 \rvert$	1.7674	0.5752	0.2399	0.8704
$\lvert X_3 - X_4 \rvert$	1.88281	1.1087	1.2194	1.1133
$\lvert X_3 - X_5 \rvert$	2.0412	1.4338	1.9865	1.7887
$\lvert X_3 - X_6 \rvert$	1.3593	0.4318	1.38278	0.5665
$\lvert X_4 - X_5 \rvert$	0.15839	0.3251	0.7671	0.6754
$\lvert X_4 - X_6 \rvert$	3.24211	1.5405	2.60218	0.5468
$\lvert X_5 - X_6 \rvert$	3.4005	1.8656	3.36928	1.2222

Table 4.3 The CC of columns in Table 4.2

	Y_1^d	Y_2^d	Y_3^d	Y_4^d
Y_1^d	1	0.661237	0.608774	0.288802
Y_2^d	0.661237	1	0.844979	0.637873
Y_3^d	0.608774	0.844979	1	0.600511
Y_4^d	0.288802	0.637873	0.600511	1

Taking $\delta = 0.5$ and $\rho = 1$, every $Y_j^d, j = 1, \dots, 4$ has at least two different columns with CC more than 0.5. Regarding lines 19-21 in DDGEN Algorithm, for each attribute we can write a conditional rule as follows:

$$DD_1 \equiv [Y_2, Y_3 \to Y_1 < \sigma_2, \sigma_3, d_1 >]$$
$$DD_2 \equiv [Y_1, Y_3 \to Y_2 < \sigma_1, \sigma_3, d_2 >]$$
$$DD_3 \equiv [Y_2, Y_4 \to Y_3 < \sigma_2, \sigma_4, d_3 >]$$
$$DD_4 \equiv [Y_3, Y_2 \to Y_4 < \sigma_3, \sigma_2, d_4 >]$$

where σ_1 =1.153877, σ_2 =0.728059, σ_3 =1.279838, σ_4 =0.872564.

In order to complete DD_1 we need to identify the value of d_1. Here we have only one pair of indices with the CC more than $\delta = 0.5$. Since the rows 1,2,5,6,9,13 in both Y_2^d and Y_3^d satisfy the condition (the difference $\leq \rho\sigma$), therefore, we must take the maximum of the set {0.004551, 0.403549, 1.762849, 0.4081, 1.7674, 0.15839} which is derived from the same rows of Y_1^d , that is 1.762849. Therefore,

$$DD_1 \equiv [Y_2, Y_3 \to Y_1 < 0.728059, 1.279838, 1.762849 >]$$

Similarly, we have , d_2 =0.6121, , d_3 =1.38278, , d_4 =1.36982 and hence

$$DD_2 \equiv [Y_1, Y_3 \to Y_2 < 1.153877, 1.279838, 0.6121 >]$$
$$DD_3 \equiv [Y_2, Y_4 \to Y_3 < 0.728059, 0.872564, 1.38278 >]$$
$$DD_4 \equiv [Y_3, Y_2 \to Y_4 < 1.279838, 0.728059, 1.36982 >]$$

4.4 Candidate generation based on DDs

In our method, finding the candidates for each null cell and generating candidates for each incomplete rows is the same as the Song's method mentioned in Section 3.3, with a little difference in the candidate's refinement.

By DDGEN Algorithm, for each Y_t we have a DD_t as demonstrated in relation (3.2). Let x_{kt} be a missed data in Y_t. Then x_{it} is a candidate for x_{kt} if none of $x_{ij_1}, \dots, x_{ij_p}, x_{it}$ are missed and all propositions in the right hand side of (3.2) are true.

After identifying all candidates for each null cell of an incomplete row $X_i^I; i = 1, \dots, \theta$, we can generate the matrix U^i by all combination of cell candidates as mentioned in Section

3.3.2. If there is at least one row in U^i that do not have any null cells and is not incompatible with the rows in X^c w.r.t **Σ**, then all rows of U^i that are incompatible with the rows of X^c must be ignored. If all complete rows in U^i are incompatible with the rows in X^c, we keep the one with minimum violation and ignore the others. The minimum violation is equivalent the maximum degree of satisfaction that is explained in Section 4.6.

4.5 Fuzzy relaxation and $\alpha - satisfactory$

The purpose of our method is to increase the number of imputed null cells. In the Song's method, every incomplete row X_i^I has a set of candidates U^i which are sorted w.r.t $y_{ij}h_{ij}$. Some of these rows in U^i are fully filled and some of them have still some null cell(s). Suppose that we are going to select a row from U^i as a candidate for X_i^I. First, we consider the highest row of in U^i, say $U_{k^*}^i; 1 \leq k^* \leq \theta$. If this row is not incompatible with none of the selected rows for $X_1^I, X_2^I \dots, X_{i-1}^I$, we select $U_{k^*}^i$ and go to the next i. Else, we examine the next row of current U^i. Finally, we can find a compatible row. Because in the worst case we have to select the row in which all null cells of X_i^I are again null. This is not incompatible with none of the selected rows, definitely.

Incompatibility occurs due to violation from the inequality $d(x_{it}, x_{kt}) \leq d_t$ in relation (3.2). If we admit a little violation, we likely can select rows with more filled cells (higher h_{ij}). Therefore, we can replace the inequality with a fuzzy inequality, such that more violations lead to less satisfactory degree in the fuzzy concept as explained in the next section.

4.6 Converting DD to FDD

Instead of the crisp inequality $d(x_{it}, x_{kt}) \leq d_t$, we define the following three cases with the degree of satisfactory α:

- $Q_1(t) \equiv d(x_{it}, x_{kt}) \leq d_t, \alpha = 1.$
- $Q_\alpha(t) \equiv d_t < d(x_{it}, x_{kt}) \leq d_t + \vartheta, \alpha = 1 - \frac{d(x_{it}, x_{kt}) - d_t}{\vartheta}$
- $Q_0(t) \equiv d(x_{it}, x_{kt}) > d_t + \vartheta, \alpha = 0.$

where $\vartheta > 0$ is the maximal violation determined by the DM. By

FDD: $(Y_r, Y_s \overset{\sim}{\rightarrow}_\alpha Y_t < d_r, d_s, d_t >)$;

we introduce the following three cases for each pair of X_i, X_K:

- $[P(r,s) \Rightarrow Q_1(t)] \equiv X_i \; and \; X_K$ are completely compatible or $1-$compatible.
- $[P(r,s) \Rightarrow Q_\alpha(t)] \equiv X_i \; and \; X_K$ are $\alpha-$compatible, $0 < \alpha < 1$
- $[P(r,s) \Rightarrow Q_0(t)] \equiv X_i \; and \; X_K$ are completely incompatible or $0-$compatible.

Summarizing all we can say

- $[P(r,s) \Rightarrow Q_\alpha(t)] \equiv X_i \; and \; X_K$ are $\alpha-$compatible, $0 \leq \alpha \leq 1$.

It is noteworthy that existing null cells in the LHS of the rules lead to completely compatibility. In addition, with null cells in the RHS we have not incompatibility and so again we consider $\alpha = 1$ in this cases.

Compatibility of a pairs of rows w.r.t a DD, compatibility of them w.r.t. all DDs in $\boldsymbol{\Sigma}$ and compatibility of a row with all rows of a matrix are denoted as follows:

- $(U_t^i, U_k^l)_{\alpha_{itlk}^w} \asymp DD_w$ means the t[th] candidate for X^i, that is U_t^i, and k[th] candidate for X^l, that is U_k^l, have the degree of compatibility α_{itlk}^w w.r.t. DD_w.
- $(U_t^i, U_k^l)_{\alpha_{itlk}^*} \vDash \boldsymbol{\Sigma}$, where $\alpha_{itlk}^* \coloneqq \min\{\alpha_{itlk}^w : 1 \leq w \leq n\}$ mean U_t^i and U_k^l are $\alpha_{itlk}^* -$ compatible w.r.t. $\boldsymbol{\Sigma}$.
- $(U_t^i, \boldsymbol{L})_{\beta_{it}^{**}} \vDash \boldsymbol{\Sigma}$ means U_t^i is compatible with all rows of L w.r.t. all DDs in $\boldsymbol{\Sigma}$ with the minimum degree of satisfactory β_{it}^{**}, where $\boldsymbol{L}$ is a $\theta \times n$ matrix, $\beta_{it}^{**} = \min\{\gamma_{it}^{l*} : 1 \leq l \leq \theta\}$, $\gamma_{it}^{l*} = min\{\gamma_{it}^{lw} : 1 \leq w \leq n\}$, $(U_t^i, L^l)_{\gamma_{it}^{lw}} \asymp DD_w$ and L^l is the l^{th} rows of matrix $\boldsymbol{L}$.

Although we use the linear membership function for the fuzzy inequality, we can use either of the nonlinear functions, depending on the sensitivity and the importance of DDs for the values greater than d_t. Some of these functions are illustrated in Figure 4.1.

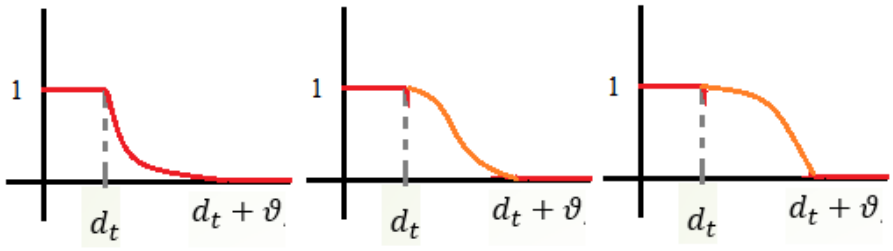


Figure 4.1. Different types of membership function for fuzzy DDs

4.7 Fuzzy Multi-Objective Linear Programming Model to Achieve the Maximum Imputation and the FROUND Algorithm

Our proposed model to find a maximum imputation that is a bi-objective fuzzy linear programming describe as follows

$$
\begin{aligned}
\mathcal{P}\text{II:}\quad \widetilde{Max}\ f(Y) &= \sum_{i=1}^{\theta}\sum_{t=1}^{r(i)} y_{it}\, h_{it},\\
\widetilde{Max}\ g(Y) &= \sum_{i=1}^{\theta}\sum_{t=1}^{r(i)}\sum_{l=1}^{\theta}\sum_{k=1}^{r(l)} \alpha^{*}_{itlk}(y_{it}+y_{lk}),\\
s.t.\ \ k_i(Y) &= \sum_{t=1}^{r(i)} y_{it} = 1 \quad 1 \le i \le \theta,\\
h(y_{it}, y_{lk}) &= (1-\alpha^{*}_{itlk})(y_{it}+y_{lk}) \lesapprox 1,\\
&\qquad\qquad 1 \le i < l \le \theta, 1 \le t \le r(i),\\
&\qquad\qquad\qquad\qquad 1 \le k \le r(l),\\
0 \le y_{it} \le 1 &\qquad\qquad 1 \le i \le \theta, \quad 1 \le t \le r(i),
\end{aligned}
$$

where $\alpha^{*}_{itlk} = \min\{\alpha^{w}_{itlk} : 1 \le w \le n\}$.

Since α^{*}_{itlk} are the degree of DDs' satisfaction, then maximization of g is equivalent to minimization of violations. While the first objective tries to achieve the maximum possible imputation without regarding to possible DDs violations, the second seeks to minimize the violations.

To solve this problem we use IZM algorithm, an improvment of Zimmerman method [18], proposed by Safi et al. [7]. This algorithm garantee the maximum value for the main objective functions of fuzzy model as well as the maximum value for the degree of satisfactory in the equivalent crisp problem.

To convert Model $\mathcal{P}$II to a crisp one, we need first two aspiration levels for f and g. The ideal values for these two functions are Δ and $\varphi = \sum_{i=1}^{\theta}\sum_{l=1}^{\theta} 2\alpha^{*}_{itlk}$, respectively, where Δ is the total number of missed data in the dataset. On the other hand, f and g can or may decrease to zero. We consider 'one unit' as the maximum violation for h.

Now by the linear functions illustrated in Figure 4.2 we can convert Model ($\mathcal{P}$II) to the equivalent LP model $\mathcal{P}$III.

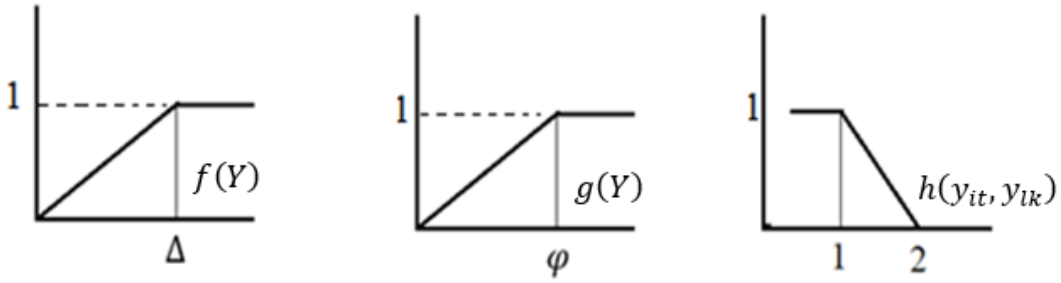


Figure 4.2 Linear membership function for fuzzy objectives and constraints

$\mathcal{P}$III: max λ

$s.t.$ $f(Y) = \sum_{i=1}^{\theta} \sum_{t=1}^{r(i)} y_{it}\, h_{it} \geq \Delta\lambda,$

$g(Y) = \sum_{i=1}^{\theta} \sum_{t=1}^{r(i)} \sum_{l=1}^{\theta} \sum_{k=1}^{r(l)} \alpha^*_{itlk}(y_{it} + y_{lk}) \geq \varphi\lambda,$

$k_i(Y) = \sum_{t=1}^{r(i)} y_{it} = 1 \qquad 1 \leq i \leq \theta,$

$h(y_{it}, y_{lk}) = (1 - \alpha^*_{itlk})(y_{it} + y_{lk}) \leq 2\text{-}\lambda,$

$1 \leq i < l \leq \theta, 1 \leq t \leq r(i),$

$1 \leq k \leq r(l),$

$0 \leq y_{it} \leq 1 \qquad 1 \leq i \leq \theta, \quad 1 \leq t \leq r(i).$

In the case $y_{it} \in \{0,1\}$, the constraint related to $k_i(Y)$ guarantee that one and only one U_t^i is selected for each incomplete row X_i^I. While, $0 \leq y_{it} \leq 1$ do not guarantee the "*only one*" condition. Moreover, consider the case $\alpha^*_{itlk} = 0$. It means U_t^i and U_k^l are completely incompatible. In this case, with $0 \leq y_{it} \leq 1$ selecting both U_t^i and U_k^l is possible that is a wrong selection.

To solve these difficulties, we propose the FROUND Algorithm, the improved version of the ROUND Algorithm suggested by Song et al. in Chapter 3, to reach a maximum imputation with a specific average of compatibility.

This algorithm give us a maximum imputation with the average of possible violation from DDs.

FROUND Algorithm

Input : X^I as the matrix of incomplete rows, $\sum$ as the set of DDs, $\boldsymbol{Y} = (y_{it})$ and $\boldsymbol{\lambda}$ as the vector of optimal solution of model $\mathcal{P}$III, the matrix U^i for i^{th} row of X^I.

Output : $\boldsymbol{L}$ as the completed X^I and $\bar{\alpha}_{\boldsymbol{L}}$ as its average of satisfactory..

1. For each $i = 1, \dots, \theta$;
2. For each $t = 1, \dots, r_i$;
3. If $U_t^i \not\succ X_i^I$ then set y_{it} to negative
4. $\boldsymbol{L} \coloneqq X^I$,
5. Sort y_{it} in descending order of $y_{it}h_{it}$ and in descending order of β_{it}^{**} if there is ties.
6. For each $y_{it} > 0$
7. If $(U_t^i, \boldsymbol{L})_{\beta_{it}^{**}} \vDash \boldsymbol{\Sigma}$, and $\beta_{it}^{**} > 0$ then
8. $L^i \coloneqq U_t^i$,
9. Put (i, t) in the set A,
10. Set y_{iw} to negative for all $U_w^i \not\succ U_t^i$.
11. For each pairs $(i, t)$and (l, k) in the set A put α_{itlk}^* in the vector $V_{A\alpha^*}$.
12. $\bar{\alpha}_{\boldsymbol{L}} \coloneqq$ the average on all members of $V_{A\alpha^*}$,

Return: $\boldsymbol{L}$ and $\bar{\alpha}_{\boldsymbol{L}}$.

In lines 1 to 4 of the above algorithm matrix $\boldsymbol{L}$ is filled by the incomplete rows of matrix $\boldsymbol{X}$**, that is** X^I. The none negative values y_{it} and h_{it} are the optimal values of Model $\mathcal{P}$III and their related weights, respectively. Line 7 starts with most valuable U_t^i, in terms of $y_{it}h_{it}$ and impute all U_t^i that are compatible with all rows of current $\boldsymbol{L}$ w.r.t. all DDs with positive degree of satisfactory. Here, for each row of L, one and only one U_t^i is selected, because the other candidates are ignored by line 10. In order to compute the average of satisfactory degree on selected candidates, first the indices of selected candidates are stored in the set A by line 9. Now recall the definition of α_{itlk}^* from Model $\mathcal{P}$II:

$$\alpha_{itlk}^* = \min\{\alpha_{itlk}^w : 1 \le w \le n\},$$

where α_{itlk}^w is the degree of compatibility for U_t^i and U_k^l, w.r.t. DD_w. Lines 11 and 12 compute the average of satisfactory degree for all pairs of rows in $\boldsymbol{L}$ w.r.t. all DDs in $\sum$**.** The experimental results in Chapter 5 prove the superiority of our proposed method with respect to the Song's method, in terms of F-measure and the percentage of imputed cells and completed rows w.r.t. the same DDs generated by our DDGEN Algorithm.

4.8 Summary

We started this chapter by outlining the challenges of DDs and the things to avoid in producing DDs. Then we presented our own innovative algorithm for generating DDs.
Since DDs are conditional rules based on tuples' distancing and their relationship to different attributes, we formed the difference matrix from the initial dataset. So the initial criterion on DDs' generation is based on the correlation coefficient between the columns of the difference matrix.

In examining the compatibility of the selected candidates w.r.t the DDs, we allowed some minor violations that are controlled with some constants and membership functions. In continue, we introduced fuzzy DDs. Depending on the sensitivity of the violation of DDs, we can consider a variety of nonlinear membership functions. Using the outputs of the DD generation algorithm and the FDD procedure, we presented a fuzzy two-objective model in order to perform the most imputation with the least amount of total violations of the DDs.

This model yields all the possible imputations by the Song model, in addition to as many imputations as possible, with some controlled violations.

Since DDs are created based on current complete rows, a slight violation does not necessarily mean a deviation from the original value. Therefore, a slight violation of the constraints does not mean an increase in the NRMS.

Numerical results in the next chapter confirm the efficiency of our method.

CHAPTER 5

EXPERIMENTAL RESULTS

5.1 The selected Kaggle datasets

Although the main purpose of this dissertation is to compare our proposed method with the Song's method, like most research in this field, we also compare our method with KNN. All programs are coded in Python 3.7.

The datasets was selected from Kaggle, which is a public site. Ten different datasets have selected from this site that have differences in data type and their variances. These datasets are complete and has no missing data. They contain 3 category only integer (Intg.), only decimal (Deci.) and a mixture of integers and decimals (Mix). Some datasets have a small variance, while some others have considerable variances in the values of tuples and/or attributes.

The status of the data set is shown in Table 5.1. In each dataset, the standard deviation of each attribute, $\sigma_1, \dots, \sigma_n$, is calculated. In this table σMAX denotes the maximum of the set $\{\sigma_1, \dots, \sigma_n\}$ and $S\sigma$ denotes the standard deviation of $\sigma_1, \dots, \sigma_n$.

Related to each dataset, there are eight incomplete datasets that the percentage of missing data and its incomplete rows are as follows:

Missing Rate $= (x, y)$, where

$x =$ The percentage of incomplete rows

$y =$ The percentage of null cells

The eight cases of missing rates are (%5, %1), (%10, %1), (%20, %5), (%20, %10), (%40, %20), (%50, %10), (%75, %10), (%95, %20). These pairs are categorized to the following three categories:

- Low Rate (LR): (%5, %1), (%10, %1), (%20, %5),
- Medium Rate (MR): (%20, %10), (%40, %20), (%50, %10),
- High Rate (HR): (%75, %10), (%95, %20),

Table 5.1 the selected datasets from Kaggle

Name	Rows	Columns	type	σMAX	$S\sigma$
4-Gauss	800	12	Deci.	3.63	0.78
Abalone	4177	8	Deci.	0.49	0.14
Bupa	345	6	Intg.	19.48	12.14
Sheart	270	13	Intg.	51.6	14.4
Glass	214	9	Deci.	0.81	0.47
Iris	150	4	Deci.	1.75	0.49
PID	768	8	Mix	115	35
Sonar	208	60	Deci.	0.15	0.05
Wine	178	13	Mix	314.02	83.19
Yeast	1484	8	Deci.	0.14	0.03

Our proposed method, the Song's method and the KNN method are implemented on all 80 incomplete datasets and are compared to their related complete datasets. The comparisons are made in terms of NRMS, F-measure, Percentage of Completed Rows, and Percentage of Imputed Null Cells. Each of these four criteria are performed in two cases datasets and missing rates.

In general, if the relationships between the features are correctly identified and the imputation is based on DDs, our method and the Song's method are expected to have a better estimate with respect to the KNN method. Because this method only considers the distance between the tuples and does not consider the possible relationships between the features.

In data sets with high variance, it is more difficult to find exact relationships between attributes, and hence the methods based on DDs may not work better than KNN.

5.2 The NRMS' comparison

Figure 5.1 illustrate the Average NRMS for three methods in different missing rates. In every three methods the error values (NRMS) rise as the rate of missing data increase. According to the figure in almost cases the FROUND is better than the ROUND, except for the case (%40, %20). But the results for KNN method is different. In the categories (%5, %1), (%10, %1), (%40, %20), (%50, %10) and (%95, %20) it is worse than the methods based on DDs, while in the category (%75, %10) it is better than both of ROUND and FROUND. However it is between ROUND and FROUND at (%40, %20) and (%50, %10).

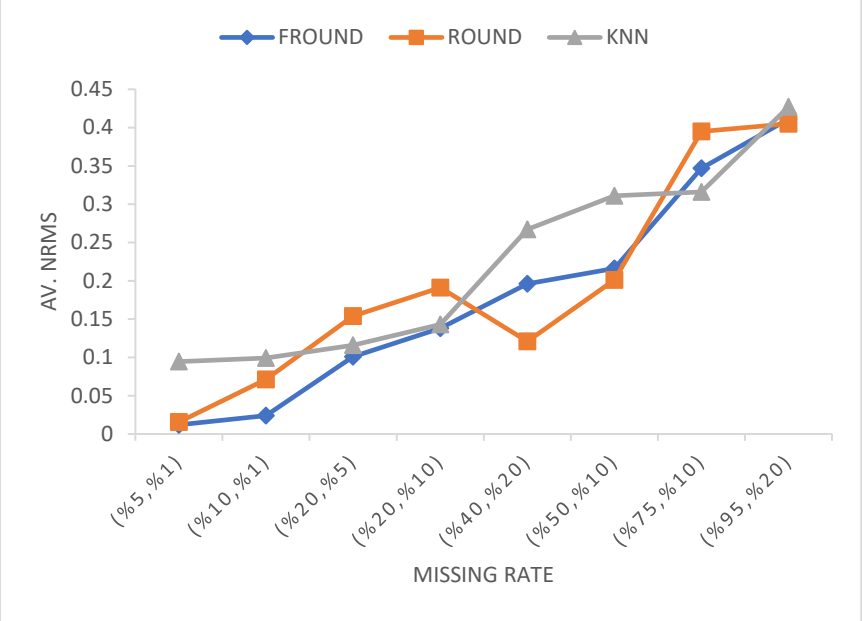


Figure 5.1Comparison of the average of NRMS in terms of different missing rates

As mentioned in Chapter 4, since DDs are created based on current complete rows, a slight violation does not necessarily mean a deviation from the original value. Therefore, a slight violation of the constraints does not mean an increase in the NRMS.

Figure 5.1 confirm this assertion.

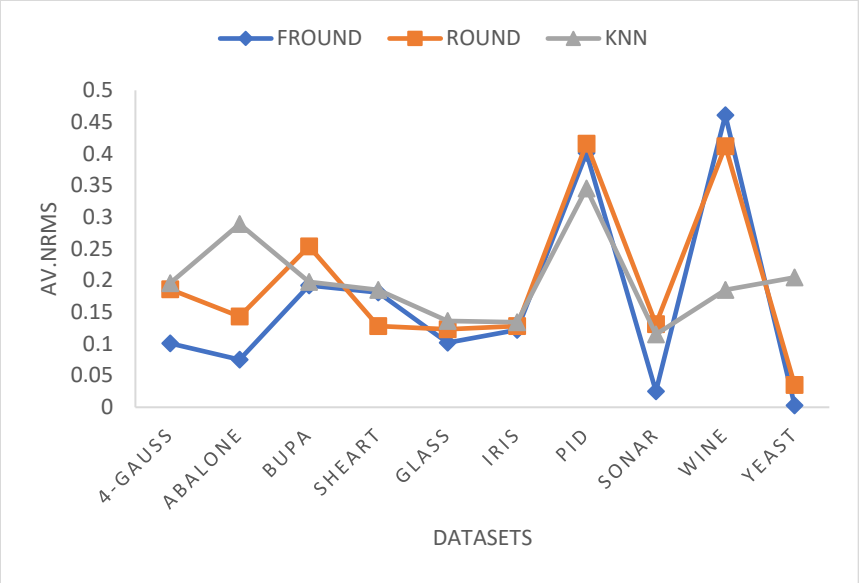


Figure 5.2 Comparison of the average of NRMS in terms of different datasets

The Average NRMS for three methods in terms of different datasets are illustrated in Figure 5.2. These values for KNN are better than the other methods in the data sets PID and WINE. According to Table 5.1 these datasets have very high variances. Since high variances could have negative impact on our method to generate DDs, it is possible that the DDs are less reliable in this case. It can be the main reason for the better results of KNN in these data sets. For YEAST and ABALON it is vice versa. In these datasets the variances are very small and the NRMS in both ROUND and FROUND are better than the KNN. It can show the reliability of DDs.

5.3 The F-Measure's comparison

One of the most important criteria for comparing the imputation methods is the F-Measure. It considers the precision and recall, simultaneously. The precision is the proportion of filled cells that are correct and the recall represent the proportion of null cells that are accurately filled. In fact, F-Measure regards the accuracy rate and the filling rate, simultaneously. However, if the method can impute all missed data with some estimated value, the amount of F-Measure and precision are the same

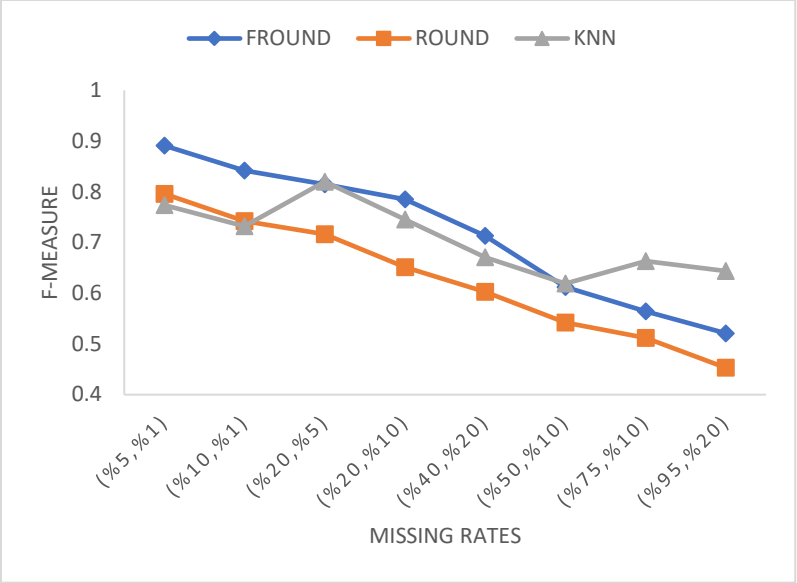


Figure 5.3 Comparison of the average of F-Measures in terms of different missing rates

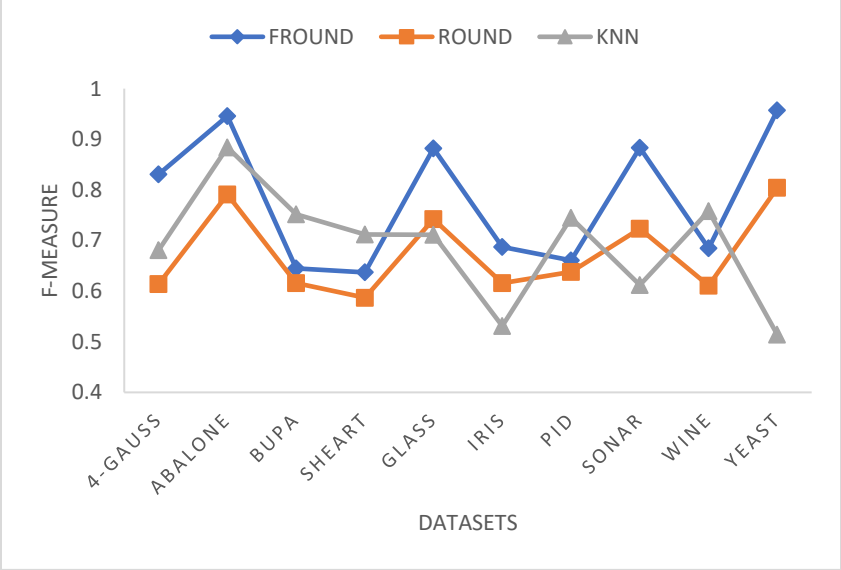


Figure 5.4 Comparison of the average of F-Measures in terms of different datasets

Because of the fuzzy relaxation in DDs it was expected that FROUND act better than ROUND and it can be confirmed by the results shown in Figure 5.3. In addition, for two high missing rate (%75, %10) and (%95, %20), the HR category, KNN have higher F-Measure. Since KNN has a fully complete imputation and the DDs are not reliable in the HR category, this results was predictable.

According to Figure 5.4, the F-Measure for FROUND is better than ROUND in almost all datasets, except for PID and WINE. These two datasets have very high variances. It shows that in the high variance cases the fuzzy relaxation in DDs could not help to achieve the better F-Measure.

5.4 The percentage of imputed cells and completed rows

Figures 5.5 to 5.8 clearly show that FROUND have acted better than ROUND in the higher percentage of imputation for both cells and rows. Of course, the comparison with KNN is not meaningful here. KNN fills all null cells by a function on some values in the dataset. So all null cells would be filled and all incomplete rows would be completed during KNN procedure. However, the results that illustrated in these four figures prove the superiority of our proposed method with respect to the Song's method.

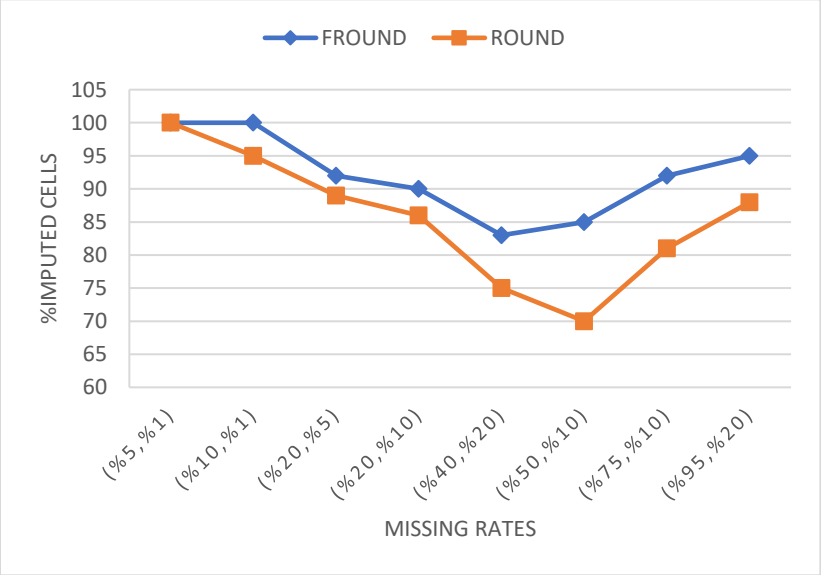


Figure 5.5 Comparison of the percentage of imputed cells in terms of different missing rates

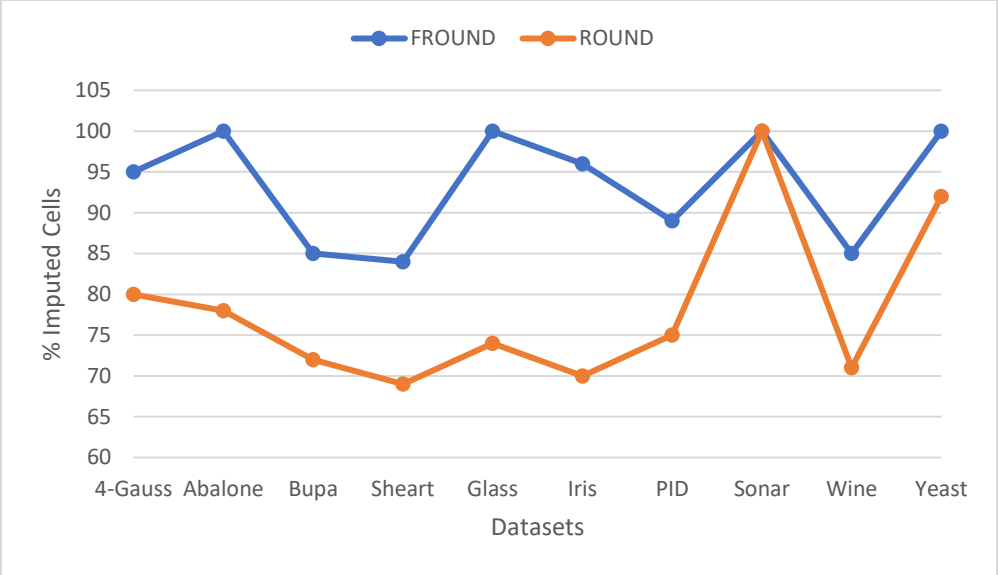


Figure 5.6 Comparison of the percentage of imputed cells in terms of different datasets

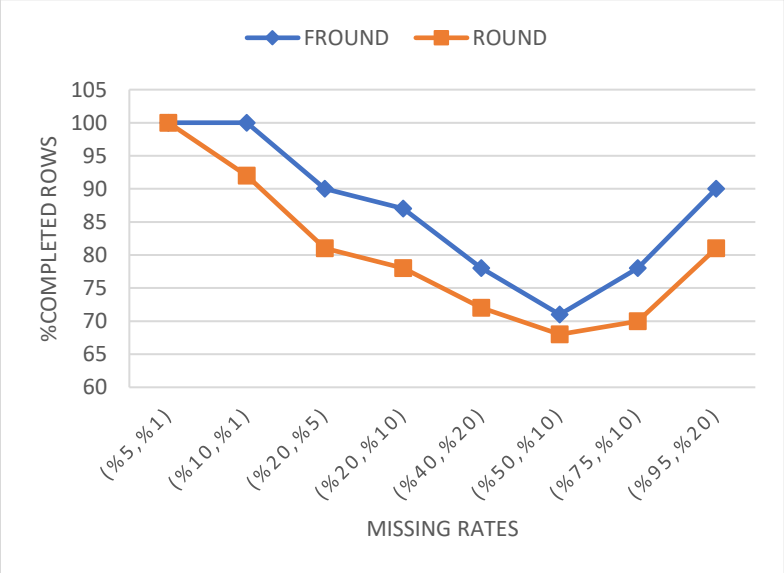


Figure 5.7 Comparison of the percentage of completed rows in terms of different missing rates

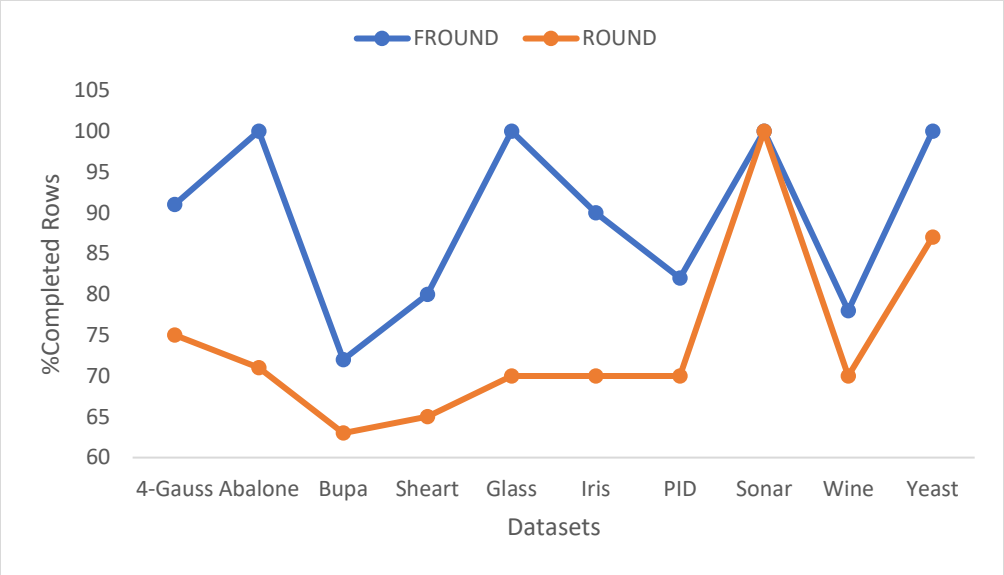


Figure 5.8 Comparison of the percentage of completed rows in terms of different datasets

5.5 The NRMS and the F-Measure in the categorized missing rates

The box plots in Figures 5.9 and 5.10, compare the three methods in terms of NRMS and F-Measure in different categories of missing rates.

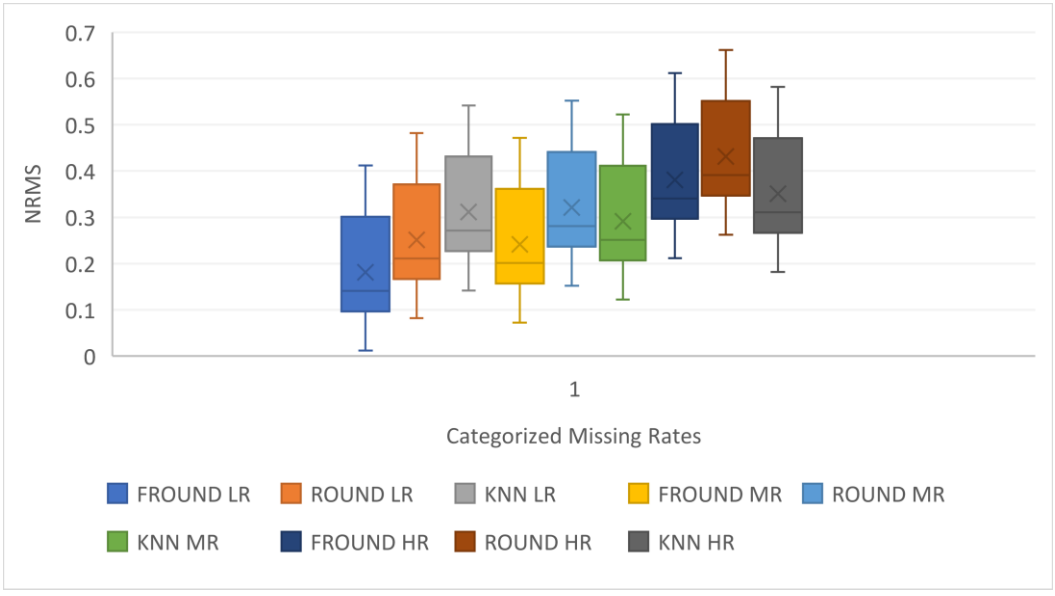


Figure 5.9 Comparison of the NRMS in terms of categorized missing rates

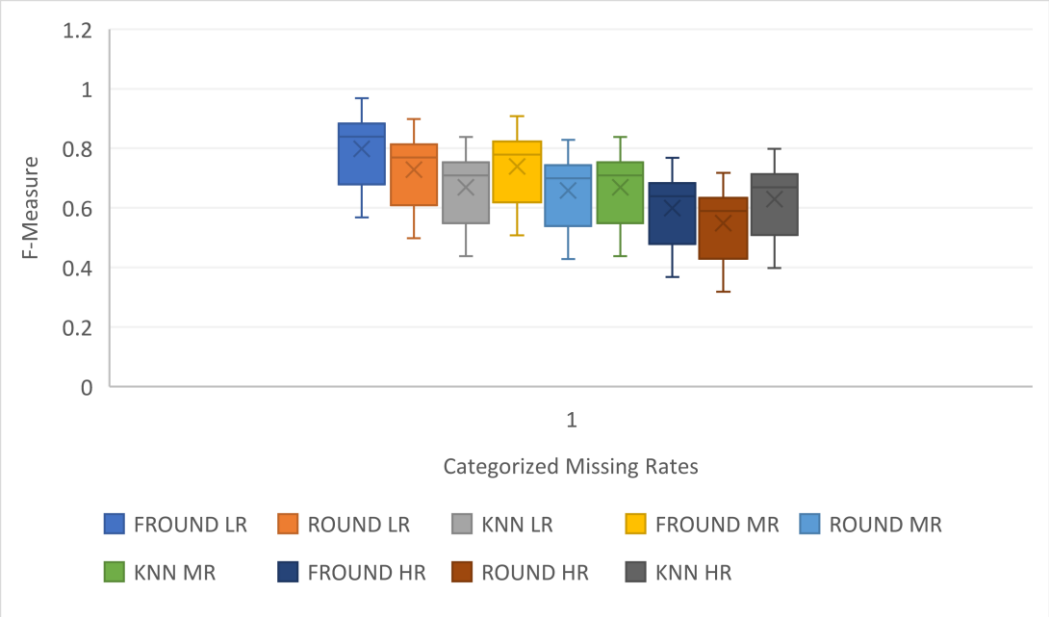


Figure 5.10 Comparison of the F-Measure in terms of categorized missing rates

5.6 The relation between the increase of imputation and the violation amounts

The main difference between FROUND and ROUND is that FROUND accepts more imputed rows with a little violation from DDs, while ROUND does not accept any violation even it leads to less imputation. The question that arises here is that "is there any relation between the percentage of completed rows and the degree of violation from DDs?"

Let us set $F = the\ percentage\ of\ completed\ rows\ in\ FROUND$ and $R = the\ percentage\ of\ completed\ rows\ in\ ROUND$. So the ratio of F to R, that is $\frac{F}{R}$, shows the ratio of completed rows by FROUND to those of the ROUND. Figures 5.11, 5.12 and 5.13, show the relationship between this ratio and the average of DDs' satisfactory degree in the different 10 types of datasets, different missing rates and in all 80 datasets, respectively.

Although experimental results show that flexibility in meeting the DDs in our method increases the percentage of completed rows, the scatter plots indicate no meaningful relation between the $\frac{F}{R}$ value and the average degree of satisfaction.

It was expected that the higher-percentage imputations would be associated with more violations, or those lower-rate violations would be associated with lower percentages of

substitutions. But the results of imputation with our proposed method in these 80 datasets do not support these claims.

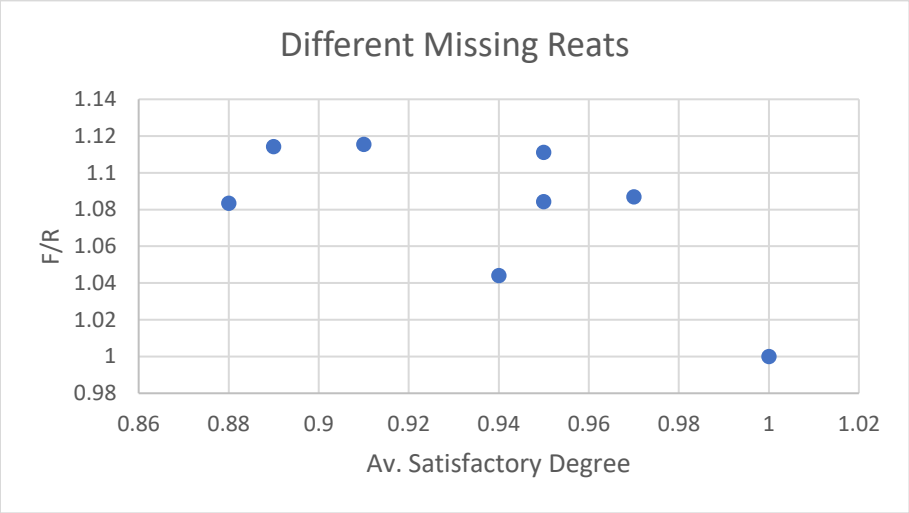


Figure 5.11 The relationship between F/R value and the average of DDs' satisfactory degree in terms of different missing rates. F/R value is the ratio of completed rows by FROUND to those of the ROUND.

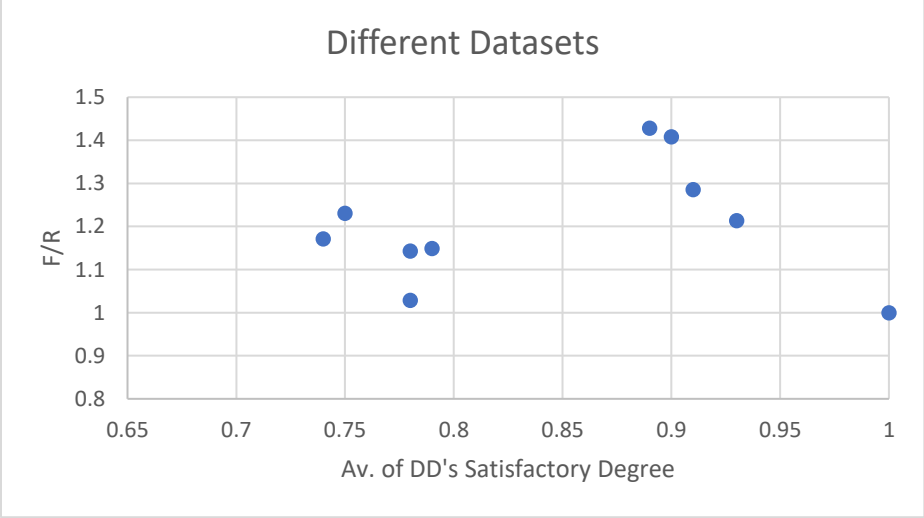


Figure 5.12 The relationship between F/R value and the average of DDs' satisfactory degree in terms of different types of datasets. F/R value is the ratio of completed rows by FROUND to those of the ROUND.

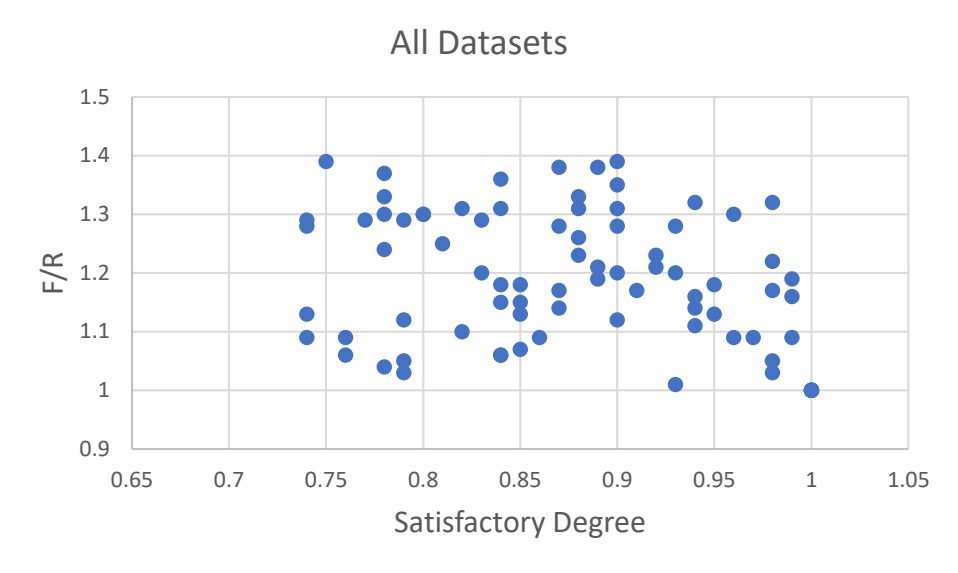


Figure 5.13 The relationship between F/R value and the average of DDs' satisfactory degree in terms of all 80 datasets. F/R value is the ratio of completed rows by FROUND to those of the ROUND.

.

5.7 Summary

In this chapter, we compared the results of our proposed method with the Song's method and KNN method. 10 numerical datasets are considered from the Kaggle site. These selected datasets have a diversity of data with a wide range of variances. From each of these datasets, some data are missed with the different percentages in 8 categories.

Data imputations were implemented by our proposed method as well as Song and KNN methods on all 80 datasets. The results compared in terms of the NRMS, the F-Measure the percentage of imputed cells and completed rows. Moreover, The NRMS and the F-Measure for the three methods were compared in categorized missing rates.

Our method is significantly superior to the Song's method in terms of the percentage of imputed data and the percentage of completed rows. This superiority is evident in both datasets and different missing rates.

In addition, in terms of F-Measure our method is much better than the Song's method. Although, in aspect of NRMS our method has better results in some datasets, but in those datasets that contain more integer numbers the Song's method is better than our method as well as in those datasets with the high missing rates.

The following tables compare the averages in different cases. The results indicate the superiority of our approach in all criteria NRMS, F-Measure, percentage of imputed cells and the percentage of completed cells at both cases different missing rates and different datasets.

Table 5.2 The average of NRMS in different missing rate and different datasets

	FROUND	ROUND	KNN
Av. NRMS (different missing rates)	0.1806	0.1942	0.2217
Av. NRMS (different datasets)	0.1663	0.1957	0.1988

Table 5.3 The average of F-Measure in different missing rate and different datasets

	FROUND	ROUND	KNN
Av. F-measure (different missing rates)	0.7179	0.6269	0.7085
Av. F-measure (different datasets)	0.7814	0.6742	0.69

Table 5.4 The average of imputed cells in different missing rate and different datasets

	FROUND	ROUND
Imputed cells (different datasets)	93.4	78.1
Imputed cells (different missing rates)	92.125	85.5

Table 5.5 The average of completed cells in different missing rate and different datasets

	FROUND	ROUND
Completed rows (different datasets)	89.3	74.1
Completed rows (different missing rates)	86.75	70.25

CHAPTER 6

CONCLUSION AND FUTURE WORKS

6.1 Summary and Conclusion

Missing data imputation is one of the most important challenges in data analysis. Its wide range of applications in various sciences and technologies has made many researchers work in this field. The publication of thousands of articles on missing data shows the high importance of this field.

The variety of related subjects and the diversity of datasets have made it impossible for any method to have the best performance in all aspects and for all datasets.

On the one hand, data clustering has led to the emergence of various methods, and on the other hand, attention to the relationship between features has developed a variety of other methods.

If the differential dependencies between the attributes are correctly identified and the missed data is replaced by taking into account these dependencies, we expect to have lower error estimates. However, there are some fundamental problems. First, determining accurate DDs requires a high level of experience in the data workspace, feature recognition, in-depth data analysis, and an expert team.

Another problem is that for the new dataset all the analysis have to be done again and the DDs of a dataset is not suitable for the another dataset.

When it comes to solving the optimization problem, the main problem is that by a little increase in the number of candidates, the number of constraints and problem variables increases sharply.

The Sang method ignores incompatible candidates even if the degree of incompatibility is very small, and even if it has to choose a candidate that fills fewer null cells. On the other hand, our method accepts a slight violation of the DDs provided that more null cells are filled. Anyway, the main purpose of this thesis was to improve the data imputation method based on DDs in order to increase the number of imputations. Numerical results confirm that our method has been successful in its goal.

Although in most of the dataset in Table 5.1 our method for NRMS and F-Measure also has a better result than the Song's method, this cannot be reliable. The more accurate and reliable the DDs are, the higher the probability of error and the amount of NRMS in our method.

If we reduce the admissible limit of violation in fuzzy membership functions due to the reliability of DDs or their high sensitivity, then the difference between the results of our method and the Sang method in the percentage of imputation will be less. However, due to the structure of our method, it will always have better or at least the same results in terms of the number of replacements than the Song's method.

6.2 Future Works

We can focus on the relationship between the rate of changes in ρ and δ in DDGEN Algorithm and its effect on NRMS and F-Measure as future works.

We have suggested DDs with one or two antecedents, but we can work on the effect of increasing it to 3 or more and compare the results.

As another future research, we can work on DI based on a combination of functional dependency and differential dependency. In this case we can use the TANE algorithm [27] and our FROUND algorithm to generate some rules.

REFERENCES

[1] P.E. McKnight, K.M. McKnight, S. Sidani, A.J. Figueredo (2007) Missing data: a gentle introduction, (Methodology in the Social Sciences), New York, Guilford Press, ISBN 9781593853938.

[2] R.J.A. Little, D.B. Rubin, (1987) Statistical analysis with missing data. Wiley, Hoboken.

[3] M. Raymond, D. Roberts, A comparison of methods for treating incomplete data in selection research. Educ. Psychol. Meas. 47:13–26, 1987.

[4] T. De Waal, J. Pannekoek, S. Scholtus, (2011), Handbook of Statistical Data Editing and Imputation. John Wiley & Sons.

[5] D.B. Rubin, (1987). Multiple imputation for nonresponse in surveys. New York, NY: Wiley.

[6] S. Song, Y. Sun, A. Zhang, L. Chen and J. Wang, Enriching Data Imputation under Similarity Rule Constraints, IEEE Transactions on Knowledge and Data Engineering, vol. 32, no. 2, 275-287, 2020.

[7] M.R. Safi, H.R. Maleki and E. Zaeimazad, A Note on the Zimmermann Method for Solving Fuzzy Linear Programming, Iranian Journal of Fuzzy Systems, Vol. 4, No. 2, 31-45. 2007.

[8] www.co2.earth/daily-co2 (Daily CO2)

[9] E. Fix, J.L. Hodges, (1951). Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties (Report). USAF School of Aviation Medicine, Randolph Field, Texas.

[10] N.S. Altman, An introduction to kernel and nearest-neighbor nonparametric regression, (PDF). The American Statistician. 46 (3), 175–185. 1992.

[11] H. Schwender, Imputing missing genotypes with weighted k nearest neighbors. Journal of Toxicology and Environmental Health, Part A 75 (8-10), 438–446, 2012.

[12] G. Tutz, S. Ramzan, Improved methods for the imputation of missing data by nearest neighbor methods. Computational Statistics & Data Analysis, 84–99, 2015

[13] M.N. Sha'abani, A.H. Fuad, N. Jamal, M.F. Ismail M, (2020) kNN and SVM Classification for EEG: A Review. In: Kasruddin Nasir A.N. et al. (eds) InECCE2019. Lecture Notes in Electrical Engineering, vol 632. Springer, Singapore.

[14] V.I. Levenshtein, Vladimir, Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady. 10 (8): 707–710, 1996.

[15] R.W. Hamming, Error detecting and error correcting codes, The Bell System Technical Journal. 29 (2): 147–160, 1950.

[16] R. E. Steuer, (1986), Multiple Criteria Optimization: Theory, Computation and Application, John Wiley & Sons, New York.

[17] L.A. Zadeh, Fuzzy sets, Information and Control 8 (3) 338–353, 1965.

[18] H. J. Zimmermann, Description and optimization of fuzzy systems, International Journal of General Systems, 2, 209- 215, 1976.

[19] M.R. Safi, H.R. Maleki, E. Zaeimazad, A Geometric Approach for Solving Fuzzy Linear Programming, Fuzzy Optimization and Decision Making, 6, 315-336, 2007.

[20] M.R., Safi, A. Razmjoo., Illustrating the Difficulties of Zimmermann Method for Solving the Fuzzy Linear Programming by the Geometric Approach", IJCCI 2012 - Proceedings of the 4th International Joint Conference on Computational Intelligence & Fuzzy Computation Theory and Application, 435-438, SciTePress, 5 - 7 October, 2012, Barcelona, Spain.

[21] F.E. Allan, J. Wishart. A method of estimating the yield of a missing plot in field experimental work. Jour. Agr. Sci., 20(3), 399-406, 1930.

[22] A. Farhangfar, L.A. Kurgan, W. Pedrycz, A novel framework for imputation of missing values in data- bases. IEEE Trans Syst Man Cybern A Syst Humans 37(5):692–709, 2007.

[23] K. Hron, M. Templ, P. Filzmoser, Imputation of missing values for compositional data using classi- cal and robust methods. Comput. Stat. Data Anal. 54:3095–3107, 2010.

[24] F. Arteaga, A.J. Ferrer-Riquelme, Missing Data, Comprehensive Chemometrics, Chemical and Biochemical Data Analysis, 285-314, 2009.

[25] D.B. Rubin, Assignment to Treatment Group on the Basis of a Covariate. Journal of Educational Statistics, 2(1), 1-26, 1977.

[26] W.C. Lin, C.F. Tsai, Missing value imputation: a review and analysis of the literature (2006–2017). Artificial Intelligence Review, 53, 1487–1509, 2020.

[27] Y. Huhtala, J. Kärkkäinen, P. Porkka, H. Toivonen, An efficient algorithm for discovering functional and approximate dependencies, The Computer Journal, 42, 2, 100-111, 1999.

[28] L. Caruccio, V. Deufemia, G. Polese, Relaxed Functional Dependencies—A Survey of Approaches, IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 1, 147-165, 2016.

[29] S. Song, L. Chen, Differential dependencies: Reasoning and discovery, ACM Trans. Database Syst., vol. 36, no. 3, Art. No. 16. 1-41, 2011.

[30] S. Song, L. Chen, H. Cheng, “Parameter-free determination of distance thresholds for metric distance constraints,” in Proc. Int. Conf. Data Eng. 846–857, 2012.

[31] S. Song, A. Zhang, L. Chen, J. Wang, “Enriching data imputation with extensive similarity neighbors,” Proc. VLDB Endowment, vol. 8, No. 11, 1286–1297, 2015.

[32] L.Y. Liang, N. Xiang, W. YuanXi, Z. Wei, Study on effect of temperature and humidity on the CO2 concentration measurement, IOP Conference Series: Earth and Environmental Science,1-5, 2017.

[33] www.canadianenergyissues.com/ontario-power-stats

[34] https://weather.gc.ca/airquality/aq_bulletins_e.html?Bulletin=fpcn48.cwao

[35] E. Lasseuguettea, M. Carta, S. Brandani, Maria-Chiara Ferrari, Effect of humidity and flue gas impurities on CO2 permeation of a polymer of intrinsic micro porosity for post-combustion capture, International Journal of Greenhouse Gas Control, 50, 93-99, 2016.

VITA AUCTORIS

NAME: Mohammadreza Safi

PLACE OF BIRTH: Tehran, Iran

YEAR OF BIRTH: 1967

EDUCATION: University of Windsor, MASc:
Electrical Engineering, Windsor, ON 2021.
University of Shahid Bahonar Kerman, PhD,
Applied Mathematics (Operations Research),
Kerman, Iran, 2006.
University of Shahid Bahonar Kerman, MSc,
Applied Mathematics (Operations Research),
Kerman, Iran, 1995.
University of Sistan and Balouchestan, BSc,
Applied Mathematics (Operations Research),
Kerman, Iran, 1991.