An Evaluation of *k*-Nearest Neighbour Imputation Using Likert Data

Per Jönsson and Claes Wohlin
School of Engineering, Blekinge Institute of Technology
PO-Box 520, SE-372 25, Ronneby, Sweden
per.jonsson@bth.se, claes.wohlin@bth.se

Abstract

Studies in many different fields of research suffer from the problem of missing data. With missing data, statistical tests will lose power, results may be biased, or analysis may not be feasible at all. There are several ways to handle the problem, for example through imputation. With imputation, missing values are replaced with estimated values according to an imputation method or model. In the k-Nearest Neighbour (k-NN) method, a case is imputed using values from the k most similar cases. In this paper, we present an evaluation of the k-NN method using Likert data in a software engineering context. We simulate the method with different values of k and for different percentages of missing data. Our findings indicate that it is feasible to use the k-NN method with Likert data. We suggest that a suitable value of k is approximately the square root of the number of complete cases. We also show that by relaxing the method rules with respect to selecting neighbours, the ability of the method remains high for large amounts of missing data without affecting the quality of the imputation.

1. Introduction

Missing data is both common and problematic in many fields of research [16], for example artificial intelligence [7], machine learning [1] and psychology [4]. The situation is, unsurprisingly, similar in software engineering [2, 11, 18]. The absence of data may substantially affect data analysis - statistical tests will lose power and results may be biased because of underlying differences between cases with and without missing data [9]. Simple ways to deal with missing data are, for example, *listwise deletion*, in which incomplete cases are discarded from the data set, or *variable deletion*, in which variables with missing data are discarded. However, the consequence is that potentially valuable data is discarded, which is even worse than having missing data in the first place. A better solution, that does not require useful data to be removed, is to use *imputation* methods. Imputation methods work by substituting the missing data with replacement values, hence increasing the amount of usable data.

A multitude of imputation methods exist (see, for example, [8] for a categorisation). Here, the focus is set on hot deck imputation. In hot deck imputation, a missing value is replaced with a value calculated from one or more complete cases (the *donors*) in the same data set. The choice of donors should depend on the case being imputed, which means that ordinary mean imputation, in which a missing value is replaced with the mean of the non-missing values, does not qualify as a hot deck method [15]. There are different ways of picking a replacement value, for example by choosing a value from one of the donors by random [9] or by calculating the mean of the corresponding values of the donors [1, 2].

The *k*-Nearest Neighbour (*k*-NN) method is a common hot deck method, in which *k* donors are selected from the *neighbours* (i.e., the complete cases) such that they minimise some similarity measure [15]. The method is briefly described in section 3.1. An advantage over mean imputation is that the replacement values are influenced only by the most similar cases rather than by all cases. Several studies have found that the *k*-NN method performs well or better than other methods, both in software engineering contexts [2, 17, 18] and in non-software engineering contexts [1, 3, 19].

1.1. Objective and Research Questions

In this paper, we evaluate the performance of the *k*-NN method with Likert data in a software engineering context. Likert data is ordinal, and is commonly used when collecting subjective opinions of individuals in surveys [14]. The evaluation is performed through a simulation, in which the method is applied to data sets with artificially removed data. The evaluation data comes from a study about software architecture documentation, described in [12].

Proceedings of the 10th International Symposium on Software Metrics (METRICS'04)
1530-1435/04 $ 20.00 IEEE

The importance of the evaluation is justified by the fact that we have not seen any studies on the use of the *k*-NN method with Likert data within empirical software engineering. Thus, the main objective and research question is whether it is feasible to use the method in said context. If so, additional research questions are of interest:

- How many donors should preferably be selected?
- At which amount of missing data is it no longer relevant to use the method?
- Is it possible to decrease the sensitiveness to the amount of missing data by allowing imputation from certain incomplete cases as well? (This relaxation of method rules is described in section 3.2.)

The remainder of the paper is structured as follows. In sections 2 and 3, we outline related work and give a presentation of the *k*-NN method. Then, we introduce the process we have used for evaluating the method in section 4, which is followed by a short description of the simulation of that process in section 5. Finally, we present the simulation results and draw conclusions in sections 6 and 7 respectively.

2. Related Work

As Cartwright et al. point out, publications about imputation in empirical software engineering are few [2]. To our knowledge, those that exist have focused on comparing the performance of different imputation methods. For example, Myrtveit et al. compare four methods for dealing with missing data: listwise deletion, mean imputation, full information maximum likelihood and similar response pattern imputation (which is related to *k*-NN with $k = 1$) [11]. They conclude, among other things, that similar response pattern imputation should only be used if the need for more data is urgent. Strike et al. describe a simulation of listwise deletion, mean imputation and hot deck imputation (in fact, *k*-NN with $k = 1$) [18], and conclude that hot deck imputation has the best performance in terms of bias and precision. Furthermore, they recommend the use of Euclidean distance as a similarity measure. In these two studies, the context is software cost estimation. Cartwright et al. themselves compare sample mean imputation and *k*-NN [2], and reach the conclusion that *k*-NN may be useful in software engineering research. In [17], Song and Shepperd evaluate the difference between MCAR and MAR using *k*-NN and class mean imputation. Their findings indicate that the type of missingness does not have a significant effect on either of the imputation methods, and furthermore that class mean imputation performs slightly better than *k*-NN. In these two studies, the context is software project effort prediction.

In other research areas, the comparison of imputation methods is common as well. Batista and Monard compare *k*-NN with the machine learning algorithms C4.5 and C2, and conclude that *k*-NN outperforms the other two, and that it is suitable also when the amount of missing data is large [1]. Engels and Diehr compare 14 imputation methods, among them one hot deck method (not *k*-NN, though), on longitudinal health care data [6]. They report, however, that the hot deck method did not perform as well as other methods. In [9], Huisman presents a comparison of imputation methods, including *k*-NN with $k = 1$. He concludes that the *k*-NN method performs well when the number of response options is large, but that corrected item mean imputation generally is the best imputation method. In the context of DNA research, Troyanskaya et al. report on a comparison of three imputation methods: one based on single value decomposition, one *k*-NN variant and row average [19]. They conclude that the *k*-NN method is far better than the other methods, and also that it is robust with respect to amount of missing data and type of data. Moreover, they recommend the use of Euclidean distance as a similarity measure.

Imputation in surveys is common, due to the fact that surveys often are faced with the problem of missing data. De Leeuw describes the problem of missing data in surveys and gives suggestions for how to deal with it [10]. In [4], Downey and King evaluate two methods for imputing Likert data, which is often used in surveys. Their results show that both methods, item mean and person mean substitution, perform well if the amount of missing data is less than 20%. Raaijmakers presents an imputation method, relative mean substitution, for imputing Likert data in large-scale surveys [13]. In comparing the method to others, he concludes that it seems to be beneficial in this setting. He also suggests that it is of greater importance to study the effect of imputation on different types of data and research strategies than to study the effectiveness of different statistics. Nevertheless, Chen and Shao evaluate *k*-NN imputation with $k = 1$ for survey data, and show that the method has good performance with respect to bias and variance of the mean of estimated values [3].

Gediga and Düntsch present in [7] an imputation method based on non-numeric rule data analysis. Their method does not make assumptions about the distribution of data, and works with consistency between cases rather than distance. Two cases are said to be consistent when their non-missing values are the same whenever they occur in both cases, i.e. donorship is allowed both for complete and incomplete cases. This resembles our relaxation of the *k*-NN method rules when it comes to selecting neighbours (see section 3.2), in that both approaches allow values that will not contribute to the similarity measure to be missing in the donor cases.

3. *k*-Nearest Neighbour

In this section, we describe how the *k*-NN method works and how its properties affect the imputation. We also discuss two different strategies for selecting neighbours. While one adheres to the method rules in that only complete cases can be neighbours, the other relaxes this restriction slightly.

3.1. Method

In the *k*-NN method, missing values in a case are imputed using values calculated from the *k* nearest neighbours, hence the name. The nearest, most similar, neighbours are found by minimising a distance function, usually the *Euclidean distance*, defined as (see, for example, [20]):

$$E(a, b) = \sqrt{\sum_{i \in D} (x_{ai} - x_{bi})^2}$$

where

- $E(a, b)$ is the distance between the two cases a and b,
- x_{ai} and x_{bi} are the values of attribute i in cases a and b, respectively, and
- D is the set of attributes with non-missing values in both cases.

The use of Euclidean distance as similarity measure is recommended by Strike et al. [18] and Troyanskaya et al. [19]. The *k*-NN method does not suffer from the problem with reduced variance to the same extent as mean imputation, because when mean imputation imputes the same value (the mean) for all cases, *k*-NN imputes different values depending on the case being imputed.

Consider the data set shown in table 1; when calculating the distance between cases Bridget and Eric, the attributes for which both have values are Q1, Q3, Q4 and Q5. Thus, $D = \{Q1, Q3, Q4, Q5\}$. We see that Bridget's answer to Q2 does not contribute to the calculation of the distance, because it is not in D. This implies that whether a neighbour has values for attributes outside D or not does not affect its similarity to the case being imputed. For example, Bridget and Eric are equally similar to Susan, because

$$E(\text{Bridget, Su.}) = E(\text{Eric, Su.}) = \sqrt{2 \cdot (4-2)^2} \approx 2{,}8$$

despite the fact that Bridget is more complete than Eric.

Another consequence of how the Euclidean distance is calculated, is that it is easier to find near neighbours when D is small. This occurs because the number of terms under the radical sign has fairly large impact on the distance. Again, consider the data set in table 1; based on the Euclidean distance, Bridget and Eric are equally similar to Quentin (in fact, their distances are zero). Still, they differ considerably on Q5, and Eric has not answered Q2 at all. This suggests that the distance function does not necessarily reflect the *true* similarity between cases when D is small.

Table 1. Example Incomplete Data Set

	Q1	Q2	Q3	Q4	Q5
Bridget	2	3	4	2	1
Eric	2	-	2	4	5
Susan	-	-	2	4	-
Quentin	2	-	-	-	-

Once the k nearest neighbours (donors) have been found, a replacement value to substitute for the missing attribute value must be estimated. How the replacement value is calculated depends on the type of data; the mode can be used for discrete data and the mean for continuous data [1]. Because the mode may be tied (several values may have the same frequency), and because we use Likert data where the magnitude of a value matters, we will instead use the median for estimating a replacement value.

An important parameter for the *k*-NN method is the value of k. Duda and Hart suggest, albeit in the context of probability density estimation within pattern classification, the use of $k \approx \sqrt{N}$, where N in our case corresponds to the number of neighbours [5]. Cartwright et al., on the other hand, suggest a low k, typically 1 or 2, but point out that $k = 1$ is sensitive to outliers and consequently use $k = 2$ [2]. Several others use $k = 1$, for example Myrtveit et al. [11], Strike et al. [18], Huisman [9] and Chen and Shao [3]. Batista and Monard, on the other hand, report on $k = 10$ for large data sets [1], while Troyanskaya et al. argue that the method is fairly insensitive to the choice of k. As k increases, the mean distance to the donors gets larger, which implies that the replacement values could be less precise. Eventually, as k approaches N, the method converges to ordinary mean imputation (median, in our case) where also the most distant cases contribute.

3.2. Neighbour Strategy

In hot deck imputation, and consequently in *k*-NN imputation, only complete cases can be used for imputing missing values [1, 2, 15]. In other words, only complete cases can be neighbours. Based on the discussion in the previous section about how the Euclidean distance between cases is unaffected by values of attributes not in D, we suggest that it is possible to relax this restriction slightly. Thus, we see two distinct strategies for selecting neighbours.

The first strategy is in line with how the method normally is used, and allows only the complete cases to be neighbours. This means that no incomplete cases can contribute to the substitution of a replacement value in an incomplete case. We will refer to this strategy as the *CC strategy*, where CC means "complete case".

The second strategy allows all complete cases and certain incomplete cases to be neighbours. More specifically, a case can act as a neighbour if and only if it contains values for all attributes that the case being imputed has values for, *and* for the attribute being imputed. We will refer to this strategy as the *IC strategy*, where IC means "incomplete case".

It is important to note that we do not permit already imputed cases to be donors in any of the strategies. Thus, imputed data will never be used to impute new data.

For an example of the two strategies, consult again table 1. Assuming we are about to impute attribute Q1 for Susan, the CC strategy would only allow Bridget to be a neighbour. The IC strategy, however, would allow both Bridget and Eric to be neighbours, because Eric contains values for at least the necessary attributes: Q1, Q3 and Q4. Because the IC strategy potentially has more neighbours to select donors from, it can be expected to be able to "survive" larger amounts of missing data than the CC strategy.

4. Evaluation Process

The process for evaluating the *k*-NN method consists of three main steps: data removal, imputation and evaluation, as is shown in the process chart in figure 1 below. In this section, we describe how the three steps work and what they produce. The simulation of this process is shortly described in section 5.

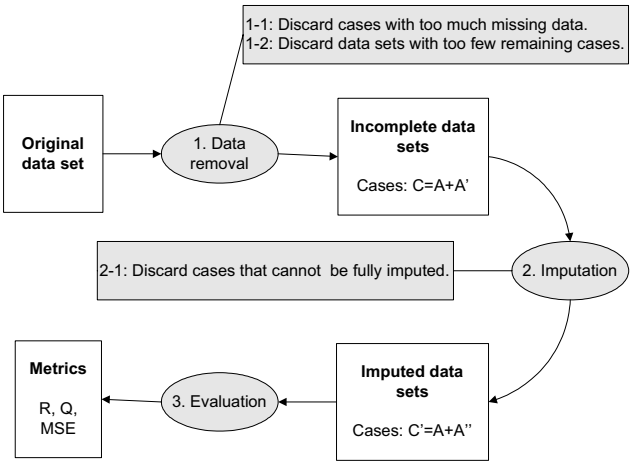

Figure 1. Process Outline

4.1. Data Removal (step 1)

The first step, numbered 1 in the process chart, requires a complete data set to work with. This original data set contains only cases without missing data. Data is removed from the original data set in order to produce artificially incomplete data sets for the imputation step. There are three main ways in which data can be missing from a data set [1, 2, 16]:

- *MCAR* (missing completely at random), means that the missing data is independent on any variable observed in the data set.
- *MAR* (missing at random), means that the missing data may depend on variables observed in the data set, but not on the missing values themselves.
- *NMAR* (not missing at random, or *NI*, non-ignorable), means that the missing data depends on the missing values themselves, and not on any other observed variable.

Any actions for dealing with missing data are dependent on why the data is missing. For example, to discard cases with missing data is dangerous unless the missing data is MCAR [16]. Otherwise, there is a risk that the remaining data is severely biased. Missing data that is NMAR is hardest to deal with, because it, obviously, is difficult to construct an imputation model based on unobserved data.

Data missing from the responses to a questionnaire is unlikely to be MCAR [13]. For example, a respondent could leave out an answer because of lack of interest, time, knowledge or because he or she did not consider a question relevant. If it is possible to distinguish between these different sources of missing data, an answer left out because of lack of question relevance could be regarded as useful information rather than a missing data point. If so, the degree of missingness would be different than if the source of missing data could not be distinguished. In any case, the missing data in a questionnaire is more likely MAR than MCAR. In order to remove data so that it is MAR, a model for the non-responsiveness is required. When analysing the results from a longitudinal study of health data, Engels and Diehr created such a model based on probabilities of missing data values [6]. In the absence of a good model for our data, however, we remove data in a completely random fashion, which means that the missing data is MCAR. We do not try to simulate different sources of missing data, so we consider all removed data points as being truly missing.

There are two parameters that guide the data removal step, the *case reduction limit* and the *data set reduction limit*. We call these reduction limits because they prevent the data from being reduced to a level where it is unusable. The effects of the parameters can be seen in the process chart. If it is decided in step 1-1 that a case contains too many missing values after data removal, as dictated by the case reduction limit, it is discarded from the data set. The

reason for having this limit is to avoid single cases with so little data that it becomes meaningless to calculate the Euclidean distance to other cases. If it is decided in step 1-2 that too few cases remain in the data set, as dictated by the data set reduction limit, the entire data set is discarded. The idea with this limit is to avoid a data set with so few cases that it no longer can be said to represent the original data set.

We acknowledge that by having these limits, we combine the *k*-NN imputation method with simple listwise deletion. As discussed earlier, this is dangerous unless the missing data *truly* is MCAR. However, we argue that keeping cases with very little data left would also be dangerous, because the imputed data would contain loosely grounded estimates. In other words, it is a trade-off that has to be made.

The removal step is executed for a number of different percentages. Furthermore, it is repeated several times for each percentage. Thus, the output from the removal step is a large number of incomplete data sets to be fed to the imputation step. For each incomplete data set coming from the removal step, we define:

- A as the number of complete cases remaining,
- A' as the number of incomplete cases remaining, and thus
- $C = A + A'$ as the total number of cases remaining.

Since entire cases may be discarded in the removal step, the actual percentage of missing data may be different from the intended percentage. For the incomplete data sets generated in the simulation, both the intended percentage and the actual percentage of missing data are presented. When analysing the results, it is the actual percentage that is used, though.

4.2. Imputation (step 2)

In the imputation step, numbered 2 in the process chart, the *k*-NN method is applied to each incomplete data set generated in the data removal step. For each incomplete data set, several imputations using different *k*-values and different neighbour strategies are performed. As mentioned earlier, we use the median value of the *k* nearest neighbours as replacement for a missing value, and because the data in the data set is of Likert type, it is not possible to insert non-integer values. Thus, only odd *k*-values are used, which results in that the median always becomes an integer value.

The *k* cases with least distances are chosen as donors, regardless of ties among the distances, i.e. two cases with equal distances are treated as two unique neighbours. This means that it is not always possible to pick *k* cases such that the remaining *K* - *k* cases (where *K* is the total number of neighbours) have distances greater to that of the *k*th case. Should such a situation occur, it is treated as follows. If l, $0 \le l < k$ cases have been picked, and there are m, $(k-l) < m \le (K-l)$ cases with distance d, then the $k - l$ first cases of the m, in the order they appear in the original data set, are picked.

If there are not enough neighbours available, cases may get lost in the imputation process. For the CC strategy, this will always happen when *k* is greater than the number of complete cases in the incomplete data set. The IC strategy has greater imputation ability, though, but will inevitably lose cases when *k* is large enough. This second situation where cases can be discarded is numbered 2-1 in the process chart.

The output from the imputation step is a number of imputed data sets, several for each incomplete data set generated in the data removal step. For each imputed data set, we define

- A'', $0 \le A'' \le A'$ as the number of cases that were imputed, i.e. that were not lost in step 2-1, and consequently
- $C' = A + A''$ as the total number of cases, and also
- B as the number of imputed attribute values.

4.3. Evaluation (step 3)

In the evaluation step, each imputed data set from the imputation step is compared to the original data set in order to measure the performance of the imputation. Three separate metrics are used: one ability metric and two quality metrics. The two quality metrics differ both in what they measure and how they measure it. The first quality metric is a measure of how many of the imputed attribute values that were imputed correctly. In other words, it is a precision metric. The second quality metric is a measure of how much those that were not imputed correctly differ from their correct values, which makes it a distance metric.

We define the *ability* metric as

$$R = \frac{A''}{A'}$$

which equals 0 if all incomplete cases were lost during the imputation (in step 2-1), and 1 if all incomplete cases were imputed.

To define the *precision* metric, let *B'* be the number of matching imputed attribute values. Then, the metric can be expressed as

$$Q = \begin{cases} \dfrac{B'}{B} & \text{if } B > 0 \\ \text{undefined} & \text{if } B = 0 \end{cases}$$

which equals 0 if all the imputed attribute values are incorrect, and 1 if all are correct.

Finally, we calculate the *mean square error* of the incorrectly imputed attribute values as

$$MSE = \begin{cases} \dfrac{\sum_i (x_i - \hat{x}_i)^2}{B - B'} & \text{if} \quad B > 0, B' < B \\ \text{undefined} & \text{if} \quad B = 0 \text{ or } B' = B \end{cases}$$

where x_i is the correct value and $\hat{x}_i$ is the imputed value of the *i*th incorrectly imputed attribute value.

Since $B = 0$ when $R = 0$, it is apparent that both the precision measure and the mean square error are invalid when the ability measure is zero. Moreover, the mean square error becomes invalid when $Q = 1$. Consequently, the three metrics need to have different priorities: R is the primary performance metric, Q is the secondary, and *MSE* is the tertiary. Recognising that it would be difficult to create one single metric for measuring the performance, no attempts to accomplish this have been made.

Average values of R, Q and *MSE* are presented in the results, because several imputations are performed with identical parameters (percentage, value of k and neighbour strategy). For R, the mean includes all measured instances, while for Q and *MSE*, only those instances where the metrics are not undefined are included.

5. Simulation

In this section, we describe the design of the simulation, which includes the original data used as input to the removal step, the parameters used when "instantiating" the process and some details about the simulation software used.

5.1. Original Data Set

The data used in the simulation comes from a case study on architecture documentation in a large Swedish organisation. The case study is described in detail in [12]. In the case study, a questionnaire containing questions about knowledge of architecture documentation was distributed to employees in the organisation. The data set on which we base the simulation consists of the answers to six of the questions in the questionnaire. 54 respondents gave answers to all of the six questions, which means that the data set used as input to the data removal step contains 54 cases.

Each of the six questions used a Likert scale for collecting answers, where the numbers 1 to 5 were used to represent different levels of agreement to some statement or query. Each of the numbers 1 to 5 was associated with a statement explaining its meaning, and we tried to make sure that the distance between two adjacent numbers was similar everywhere.

5.2. Parameters

Each of the three steps in the process described in section 4 is guided by a number of parameters. This section describes the values used for those parameters in the simulation.

As discussed, two reduction limits, the case reduction limit and the data set reduction limit, constrain the data removal step. In the simulation, we used the following values:

- Case reduction limit = 3 (inclusively)
- Data set reduction limit = 27 (inclusively)

With six attributes in each case, the case reduction limit means that cases with less than 50% of the attribute values left were discarded in step 2-1. The reason for this limit is that we wanted each imputed case to have at least equally much real data as imputed data.

With 54 cases in the original data set, the data set reduction limit means that data sets with less than 50% of the cases left were discarded in step 2-2. Since each case is a respondent, we wanted to make sure that each data set being imputed contained at least half of the respondents in the original data set.

The removal step generated data sets where 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55 and 60 percent data had been removed (however, as discussed in section 4.1, the actual percentages became different). For each percentage, 1 000 data sets were generated, which means that a total of 12 000 data sets were generated. The simulation was controlled so that the removal step would generate the requested number of data sets even if some data sets were discarded because of the data set reduction limit.

In the imputation step, the only controlling parameter is the choice of which k-values to use when imputing data sets. We decided to use odd values in an interval from 1 to C, inclusively. Even though we knew that the CC strategy would fail at $k = A + 1$, we expected the IC strategy to be able to handle larger k-values.

5.3. Software

In order to execute the simulation, an application for carrying out the data removal, imputation and evaluation steps was written. In addition, Microsoft Excel was used for analysing some of the results from the evaluation step.

In order to validate that the application worked correctly, a special data set was designed. The data set contained a low number of cases, in order to make it feasible to impute data manually, and was crafted so that the imputa-

tion should give different results both for different *k*-values, and for the two neighbour strategies.

By comparing the outcome of the imputations performed by the application to the outcome of imputations made manually, it was decided that the application was correct. To further assess this fact, a number of application features were inspected in more detail: the calculation of Euclidean distance, the calculation of median, and the selection of *k* donors for both strategies. Finally, a number of entries in the simulation results summary were randomly picked and checked for feasibility and correctness.

6. Results

In this section, we present the results of the simulation in two ways. First, we compare the ability and quality of the *k*-NN method for different *k*-values. In order to better understand how *k* is affected by the amount of missing data, we perform two additional simulations with increased numbers of attributes. Then, we compare the ability of the method for different amounts of missing data. We begin, however, with showing some descriptive statistics for the incomplete data sets generated in the removal step.

6.1. Incomplete Data Sets

As discussed in section 4.1, there is a difference between the amount of data removed from the original data set and the amount of data actually missing from the resulting, incomplete, data sets. The main reason for this is that entire cases may be discarded because of the case reduction limit. Another, less significant, reason is rounding effects. For example, removing 5% of the data in the original data set means removing 16 attribute values out of 324, which equals 4.9%.

Table 2 shows descriptive statistics for the incomplete data sets generated in the removal step. Each row represents the 1 000 data sets generated for the percentage stated in the left-most column. The second and third columns contain the mean and standard deviation (expressed with the same magnitude as the mean) of the percentage of missing data, respectively. The fourth and fifth columns contain the average number of cases and the average number of complete cases in each data set, respectively. Finally, the sixth column contains the average number of imputations made on each data set. This corresponds roughly to the average number of cases ($\overline{C}$), which is our upper limit of *k*.

Table 2. Overview of Incomplete Data Sets

Pct.	Mean missing data (%)	*s*	$\overline{C}$	$\overline{A}$	Avg. #imp.
5	4.9	0.1	54.0	39.8	54.0
10	9.8	0.3	53.9	28.8	54.0
15	14.5	0.5	53.7	20.4	53.9
20	19.0	0.8	53.2	14.2	53.6
25	23.4	1.0	52.1	9.6	52.6
30	27.2	1.2	50.5	6.3	51.0
35	30.8	1.3	48.4	4.0	48.9
40	34.4	1.3	46.0	2.4	46.5
45	38.0	1.3	43.1	1.5	43.6
50	42.1	1.3	40.1	0.8	40.6
55	46.5	1.3	37.4	0.4	37.9
60	51.5	1.3	34.9	0.2	35.4

6.2. Comparison of *k*-Values

For each percentage of missing data, we plotted the ability metric and the quality metrics for different values of *k* and for both neighbour selection strategies. Because of space constraints, we cannot show all 24 diagrams. This is not necessary, however, because there is a common pattern for all percentages. To illustrate this pattern, we show the diagrams for the data sets with 14.5% and 19.0% missing data, respectively, in figure 2.

The diagrams in the figure show the ability and quality for both neighbour strategies. In the upper diagram, the ability (*R*) is 1.0 up until *k* is around 15 for both strategies, after which it falls and reaches 0.5 when *k* is around 21 for the CC strategy and slightly more for the IC strategy. The latter limit coincides with the average number of complete cases ($\overline{A}$) in the data sets for this percentage. Similarly, in the lower diagram we see that the ability is 1.0 up until *k* is around 9, and falls to 0.5 when *k* is around 15. Such limits, albeit different, exist for other percentages as well.

Both diagrams further show that the precision (*Q*) of the method starts at around 0.4 when *k* is 1, and increases up to around 0.5 when *k* reaches 5. Thereafter, the precision is fairly unaffected by the value of *k* and varies only slightly on a "ledge" of *k*-values, an observation similar to that made by Troyanskaya et al. [19]. This is true for both strategies. Because of the priorities of the performance metrics, discussed in section 4.3, the ledge has a natural upper limit as the ability of the method drops. The initial increase in precision and the ledge of *k*-values exist for other percentages as well, up to a percentage where the drop in ability occurs already for a low *k*. In our data, this happens when

around 30% data is missing, in which case the ability drops to 0.8 for the CC strategy and 0.9 for the IC strategy already when *k* is 3.

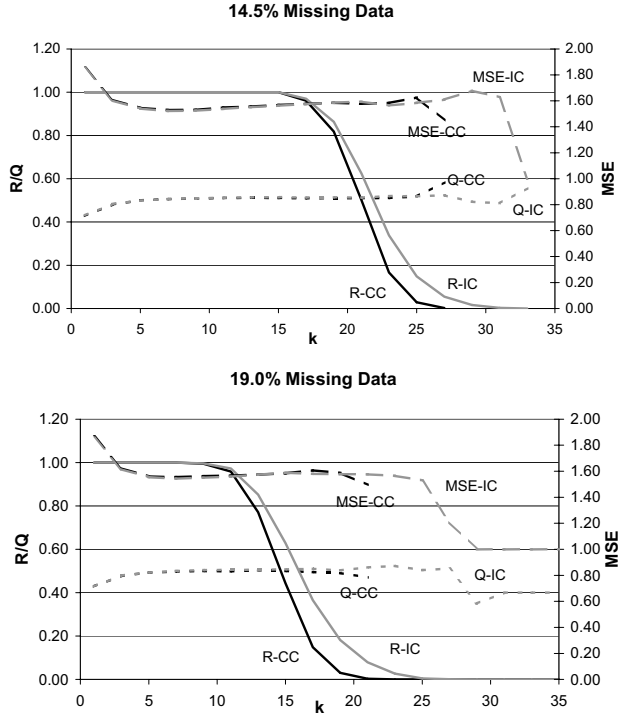

Figure 2. Performance at 14.5% and 19.0% Missing Data, CC and IC

The mean square error (*MSE*), which is the tertiary performance metric, starts off high but shows a noticeable decrease as *k* increases to 7. Then, it slowly increases for higher *k*-values on the aforementioned ledge. Although the increase is minimal, it seems to concur with the observation made in section 3.1, that the estimated replacement values get worse as the mean distance to the donors increase. The described pattern in mean square error occurs for both strategies and for other percentages as well.

The differences between the neighbour strategies can be seen by comparing the black curves, representing the CC strategy, to the grey curves, representing the IC strategy. As can be seen, the curves for *R*, *Q* and *MSE* are nearly identical between the strategies. The main difference is that the ability of the method, as expected, does not drop as fast for the IC strategy as it does for the CC strategy. Two important observations regarding the IC strategy are that the precision is generally not lower than for the CC strategy, and the mean square error is not larger.

We see, based on the discussion about the performance metrics above, that *k* should be selected so that it is large enough to be on the ledge, but low enough to minimise the mean square error. Since the ledge gradually diminishes for higher percentages of missing data, *k* would preferably depend on the amount of missing data. In fact, the dependency should be on the number of available neighbours for at least two reasons. First, the drop in ability occurs because the number of available neighbours decreases. For the CC strategy, the number of available neighbours is the number of complete cases. For the IC strategy, it is slightly more, but not so much more that the number of complete cases is an unfit approximation. Second, removing a certain percentage of data from two data sets with different numbers of attributes but the same number of cases would result in different numbers of complete cases.

Table 3 shows the observed optimal *k*-values for both neighbour selection strategies given the average number of complete cases for the simulated percentages. In the table, the rightmost column represents the data sets with four complete cases or less. It can be seen that the optimal value of *k* for a certain number of neighbours is the same for both strategies.

Table 3. Optimal *k*-Values for CC and IC

$\bar{A}$ =	**39.8**	**28.8**	**20.4**	**14.2**	**9.6**	**6.3**	**4.0-**
CC	7	7	7	7	5	3	1
IC	7	7	7	7	5	3	1

Looking for an appropriate model for *k*, we compared each optimal *k*-value to the square root of the average number of complete cases, as suggested by Duda and Hart. The reason they suggest this model is that *k* should be large enough to give a reliable result, but small enough to keep the donors as close as possible [5]. This concurs with our own requirements on *k*. Thus, we have chosen to examine $k = \text{round\_odd}(\sqrt{\bar{A}})$, i.e. the square root of the average number of complete cases after data removal, rounded to the nearest odd integer. This function is compared to the optimal *k*-values in table 4. As can be seen, the function underestimates *k* somewhat in the mid-range of missing data. This does not mean that the calculated *k*-values are inappropriate, though. The mean relative errors in *R*, *Q* and *MSE* between the calculated and the optimal *k*-values are for the CC strategy 0.07%, 0.67% and 0.73%, respectively, and for the IC strategy 0.04%, 0.78% and 0.80%, respectively.

As mentioned, the number of complete cases for a data set with a certain percentage missing data depends on, among other things, the number of attributes in the data set. Thus, in order to further test our findings, we performed two additional simulations. In the first, the number of attributes was increased to 12 by simply appending a copy

of each case to itself. In the second simulation, the number of attributes was increased to 18 in a similar way. The case reduction limits were increased accordingly. The diagrams in figure 3 show the results of imputing data sets with on average 9.9% missing data using the IC strategy. With 12 attributes, the average number of complete cases at this percentage is 15.3, and with 18 attributes it is 8.0. The precision (Q) is highest at $k = 3$ in both diagrams, but declines as k increases, instead of showing a ledge as was the case with six attributes. Another difference is that the precision generally is higher with more attributes. Also, the mean square error starts low in both diagrams, and the increase as k grows larger is articulated compared to the results with six attributes. These observations further support our requirements on k, as stated earlier. In total, the results from the two additional simulations indicate that it is suitable to use $k = \text{round\_odd}(\sqrt{\bar{A}})$ with higher numbers of attributes as well.

Table 4. Optimal *k* vs. Calculated *k*

$\bar{A}$ =	39.8	28.8	20.4	14.2	9.6	6.3	4.0-
Opt.	7	7	7	7	5	3	1
Calc.	7	5	5	3	3	3	1

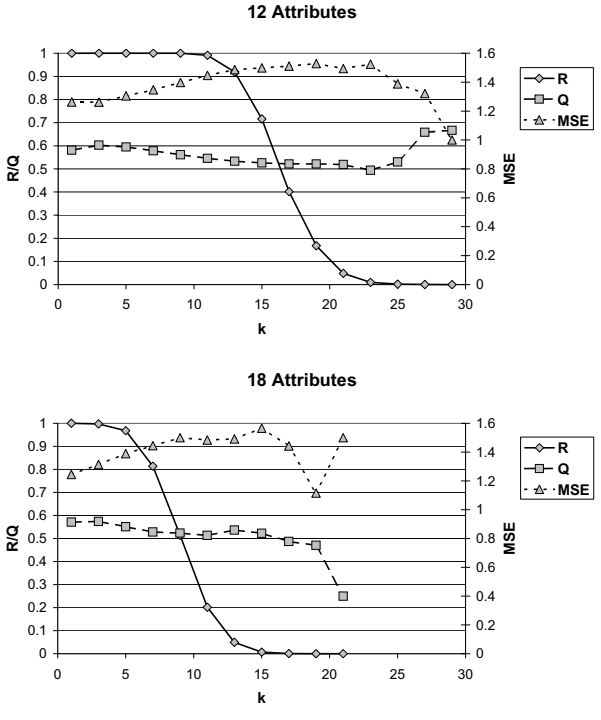

Figure 3. 9.9% Missing Data, 12/18 Attributes, IC

6.3. Comparison of Percentages

In addition to comparing the ability and quality for different *k*-values, we compared the ability of the method for different amounts of missing data, using for each percentage the optimal *k*-value found earlier. The diagram (for six attributes) can be seen in figure 4. Both neighbour strategies provide nearly maximum ability (R) up to around 30% missing data (when, on average, 88% of the cases are incomplete). After that, the ability when using the CC strategy drops rapidly down to 0.2 at around 50% missing data (when, on average, 98% of the cases are incomplete), meaning that only 20% of the incomplete cases were recovered. The IC strategy, on the other hand, drops less drastically and can recover nearly 60% of the incomplete cases at around 50% missing data.

The figure clearly shows that the IC strategy is more advantageous when more data is missing. Because the comparison of *k*-values showed that the IC strategy does not give lower precision or larger mean square error than the CC strategy, we consider it more favourable regardless of the amount of missing data.

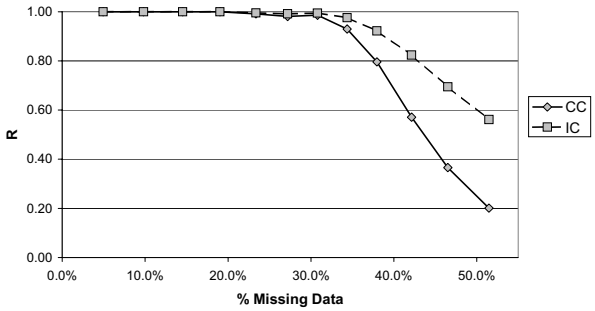

Figure 4. Ability vs. Amount of Missing Data

6.4. Interpretation of the Results

Our results indicate that the *k*-NN method performs well on the type of data we have used, provided that a suitable value of *k* is selected. We base our judgement on the following indicators (see figures 2, 3 and 4):

- The ability of the method, in particular when using the IC strategy, is high even when the amount of missing data (and thus the proportion of incomplete cases) is high.
- The precision is on average between 0.5 and 0.6, which means that at least half of the missing data points are imputed correctly.
- The mean square error is at worst 1.6 (for six attributes, and even lower for more attributes), which means that the incorrectly imputed data points are at most off by slightly more than one.

In the absence of something to compare with, it is obviously hard to assess the goodness of the values obtained on the quality metrics. However, we consider correct imputation for at least half of the data points and a deviation of slightly more than one for the other half of the data points to be good from a practical point of view. Put differently, we would not regard the imputation of data a serious threat to validity in a real-world study of this type.

It is of course desirable to achieve good values on all three performance metrics. However, when the performance decreases for whichever of the metrics, it is the priorities between them that should determine whether the imputation was successful or not. For example, if the quality drops but the ability stays high, the imputation may still be considered successful, because resorting to listwise deletion (or any other type of deletion procedure) may not be an option.

6.5. Threats to Validity

In the method, we used Euclidean distance as the similarity measure. However, the data was of Likert type, which means that it was on an ordinal scale. This makes it debatable to perform distance calculations, which normally requires an interval scale. Still, we argue that the distance calculations were relevant, and thus the validity threat minimal, because effort was put into making the distances between Likert numbers similar. Furthermore, our results show that the imputations were successful after all.

In step 1 of the evaluation, we removed data from the original data set completely randomly, which means that the missing data was MCAR. It is more likely, though, that missing responses to a questionnaire are MAR, as pointed out by Raaijmakers [13]. In other words, the missingness mechanism used in the evaluation did not fully represent a real-world situation.

It may be dangerous to use incomplete cases as donors when the missing data is MAR, for example if incomplete cases can be said to contain less valuable data. This could be the case if missing answers were an indication that the respondents did not take the questionnaire seriously. As a precaution, we recommend using a limit to prevent cases with far too much missing data both from being imputed and from acting as donors.

A threat to the generalisability of the results is that we used a fairly small data set with 54 cases as a basis for the simulation. With a small data set with missing data, the neighbours that can be used as donors are few, and thus the outcome of the imputation is sensitive to disturbances, such as outliers, in the data. We do, however, believe that it is not uncommon to get a small data set when collecting data from a survey, which means that our simulation should be relevant from this point of view.

7. Conclusions

In this paper, we have presented an evaluation of the performance of the *k*-Nearest Neighbour imputation method when using Likert data. This type of ordinal data is common in surveys that collect subjective opinions from individuals. The evaluation process was simulated using custom simulation software.

In the evaluation, we removed data randomly from a complete data set, containing real data collected in a previous study. Since we simulated the evaluation process, we were able to perform a number of imputations using different imputation parameters on a large number of incomplete data sets with different amounts of missing data. In the imputation process, we used different values of *k*, and also two different strategies for selecting neighbours, the CC strategy and the IC strategy. The CC strategy, which concurs with the rules of the *k*-NN method, allows only complete cases to act as neighbours. The IC strategy allows as neighbours also incomplete cases where attribute values that would not contribute to the distance calculation are missing.

In order to measure the performance of the method, we defined one ability metric and two quality metrics. Based on the results of the simulation, we compared these metrics for different values of *k* and for different amounts of missing data. We also compared the ability of the method for different amounts of missing data using optimal values of *k*.

Our findings lead us to conclude the following in response to our research questions:

- Imputation of Likert data using the *k*-NN method is feasible. Our results show that imputation was successful (in terms of quality) provided that an appropriate value of *k* was used.
- It is not best to use $k = 1$, as we have seen is common, in all situations. Our results show that using the square root of the number of complete cases, rounded to the nearest odd integer, is a suitable model for *k*.
- The outcome of the imputation depends on the number of complete cases more than the amount of missing data. The method was successful even for high proportions of incomplete cases.
- When using the IC strategy, the ability of the method increased substantially compared to the CC strategy for larger amounts of missing data, while there was no negative impact on the quality of the imputations for smaller amounts of missing data. Consequently, the IC strategy seems, from a quality perspective, safe to use in all situations.

8. Acknowledgements

This work was partly funded by The Knowledge Foundation in Sweden under a research grant for the project "Blekinge - Engineering Software Qualities (BESQ)" (http://www.bth.se/besq).

The authors would like to thank the reviewers for valuable comments that have helped to improve this paper.

9. References

[1] Batista, G. E. A. P. A. and Monard, M. C., "A Study of K-Nearest Neighbour as a Model-Based Method to Treat Missing Data", in *Proceedings of the Argentine Symposium on Artificial Intelligence* (ASAI'03), vol. 30 (2001), Buenos Aires, Argentine, pp. 1-9.

[2] Cartwright, M. H., Shepperd, M. J. and Song, Q., "Dealing with Missing Software Project Data", in *Proceedings of the 9th International Software Metrics Symposium*, 2003, Sydney, Australia, pp. 154-165.

[3] Chen, J. and Shao, J., "Nearest Neighbor Imputation for Survey Data", in *Journal of Official Statistics*, vol. 16, no. 2, 2000, pp. 113-131.

[4] Downey, R. G. and King, C. V., "Missing Data in Likert Ratings: A Comparison of Replacement Methods", in *Journal of General Psychology*, 1998, pp. 175-191.

[5] Duda, R. O. and Hart, P. E., *Pattern Classification and Scene Analysis*, John Wiley & Sons, 1973.

[6] Engels, J. M. and Diehr, P., "Imputation of Missing Longitudinal Data: A Comparison of Methods", in *Journal of Clinical Epidemiology*, vol. 56, 2003, pp. 968-976.

[7] Gediga, G. and Düntsch, I., "Maximum Consistency of Incomplete Data via Non-Invasive Imputation", in *Artificial Intelligence Review*, vol. 19, no. 1, 2003, pp. 93-107.

[8] Hu, M., Salvucci, S. M. and Cohen, M. P., "Evaluation of Some Popular Imputation Algorithms", in *Proceedings of the Survey Research Methods Section*, American Statistical Association, 1998, pp. 308-313.

[9] Huisman, M., "Imputation of Missing Item Responses: Some Simple Techniques", in *Quality and Quantity*, vol. 34, 2000, pp. 331-351.

[10] de Leeuw, E. D., "Reducing Missing Data in Surveys: An Overview of Methods", in *Quality and Quantity*, vol. 35, 2001, pp 147-160.

[11] Myrtveit, I., Stensrud, E. and Olsson, U. H., "Analyzing Data Sets with Missing Data: An Empirical Evaluation of Imputation Methods and Likelihood-Based Methods", in *IEEE Transactions on Software Engineering*, vol. 27, 2001, pp. 999-1013.

[12] Jönsson, P. and Wohlin, C., "Understanding the Importance of Roles in Architecture-Related Process Improvement", submitted for publication.

[13] Raaijmakers, Q. A. W., "Effectiveness of Different Missing Data Treatments in Surveys with Likert-Type Data: Introducing the Relative Mean Substitution Approach", in *Educational and Psychological Measurement*, vol. 59, no. 5, Oct. 1999, pp. 725-748.

[14] Robson, C., *Real World Research*, 2nd ed., Blackwell Publishing, 2002.

[15] Sande, I. G., "Hot-Deck Imputation Procedures", in Madow, W. G. and Olkin, I., eds., *Incomplete Data in Sample Surveys, Volume 3, Proceedings of the Symposium*, Academic Press, 1983, pp. 334-350.

[16] Scheffer, J., "Dealing with Missing Data", in *Research Letters in the Information and Mathematical Sciences*, vol. 3, 2002, pp. 153-160.

[17] Song, Q. and Shepperd, M., "A Short Note on Safest Default Missingness Mechanism Assumptions", in *Empirical Software Engineering*, accepted for publication, 2004.

[18] Strike, K., El Emam, K. and Madhavji, N., "Software Cost Estimation with Incomplete Data", in *IEEE Transactions on Software Engineering*, vol. 27, 2001, pp. 890-908.

[19] Troyanskaya, O., Cantor, M., Sherlock, G., et al., "Missing Value Estimation Methods for DNA Microarrays", in *Bioinformatics*, vol. 17, 2001, pp. 520-525.

[20] Wilson, D. R. and Martinez, T. R., "Improved Heterogeneous Distance Functions", in *Journal of Artificial Intelligence Research*, vol. 6, 1997, pp. 1-34.