P. Jönsson and C. Wohlin, "Benchmarking k-Nearest Neighbour Imputation with Homogeneous Likert Data", Empirical Software Engineering: An International Journal, Vol. 11, No. 3, pp. 463-489, 2006.

Benchmarking *k*-Nearest Neighbour Imputation with Homogeneous 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. Missing data are common in surveys regardless of research field, undermining statistical analyses and biasing results. One solution is to use an imputation method, which recovers missing data by estimating replacement values. Previously, we have evaluated the hot-deck *k*-Nearest Neighbour (*k*-NN) method with Likert data in a software engineering context. In this paper, we extend the evaluation by benchmarking the method against four other imputation methods: Random Draw Substitution, Random Imputation, Median Imputation and Mode Imputation. By simulating both non-response and imputation, we obtain comparable performance measures for all methods. We discuss the performance of *k*-NN in the light of the other methods, but also for different values of *k*, different proportions of missing data, different neighbour selection strategies and different numbers of data attributes. Our results show that the *k*-NN method performs well, even when much data are missing, but has strong competition from both Median Imputation and Mode Imputation for our particular data. However, unlike these methods, *k*-NN has better performance with more data attributes. We suggest that a suitable value of *k* is approximately the square root of the number of complete cases, and that letting certain incomplete cases qualify as neighbours boosts the imputation ability of the method.

1 Introduction

Missing data pose a serious problem to researchers in many different fields of research, for example artificial intelligence (Gediga and Düntsch, 2003), machine learning (Batista and Monard, 2001) and psychology (Downey and King, 1998). The

situation is, unsurprisingly, similar in software engineering (Cartwright et al., 2003; Myrtveit et al., 2001; Strike et al., 2001). The absence of data may substantially affect data analysis as statistical tests will lose power and results may be biased because of underlying differences between cases with and without missing data (Huisman, 2000). Simple ways to deal with missing data are, for example, listwise deletion, in which incomplete cases are simply discarded from the data set, or variable deletion, in which variables with missing data are discarded. However, a consequence of using a deletion procedure is that potentially valuable data are discarded, which is even worse than having missing data in the first place. Another approach, advantageous because it does not require useful data to be removed, is to use a method for imputing data. Imputation methods work by substituting replacement values for the missing data, hence increasing the amount of usable data.

A multitude of imputation methods exist (see, for example, the paper by Hu et al. (2000) for an overview), whereas this paper deals mainly with hot-deck *k*-Nearest Neighbour imputation, but also with Random Draw Substitution, Random Imputation, Median Imputation and Mode Imputation. In hot-deck imputation, a missing value is replaced by a value derived 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 Median Imputation, for example, in which a missing value is replaced with the median of the non-missing values, does not qualify as a hot-deck method (Sande, 1983). There are different ways of picking a replacement value, for example by choosing a value from one of the donors by random (Huisman, 2000) or by calculating the mean of the values of the donors (Batista and Monard, 2001; Cartwright et al., 2003).

The *k*-Nearest Neighbour (*k*-NN) method is a common hot-deck method, in which *k* donors are selected from the available neighbours (i.e., the complete cases) such that they minimise some similarity metric (Sande, 1983). The method is further described in Section 4.5. An advantage over many other methods, including Median and Mode 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 (Cartwright et al., 2003; Song et al., 2005; Strike et al., 2001) and in non-software

engineering contexts (Batista and Monard, 2001; Chen and Shao, 2000; Troyanskaya et al., 2001).

We evaluated the *k*-NN method in a previous paper, and concluded that the performance of the method was satisfactory (Jönsson and Wohlin, 2004). In order to better assess the relative performance of the method, we extend the evaluation in this paper by benchmarking the *k*-NN method against four other methods: Random Draw Substitution, Random Imputation, Median Imputation and Mode Imputation. These methods are clearly simpler in terms of imputation logic than *k*-NN, but can be said to form an imputation baseline. Thus, the main research question concerns the performance of the *k*-NN method in relation to the other methods.

The data used in the evaluation are of Likert type in a software engineering context. A Likert scale is ordinal, and consists of a number of alternatives, typically weighted from one and up, that concern level of agreement (e.g., disagree, agree, strongly agree). Such scales are commonly used when collecting subjective opinions of individuals in surveys (Robson, 2002). The evaluation is performed by running the *k*-NN method and the other imputation methods on data sets with simulated non-response.

Apart from the benchmarking, we discuss the following questions related to the *k*-NN method:

- How many donors should preferably be selected?
- At which proportion of missing data is it no longer relevant to use the method?
- Is it possible to decrease the sensitivity to the proportion of missing data by allowing imputation from certain incomplete cases as well?
- What effect has the number of attributes (variables) on the results?

The remainder of the paper is structured as follows. In Sections 2 and 3, we outline related work, describe the data used in the evaluation and discuss different mechanisms for missing data. In Section 4, we present the *k*-NN method as well as the other imputation methods against which we benchmark *k*-NN. In Section 5, we describe the process we have used for evaluating the *k*-NN method. Although the process is generic in the sense that it supports any imputation method, we focus on *k*-NN. In Section 6, we briefly describe how we instantiated the process in a simulation,

but also how we performed additional simulations with other imputation methods. In Section 7, we present the results and relate them to our research questions. In Section 8, we discuss validity threats and outline possible future work. Finally, we draw conclusions in Section 9.

2 Related Work

As Cartwright et al. (2003) point out, publications about imputation in empirical software engineering are few. To our knowledge, those that exist have focused on comparing the performance of different imputation methods. For example, Myrtveit et al. (2001) 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 $k = 1$). 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. (2001) describe a simulation of listwise deletion, mean imputation and hot-deck imputation (in fact, *k*-NN with $k = 1$), 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. (2003) themselves compare sample mean imputation and *k*-NN, and reach the conclusion that *k*-NN may be useful in software engineering research. Song et al. (2005) evaluate the difference between the missingness mechanisms MCAR and MAR (see Section 3.2) 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.

It is common to compare imputation methods in other research areas as well. Batista and Monard (2001) 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 proportion of cases with missing data is high (up to 60%). Engels and Diehr (2003) compare 14 imputation methods, among them one hot-deck method (however,

not *k*-NN), on longitudinal health care data. They report, however, that the hot-deck method did not perform as well as other methods. Huisman (2000) presents a comparison of imputation methods, including Random Draw Substitution and *k*-NN with $k = 1$. He concludes that Random Draw Substitution is among the worst performers, that the *k*-NN method performs better with more response options, but that corrected item mean imputation generally is the best imputation method. In the context of DNA research, Troyanskaya et al. (2001) report on a comparison of three imputation methods: one based on single value decomposition, one *k*-NN variant and row average. They conclude that the *k*-NN method is far better than the other methods, and also that it is robust with respect to proportion of missing data and type of data. Moreover, they recommend the use of Euclidean distance as a similarity measure. Gmel (2001) compares four different imputation methods, including single-value imputation based on median and *k*-NN with $k = 1$. He argues that single-value imputation methods are considered poor in general as they disturb the data distribution by repeatedly imputing the same value. He concludes that the *k*-NN method seems to perform better than the other methods. Chen and Åstebro (2003) evaluate six methods for dealing with missing data, including Random Draw Substitution and Mode Imputation, by looking at the sample statistics mean and variance. They report that Random Draw Substitution systematically biases both the mean and the variance, whereas Mode Imputation only systematically biases the variance.

Imputation in surveys is common, due to the fact that surveys often are faced with the problem of missing data. De Leeuw (2001) describes the problem of missing data in surveys and gives suggestions for how to deal with it. Downey and King (1998) evaluate two methods for imputing data of Likert type, which often are used in surveys. Their results show that both methods, item mean and person mean substitution, perform well if the proportion of missing data is less than 20%. Raaijmakers (1999) presents an imputation method, relative mean substitution, for imputing Likert data in large-scale surveys. 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 (2000) 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.

Gediga and Düntsch (2003) present 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. Thus, 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 4.5), in that both approaches allow values that will not contribute to the similarity measure to be missing in the donor cases.

3 Research Data

In this section, we present the data used in the evaluation. We also discuss different missingness mechanisms (i.e., different ways in which data can be missing).

3.1 Evaluation Data

The data used in the evaluation come from a case study on software architecture documentation in a large Swedish organisation. The case study is described in detail elsewhere (Jönsson and Wohlin, 2005). In the case study, a questionnaire about viewpoints on architecture documentation was distributed to employees in the organisation. For the evaluation, we chose to use the answers to six questions, selected such that we could extract as many complete cases as possible. Initially, the questions were answered by 66 persons, of which 12 (18.2%) gave incomplete answers (resulting in a data set with 7.8% missing data). Thus, the evaluation data set contained 54 complete 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 short text explaining its

meaning, and we tried to make sure that distances between two adjacent numbers were conceptually similar everywhere.

We have previously examined the original data with respect to differences between roles, and found that there are no differences for the questions involved in this evaluation (Jönsson and Wohlin, 2005). We have also sought differences based on other ways to group the data, but found none. Hence, we presuppose that the data are homogeneous. Fig. 1 shows the distribution of response options of the six questions as well as on average (rightmost bar). As can be seen, options 1 and 5 are largely underrepresented, while in particular options 3 and 4 are common answers to most of the questions.

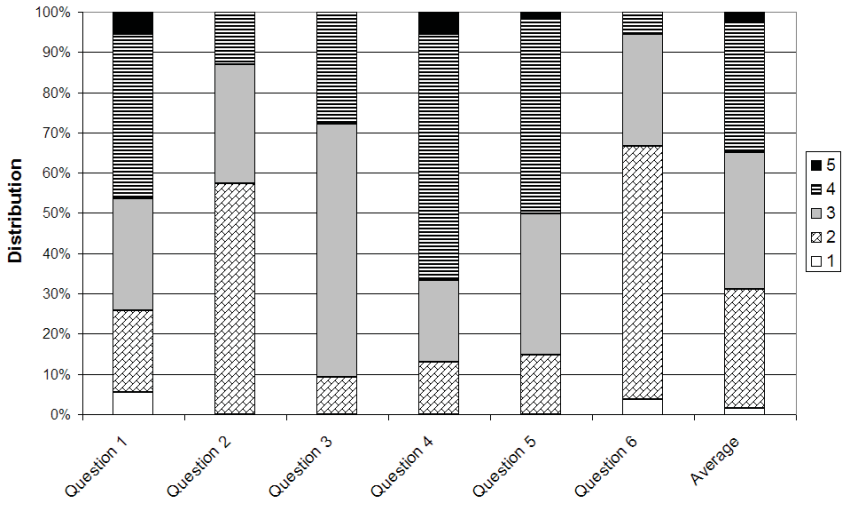

Fig. 1. Distribution of Response Options

3.2 Missing Data

There are three main ways in which data can be missing from a data set (Batista and Monard, 2001; Cartwright et al., 2003; Scheffer, 2002). These ways, or missingness mechanisms, are:

- MCAR (Missing Completely At Random), means that the missing data are 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 depend on the missing values themselves, and not on any other observed variable.

Any action for dealing with missing data must take the missingness mechanism into account. For example, to discard cases with missing data altogether is dangerous unless the missingness mechanism is MCAR (Scheffer, 2002). Otherwise, there is a risk that the remaining data are severely biased. NMAR is the hardest missingness mechanism to deal with, because it, obviously, is difficult to construct an imputation model based on unobserved data.

When data are missing from the responses to a questionnaire, it is more likely that the missingness mechanism is MAR than MCAR (Raaijmakers, 1999). 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. Thus, it is recommended to include a response option for lack of relevance. It should be noted, however, that the questions in our data did not offer such a response option.

4 Imputation Methods

In this section, we describe the *k*-NN imputation method as well as the imputation methods used for benchmarking. We divide the imputation methods into the three categories *uninformed*, *informed* and *intelligent* (see the paper by Huisman (2000) for other ways of categorising imputation methods):

- Uninformed imputation methods do not take into consideration properties of the data that are important from an imputation perspective, such as distribution of response options. Random Draw Substitution, where a replacement value is randomly drawn from the set of response options, falls into this category.
- Informed imputation methods do take data properties into consideration. Random Imputation, where a replacement value is randomly drawn from the available (observed) answers, as well as Median Imputation and Mode Imputation fall into this category.
- Intelligent imputation methods are those that base the imputation on hypothesised relationships in the data. The *k*-NN method falls into this category.

We go deeper into details for the *k*-NN method than for the other methods, in particular with respect to how the properties of the method affect the imputation results. Based on this, we differentiate between two different strategies for selecting neighbours. The standard strategy adheres to the rules of the method in that only complete cases qualify as neighbours, while the other relaxes this restriction slightly.

4.1 Random Draw Substitution

Random Draw Substitution (RDS) is an imputation method in which a missing value is replaced by a value randomly drawn from the set of available response options (Huisman, 2000). In our case, this means that we randomly generate replacement values from 1 to 5 such that all values have equal possibilities of being generated.

RDS falls into the category of uninformed imputation methods, as it does not consider data distribution or any other relevant properties. The relevance in benchmarking against RDS, or any other uninformed method for that matter, can of course be debated. However, we argue that a hallmark of any method necessarily must be to beat the entirely random case.

4.2 Random Imputation

Hu et al. (1998) describe generic Random Imputation (RI) as a method where replacement values are drawn at random from observed data, given some sampling

scheme. In our use of the method, we replace a missing value for a particular question with a value drawn randomly from all available answers to the question. Thus, we effectively set the probabilities of the response options in accordance with the distribution of response options for the question. This means that RI can be categorised as an informed imputation method.

By obeying the distribution of observed response options, we can expect RI to outperform RDS unless the possible response options are equally distributed for each question. This is not the case in our data, where response options 1 and 5 in particular are largely underrepresented (see Fig. 1).

4.3 Median Imputation

Due to the fact that our original data are ordinal, we impute based on median rather than mean. In Median Imputation (MEI), a missing value is replaced by the median of all available answers to the question. As with any type of single-value imputation, this method disturbs the distribution of response options, since the same value is used to replace each missing value for a particular question (Gmel, 2001).

If the number of available answers to a question is even, the median may become a non-integer value. Since non-integer values are not compatible with ordinal data, we round the value either downwards or upwards at random (for each missing value) in order to get an integer value.

MEI is highly sensitive to the distribution of response options for a question. More specifically, if the median corresponds to a response option with low frequency, the percentage of correct imputations will be low. Conversely, if the median corresponds to a frequent response option, MEI will have good performance.

4.4 Mode Imputation

Mode Imputation (MOI) is similar to MEI, except the mode is used instead of the median. As with MEI, MOI disturbs the distribution by imputing the same value for all missing values for a particular question.

A problem with using the mode as replacement value is that the distribution of available answers may be multimodal (i.e., have several modes). If that is the case, we obtain a unique replacement value by randomly selecting one of the modes for each missing value.

If the mode corresponds to a response option with high frequency compared to the other response options, the percentage of correct imputations will be high. Otherwise, that is if the difference in frequency to the next most common value is small, the imputation performance decreases. Similarly, if the mode is a response option towards one of the ends of the scale, and a response option in the other end is common as well, the relative error of incorrectly imputed values will be high.

4.5 *k*-Nearest Neighbour

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, the paper by Wilson and Martinez (1997))

$$E(a,b) = \sqrt{\sum_{i \in D} (x_{ai} - x_{bi})^2} \quad \textbf{(1)}$$

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. (2001) and Troyanskaya et al. (2001). The *k*-NN method does not suffer from the problem with reduced variance to the same extent as single-value 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 the 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

$$E\left(\text{Bridget}, \text{Susan}\right) = E\left(\text{Eric}, \text{Susan}\right) = \sqrt{2 \times \left(4-2\right)^2} \approx 2.8$$

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	-	4	2	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 (Batista and Monard, 2001). 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 (1973) 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. Cartwright et al. (2003), 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 $k = 2$. Several others use $k = 1$, for example Myrtveit et al. (2001), Strike et al. (2001), Huisman (2000) and Chen and Shao (2000). Batista and Monard (2001), on the other hand, report on $k = 10$ for large data sets, while Troyanskaya et al. (2001) 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.

Neighbour Strategy

In hot-deck imputation, and consequently in k-NN imputation, only complete cases can be used for imputing missing values (Batista and Monard, 2001; Cartwright et al., 2003; Sande, 1983). In other words, only complete cases qualify as 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 (Complete Case) strategy.

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 (Incomplete Case) strategy.

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 handle large proportions of missing data better than the CC strategy.

5 Evaluation Process

The process for evaluating the *k*-NN method consists of the three main steps *data removal*, *imputation* and *evaluation* (illustrated in Fig. 2). In this section, we describe each step with respect to consumed input, responsibility, calculated metrics and produced output. The process is generic in that it does not depend on any particular data removal mechanism or imputation method. Here, we present it using *k*-NN as the imputation method. In Section 6, we detail the actual simulation of the process and describe how we have reused parts of it for benchmarking *k*-NN against the other imputation methods.

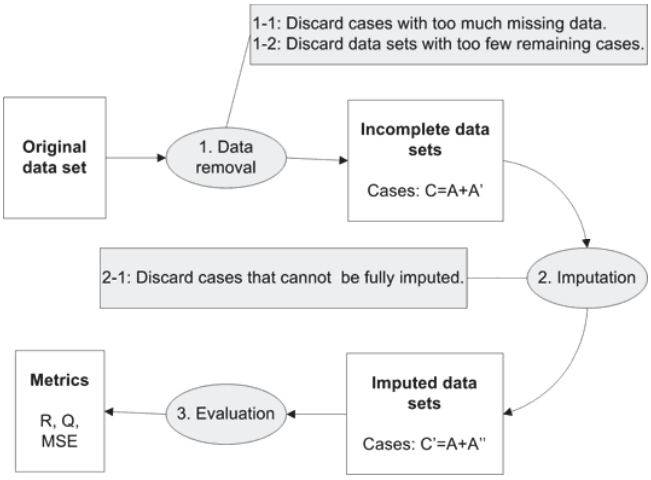

Fig. 2. Evaluation Process Outline

5.1 Data Removal – Step 1

Input to the data removal step is a data set where no data are missing. The responsibility of the step is to generate one or more artificially incomplete data sets from the complete data set, in order to simulate non-response. The generated data sets are subsequently sent to the imputation step.

In order to obtain a wide range of evaluation conditions for the *k*-NN method, it would be beneficial to use both the MCAR and MAR missingness mechanisms when generating incomplete data sets. In order to remove data to simulate MAR, a model for the non-responsiveness is required. For example, in a longitudinal study of health data, Engels and Diehr (2003) devised a model where the probability of removing a value increased if the previous value had been removed, thereby modelling a situation where a serious health condition could result in repeated non-response.

Possible models for simulating MAR in our data could involve, for example, experience in software architecture issues, organisational role or number of years in the industry, where different values would yield different probabilities for removing data from a case. However, given that our data are homogeneous, these models would not affect the imputation in other ways than would an MCAR-based model. Thus, we use only MCAR, and remove data in a completely random fashion.

We do not try to simulate different sources of missing data (e.g., lack of relevance, simple omission), which means that 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*. These are called reduction limits because they prevent the data from being reduced to an unusable level. The effects of the parameters can be seen in Fig. 2. 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.

These limits mean, in a way, that 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 percentages and the actual percentages of missing data are presented. When analysing and discussing the results, it is the actual percentages that are used, though.

5.2 Imputation – Step 2

Input to the imputation step is the incomplete data sets generated in the data removal step. Here, each data set is fed to the imputation method in order to have its missing values imputed. We exemplify this step using the *k*-NN method.

With the *k*-NN method, several imputations using different *k*-values and different neighbour strategies are performed for each incomplete data set. As discussed earlier, a missing value is replaced by the median of the answers given by the *k* nearest neighbours, which means that the replacement value may become a non-integer value if *k* is even. However, since the data in the data set are of Likert type, non-integer values are not permitted. To avoid this problem, only odd *k*-values are used.

The *k* cases with least distances are chosen as donors, regardless of ties among the distances. That is, 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. This procedure is safe since the cases in the original data set are not ordered in a way that could affect the imputation.

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 Fig. 2.

The output from the imputation step is a number of imputed data sets, possibly several for each incomplete data set generated in the data removal step (depending on the imputation method used and its parameters). 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.

5.3 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 (or error) metric.

We define the *ability* metric as

$$R = \frac{A''}{A'} \tag{2}$$

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

$$Q = \begin{cases} \dfrac{B'}{B} & \text{if} \quad B > 0 \\ \text{undefined} & \text{if} \quad B = 0 \end{cases} \tag{3}$$

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

$$MSE = \begin{cases} \dfrac{\sum_i \left(x_i - \hat{x}_i\right)^2}{B - B'} & \text{if} \quad B > 0,\ B' < B \\ \text{undefined} & \text{if} \quad B = 0 \text{ or } B' = B \end{cases} \tag{4}$$

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 metric and the mean square error are invalid when the ability metric 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, and for *k*-NN, 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.

6 Simulation

The previous section described the outline of the evaluation process. In this section, we briefly address the actual simulation of the process and which parameters we used to control it. We also provide some information about the simulation software used. Finally, we explain how we reused the process to run additional simulations with different imputation methods in order to obtain benchmarking figures.

6.1 Parameters

Each of the three steps in the process described in Section 5 is guided by a number of parameters. As discussed, two reduction limits, the case reduction limit and the data set reduction limit, constrain the data removal step. Based on the number of attributes and cases in the original data set, we used the following values in the simulation:

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

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 5.1, 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 controlling parameters depend on the imputation method. For the *k*-NN method, 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.

6.2 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 and Microsoft Access were used for analysing some of the results from the evaluation step.

In order to validate that the application worked correctly with respect to the *k*-NN method, 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 imputation 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 implementation of the *k*-NN method 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 were randomly picked and checked for feasibility and correctness.

The implementation of the remaining imputation methods was deemed correct through code reviews.

6.3 Process Reuse

To be able to benchmark *k*-NN against the other imputation methods, we took advantage of the fact that we could reuse the results from the data removal step. We instructed the application to save the 12 000 incomplete data sets before passing them on to the imputation step. To obtain the benchmarking figures, we ran the simulation again for each of the other imputation methods except Random Draw Substitution, this time skipping the data removal step and feeding the saved incomplete data sets directly to the imputation step. This way, the other imputation methods worked with the same incomplete data sets as the *k*-NN method.

Moreover, we had constructed the application to accept a reference data set in the evaluation step, in case the data removal step was omitted. This allowed us to obtain values for the *R*, *Q* and *MSE* metrics.

7 Results

In this section, we present the results from the simulations of the *k*-NN method and the other imputation methods. First, we provide descriptive statistics of the incomplete data sets generated in the initial simulation (and reused in subsequent simulations). Then, we address the questions posed in Section 1 as follows:

- We compare the results for different values of *k* in order to find the appropriate number of donors. In doing so, we also look at differences between the CC and IC strategies, to assess whether or not the IC strategy is appropriate to use.
- We compare the results from the original simulation with results from simulations using data sets with 12 and 18 attributes, respectively.
- We look at how the performance of *k*-NN changes for different percentages of missing data, in order to find a limit where the method stops being usable.
- Finally, we compare the performance of *k*-NN with the performance of the other imputation methods described in Section 4, in order to be able to judge its relative effectiveness.

7.1 Incomplete Data Sets

As discussed in Section 5.1, 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 ($\bar{C}$), which is the upper limit of *k*.

Table 2. Overview of Incomplete Data Sets

Pct.	**Mean missing data (%)**	***s***	$\bar{C}$	$\bar{A}$	**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

7.2 Comparison of *k*-Values and Strategies

For each percentage of missing data, we plotted the ability metric and the quality metrics for different values of *k* and for both of the neighbour selection strategies. It is not necessary to show all the 24 resulting diagrams, as 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 Fig. 3.

The diagrams in the figure show the ability and quality for both the CC strategy and the IC strategy. 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 ($\bar{A}$) in the data sets for this percentage (see Table 2). 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. (2001). This is true for both strategies. Because of the priorities of the performance metrics, discussed in Section 5.3, 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 are 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.

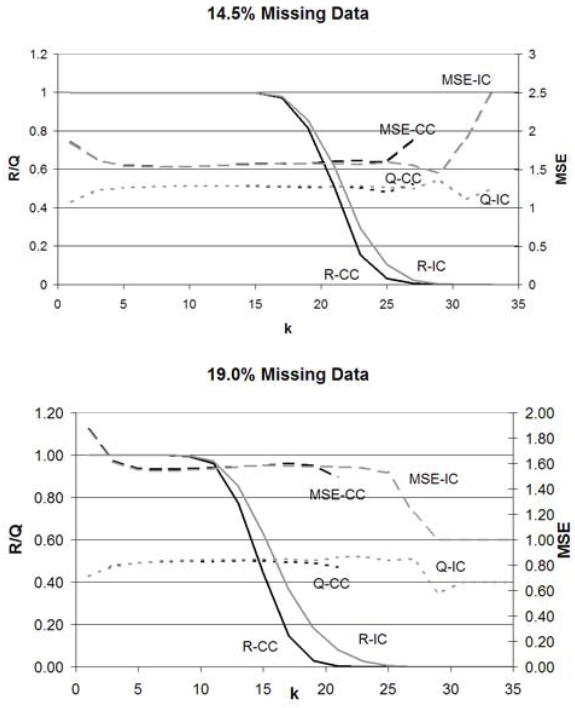

Fig. 3. Performance at 14.5% and 19.0% Missing Data, CC and IC

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 4.5 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 (*R*) 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 proportion 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 and Table 4 show the observed optimal *k*-values for the CC strategy and the IC strategy, respectively, given the average number of complete cases for the simulated percentages. It can be seen that the optimal value of *k* for a certain number of neighbours is the same regardless of strategy. The tables also show the values of *R*, *Q* and *MSE* for each optimal *k*-value. As can be seen, the quality metrics get gradually worse as the number of complete cases, and thus the ability of the method, decreases.

Table 3. Optimal *k*-Values with *R*, *Q* and *MSE* for the CC Strategy

$\overline{A}$ =	**39.8**	**28.8**	**20.4**	**14.2**	**9.6**	**6.3**	**4.0**	**2.4**	**1.5**	**0.8**	**0.4**	**0.2**
k	7	7	7	7	5	3	1	1	1	1	1	1
R	1.00	1.00	1.00	1.00	0.99	0.98	0.99	0.93	0.80	0.57	0.37	0.20
Q	0.52	0.51	0.51	0.50	0.48	0.47	0.42	0.42	0.41	0.41	0.40	0.40
MSE	1.56	1.54	1.53	1.55	1.58	1.63	1.88	1.89	1.94	1.93	1.95	1.95

Table 4. Optimal *k*-Values with *R*, *Q* and *MSE* for the IC Strategy

$\overline{A}$ =	**39.8**	**28.8**	**20.4**	**14.2**	**9.6**	**6.3**	**4.0**	**2.4**	**1.5**	**0.8**	**0.4**	**0.2**
k	7	7	7	7	5	3	1	1	1	1	1	1
R	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.98	0.92	0.82	0.69	0.56
Q	0.52	0.51	0.51	0.50	0.49	0.47	0.43	0.42	0.42	0.42	0.42	0.42
MSE	1.56	1.54	1.52	1.54	1.57	1.62	1.87	1.88	1.90	1.90	1.90	1.90

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 (1973). 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. This concurs with our own requirements on *k*. Thus, we have chosen to examine $k = \text{RoundOdd}(\sqrt{\overline{A}})$, which is 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 5. 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 relative errors in *R*, *Q* and *MSE* between the non-matching calculated and optimal *k*-values are for the CC strategy within the ranges 0–0.80%, 0.63–4.03% and 0.44–4.19%, respectively, and for the IC strategy within the ranges 0–0.45%, 0.80–4.42% and 0.31–4.48%, respectively. It should be noted that, since lower *k* means that fewer donors are required, the imputation ability errs on the positive side. Furthermore, as the calculated *k* does not drop to 1 in the mid-range of missing data, both *Q* and *MSE* will be better than their initial, unfavourable values (see Fig. 3).

Table 5. Optimal k vs. Calculated k

$\overline{A}$ =	**39.8**	**28.8**	**20.4**	**14.2**	**9.6**	**6.3**	**4.0-**
Optimal	7	7	7	7	5	3	1
Calculated	7	5	5	3	3	3	1

7.3 Comparison of Attribute Counts

As mentioned, the number of complete cases for a data set with a certain percentage of 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 with the *k*-NN method. 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. Since the number of cases was unchanged in these extended data sets, a certain percentage of removed data yielded more incomplete cases compared to the data set with six attributes. Consequently, the ability of the method drops quicker with more attributes.

For 12 attributes and 4.9% missing data (5% removed), using $k = 3$ and the IC strategy results in $Q \approx 0.65$ and $MSE \approx 1.15$. The results are the same with 18 attributes, also with 4.9% missing data (5% removed), $k = 3$ and the IC strategy.

The diagrams in Fig. 4 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 $k = \text{RoundOdd}(\sqrt{A})$ with higher numbers of attributes as well, although comparing the optimal *k*-values and the calculated ones reveals that the optimal values are slightly lower for low percentages of missing data. As with six attributes, both Q and MSE get gradually worse as the percentage of missing data increases. For 12 attributes, the method can maintain maximum ability (at $k = 1$) up to 19.8% missing data (20% removed), whereas for 18 attributes, the corresponding limit is at 14.9% missing data (15% removed).

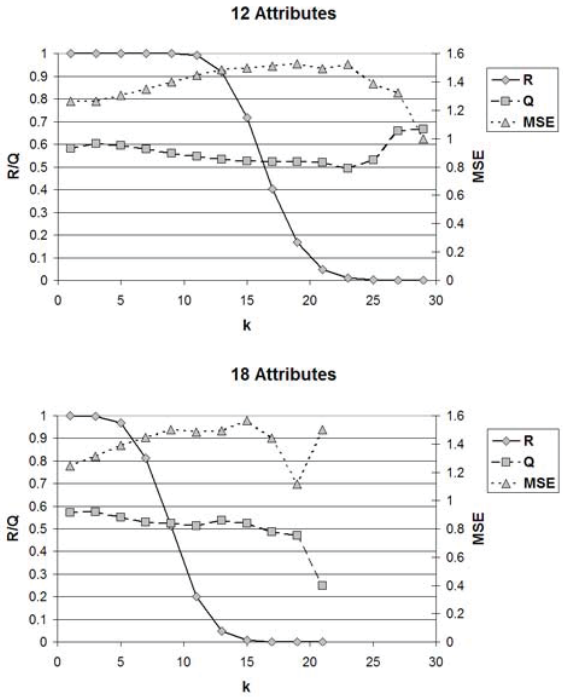

Fig. 4. 9.9% Missing Data, 12 (*top*) and 18 (*bottom*) Attributes, IC

7.4 Comparison of Percentages

In addition to comparing the ability and quality for different *k*-values, we compared the ability of the method for different proportions of missing data, using for each percentage the optimal *k*-value found earlier. The diagram (for six attributes) can be seen in Fig. 5 (for the raw numbers, see Table 3 and Table 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 are missing. Because the comparison of *k*-values showed that the IC strategy does not give lower precision (*Q*) or larger mean square error (*MSE*) than the CC strategy, we consider it more favourable regardless of the proportion of missing data.

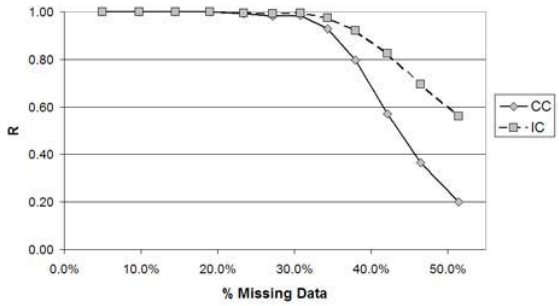

Fig. 5. Ability vs. Proportion of Missing Data

7.5 Benchmarking

Here, we present the benchmarking of the *k*-NN method against the four other imputation methods. As basis for the comparisons, we use only the results from the original data set with six attributes imputed using the IC strategy.

For Random Draw Substitution, where the selection of replacement values does not depend on the data distribution, it is straightforward to calculate the expected values of *R*, *Q* and *MSE*. For the informed imputation methods, we have performed additional simulations reusing the incomplete data sets generated initially (see Section 6.3).

Random Draw Substitution

With RDS, we randomly draw a replacement value from the set of response options (i.e., from 1 to 5). Each response option has a probability of 0.2 for being selected, which means that the expected value of *Q* is 0.2. Since this imputation technique

never can fail to impute (as opposed to *k*-NN, which fails when there are too few neighbours), the expected value of *R* is 1.

The expected value of *MSE* can be calculated given the average distribution of response options in the original data set. Let *P*(*z*) denote the probability that the correct value of a missing value is *z*. Given the fact that the probability of imputing any value is 0.2, the expected total *MSE* (*TMSE*) for all imputations can be expressed as

$$TMSE = 0.2 \times \sum_x \sum_y \left(P(y) \times |x - y|^2 \right) \quad \textbf{(5)}$$

where *x* is the imputed value and *y* is the correct value. The problem is that *TMSE* includes the errors also when the correct value is imputed. These errors are all zero, which means that *TMSE* is lower than the expected *MSE*, which is defined as the relative error for the incorrectly imputed values. To obtain the correct *MSE*, *TMSE* must be divided by the probability of imputing an incorrect value, thus

$$MSE = \frac{TMSE}{0.8} \quad \textbf{(6)}$$

With $P(z) = \{0.015, 0.296, 0.340, 0.327, 0.022\}$, $1 \le z \le 5$, which is the average distribution of response options in the original data, we obtain $MSE \approx 3.465$.

Random Imputation

With RI, we draw a replacement value from the set of available answers to the current question, which means that the distribution of response options is taken into consideration. Given that the missingness mechanism for our data is MCAR, the distributions in the incomplete data sets can be assumed to equal the distribution in the complete data set. Thus, RI can be expected to perform reasonably well. With MAR as the missingness mechanism, the performance could be worse.

As with RDS, this imputation technique cannot fail, and the expected value of R is 1. The simulation with RI as the imputation method resulted in $Q \approx 0.41$ and $MSE \approx 1.97$ averaged over all 12 000 incomplete data sets. The averages for each individual percentage did not deviate much from the total averages.

Median Imputation

With MEI, the replacement value is the median of all available answers to the current question. As pointed out in Section 4.3, the frequency of the response option that corresponds to the median has large effect on the imputation performance. Fig. 1 shows that our data are favourable for MEI in this aspect, since most questions have frequent median response options.

MEI cannot fail to impute, which means that the expected value of R is 1. The simulation with MEI as the imputation method gave the values $Q \approx 0.50$ and $MSE \approx 1.59$ averaged over all incomplete data sets. The averages of Q for individual percentages did not differ much from the total average. However, for *MSE*, the averages ranged from 1.55 to 1.61.

Mode Imputation

With MOI, the replacement value is the mode of all available answers to the current question. If the distribution of answers is multimodal, one of the modes is selected randomly. As described in Section 4.4, MOI does not perform well if the response option that corresponds to the mode is only slightly more frequent than other response options. Fig. 1 clearly shows that this is not the case in our data, which means that we can expect MOI to perform well.

MOI cannot fail to impute, which means that the expected value of R is 1. The simulation with MOI as the imputation method resulted in $Q \approx 0.54$ and $MSE \approx 1.85$ averaged over all incomplete data sets. The variations in average Q and average *MSE* for individual percentages were noticeable; Q varied from 0.51 to 0.56, and *MSE* varied from 1.79 to 1.90.

7.6 Summary and Interpretation of the Results

The 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. Table 6 presents an overview of the comparisons made in evaluating *k*-NN, whereas Table 7 shows an overview of the benchmarking against the other imputation methods.

Table 6. Results Overview

	CC	**IC**
6 attributes	• *R* starts to drop at around 30% missing data. • With maximum ability, *Q* is at best 0.52 and at worst 0.42. • With maximum ability, *MSE* is at best 1.53 and at worst 1.88.	• The ability drops less drastically than CC when the percentage of missing data increases. • *R* starts to drop between 30 and 35% missing data. • *Q* and *MSE* are similar to when using CC.
12 attributes	-	• *R* drops earlier (as there are fewer complete cases), at around 20% missing data. • *Q* is higher, at best 0.65. • *MSE* is lower, at best 1.15.
18 attributes	-	• *R* drops even earlier, at around 15% missing data, than with 12 attributes. • *Q* and *MSE* are similar to when using 12 attributes.

Table 6 shows that the IC strategy is favourable over the CC strategy, since it allows *k*-NN to maintain high ability for higher percentages of missing data, while the precision and mean square error are equally good. In addition, Fig. 5 shows that, when using the IC strategy, nearly 60% of the incomplete cases could be saved when 50% of the data were missing.

It can be seen that *k*-NN performs better with more attributes, both with respect to precision and mean square error. This is due to the fact that with more attributes, the method has more information available to discriminate between neighbours when it comes to distance.

Table 7. Benchmarking Overview

	R	***Q***	***MSE***
k-NN (IC), 6 attr.	1 (up to 30-35% missing data)	0.42 to 0.52	1.53 to 1.88
k-NN (IC), 12/18 attr.	1 (up to 15-20% missing data)	up to 0.65	down to 1.15
RDS	1	0.2	3.465
RI	1	0.41	1.97
MEI	1	0.50	1.55 to 1.61
MOI	1	0.51 to 0.56	1.79 to 1.90

It can be seen in Table 7 that RDS, as expected, does not perform very well. Comparing with the values of *Q* and *MSE* for *k*-NN, it is clear that *k*-NN easily outperforms the entirely random case. Furthermore, RI performs much better than RDS. The precision (*Q*) is twice as high, and the error (MSE) is much lower. However, compared to *k*-NN, RI falls short.

MEI touches upon the six-attribute *k*-NN in terms of both precision and mean square error, and given that the ability (*R*) is always 1, it seems to be a viable alternative. Disadvantages of MEI are that it is more sensitive than *k*-NN to the distribution of response options (see Section 4.3), and that it does not perform better when the number of attributes increases. Furthermore, as with all single-value imputation, MEI may result in a disturbed data distribution, which becomes particularly noticeable when much data are missing.

MOI does perform slightly better than the six-attribute *k*-NN with respect to precision (*Q*), but performs worse with respect to relative error (*MSE*). As with MEI, MOI is sensitive for the distribution of response option and does not improve with more attributes. The reason for MOI having worse *MSE* than MEI is that the modes of the majority of the questions in our data set correspond to response options 2 or 4, which do not dominate the distributions. Thus, if the mode is not the correct value, the error will be rather large.

With 12 or 18 attributes, *k*-NN outperforms both MEI and MOI. These methods does not scale with respect to number of attributes, since they only work with one attribute at a time.

Judging from the results, *k*-NN proved to have good performance. However, both Median Imputation and Mode Imputation could compete with *k*-NN, given that both these methods were favoured by the distribution of our data. Median Imputation had similar precision and similar relative error, whereas Mode Imputation had slightly better precision, but worse relative error. Both methods will always have maximum ability (i.e., save all incomplete cases), which makes them attractive when much data are missing and there are many incomplete cases.

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.

8 Validity and Future Work

In this section, we discuss threats to the validity of the evaluation and outline possible future work.

8.1 Threats to Validity

In the *k*-NN method, we used Euclidean distance as the similarity measure. However, since the data were of Likert type (i.e., on an ordinal scale), it is 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 *k*-NN imputations were successful after all.

In step 1 of the evaluation, we removed data from the original data set completely at random, which means that the missingness mechanism was MCAR. It is more likely, though, that missing responses to a questionnaire are MAR, as pointed out by Raaijmakers (1999). In other words, the missingness mechanism used in the evaluation did not fully represent a real-world situation. Due to the nature of our data, we could not avoid this problem.

It may be dangerous to use incomplete cases as donors when the missingness mechanism 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. Hence, the outcome of the imputation is sensitive to disturbances in the data, such as outliers. 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. Furthermore, that *k*-NN in our case generates replacement values based on the median of the donors' values should alleviate the effect of outliers.

A threat to the evaluation of *k*-NN with 12 or 18 attributes is that we created these extended data sets by appending one or two copies of each case to itself. This means that the similarity, if any, between two cases is duplicated as well. In a real data set with 12 or 18 attributes, two cases could be similar for six of the attributes, but different for six other attributes. This means that the performance metrics for 12 and 18 attributes may be overly positive.

8.2 Future Work

Due to the nature of our data, we used only MCAR as missingness mechanism (see Section 5.1). In future work, it would be interesting to study imputation of Likert data with systematic differences that allow MAR missingness. For example, Song et al.

(2005) have concluded that the missingness mechanism does not significantly affect the *k*-NN method in the context of software project effort data.

Each of the questions in the original questionnaire used a Likert scale with five response options. However, it is also common to use Likert scales with more response options, for example seven or ten. Hypothetically speaking, *k*-NN should gain from the fact that more response options means that it would be easier to differentiate between neighbours (and find close donors), but should lose from the fact that more response options makes it easier for two respondents to answer similarly, yet differently. The other imputation methods would likely have worse performance, as more response options, provided they were used, would result in a wider distribution with smaller frequencies for individual response options. These hypotheses would be interesting to test in future work.

9 Conclusions

In this paper, we have presented an evaluation of the performance of the *k*-Nearest Neighbour imputation method when using homogeneous Likert data. This type of ordinal data is common in surveys that collect subjective opinions from individuals. We performed the evaluation by simulating non-responsiveness in questionnaire data and subsequent imputation of the incomplete data.

Since we simulated the evaluation process, we were able to obtain great variation in the imputation parameters and operate on a large number of incomplete data sets. In the main 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 proportions of missing data. We also

compared the ability of the method for different proportions of missing data using optimal values of *k*. Furthermore, we performed additional simulations with more attributes, in order to see how the number of attributes affected the performance of the method. Finally, we benchmarked the method against four other imputation methods, in order to be able to assess its relative effectiveness. The methods were Random Draw Substitution, Random Imputation, Median Imputation and Mode Imputation. The benchmarking was performed through additional simulations where the *k*-NN method was replaced by the other methods.

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

- **What is the performance of the *k*-NN method in relation to the other methods**? Our results show that the *k*-NN method performed well when imputing homogeneous Likert data, provided that an appropriate value of *k* was used. It outperformed both Random Draw Substitution and Random Imputation, while both Median Imputation and Mode Imputation performed equally good or slightly better. However, it is clear that our data were favourable both for Median Imputation and Mode Imputation. With a different distribution of response options, these methods could perform worse, whereas the *k*-NN method should not, given that it is less sensitive to the data distribution.
- **How many donors should preferably be selected**? 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*.
- **At which proportion of missing data is it no longer relevant to use the method**? The outcome of the imputation depends on the number of complete cases more than the proportion of missing data. The method was successful even for high proportions of incomplete cases. For example, with six attributes and the IC strategy, the method had close to maximum ability when 95% of the cases were incomplete. Thus, we are confident that the method would have been able to handle our initial situation with missing data (see Section 3.1) very well.

- **Is it possible to decrease the sensitivity to the proportion of missing data by allowing imputation from certain incomplete cases as well**? When using the IC strategy, the ability of the method increased substantially compared to the CC strategy for larger proportions of missing data, while there was no negative impact on the quality of the imputations for smaller proportions of missing data. Consequently, the IC strategy seems, from a quality perspective, safe to use in all situations.
- **What effect has the number of attributes (variables) on the results**? The *k*-NN method proved to scale well to more attributes, as both the precision and the mean square error improved for 12 and 18 attributes compared to six attributes. It is also evident that the other imputation methods are not positively affected by the number of attributes, as they do not make use of the additional amount of information.

10 Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments that have allowed us to improve the paper significantly. 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).

11 References

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 3rd Argentine Symposium on Artificial Intelligence, vol. 30 (2001), Buenos Aires, Argentine, pp. 1-9.

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.

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

Chen, G. and Åstebro, T., "How to Deal With Missing Categorical Data: Test of a Simple Bayesian Method", in Organizational Research Methods, vol. 6, 2003, pp. 309-327.

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.

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

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.

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.

Gmel, G., "Imputation of Missing Values In the Case of a Multiple Item Instrument Measuring Alcohol Consumption", in Statistics in Medicine, vol. 20, 2001, pp. 2369-2381.

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.

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

Jönsson, P. and Wohlin, C., "Evaluation of k-Nearest Neighbour Imputation Using Likert Data", in Proceedings of the 10th International Metrics Symposium, Sep. 14-16, 2004, Chicago, USA, pp. 108-118.

Jönsson, P. and Wohlin, C., "Understanding the Importance of Roles in Architecture-Related Process Improvement – A Case Study", in Proceedings of the 6th International Conference on Product Focused Software Process Improvement, June 13-15, 2005, Oulu, Finland, pp. 343-357.

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

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.

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.

Robson, C., Real World Research, 2nd ed., Blackwell Publishing, 2002.

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.

Scheffer, J., “Dealing with Missing Data”, in Research Letters in the Information and Mathematical Sciences, vol. 3, 2002, pp. 153-160.

Song, Q., Shepperd, M. and Cartwright, M. H., “A Short Note on Safest Default Missingness Mechanism Assumptions”, in Empirical Software Engineering, vol. 10, 2005, pp. 235-243.

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.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D. and Altman, R. B., “Missing Value Estimation Methods for DNA Microarrays”, in Bioinformatics, vol. 17, 2001, pp. 520-525.

Wilson, D. R. and Martinez, T. R., “Improved Heterogeneous Distance Functions”, in Journal of Artificial Intelligence Research, vol. 6, 1997, pp. 1-34.