Classification based on Neighborhood from Datasets with Low Quality Data

J.M. Cadenas[1] M.C. Garrido[1] R. Martínez[1] A. Muñoz-Ledesma[2]

[1]Dep. of Information Engineering and Communication, University of Murcia. Murcia, Spain
jcadenas@um.es, carmengarrido@um.es, raquel.m.e@um.es
[2]Dep. of Didactics of Mathematical and Social Sciences, University of Murcia. Murcia, Spain
amledesma@um.es

Abstract

Currently there are not many data mining method available to solve the classification task in datasets with low quality values. In this paper we propose a method of imputation/classification based on neighborhood that can work with nominal and numerical attributes which can contain low quality values. Performing a series of experiments we observe that the method not only is competitive to other similar method when working with datasets without low quality values, but it also obtains robust results when working with datasets with low quality values.

Keywords: Low quality data, Nearest neighbor, Classification, Fuzzy distance

1. Introduction

Within the data mining phase of the Intelligent Data Analysis process, the classification task has always been a challenging problem [1], [9]. Many methods and algorithms have been developed to address this issue. However, these methods and algorithms must work with information that is not always precise and accurate.

Nevertheless, imperfect information or low quality data inevitably appear in real world applications [16], [17]. The errors in the instruments and/or the corruption due to noise during experiments may lead to the obtaining of information with incomplete data when a value of a specific attribute is being obtained. In other cases, the extraction of accurate information can be excessively expensive or unfeasible. Moreover, it could be useful to complement the available data with additional information from an expert. This information is normally obtained by imprecise data values such as: interval data, fuzzy concepts, etc. For these cases, it is necessary to incorporate the handling of information with attributes which may present imperfect values in the classifier's learning and classification phases.

One of the best known methods for classification is the k nearest neighbors method (KNN), where k is the number of neighbors considered, [8]. In this work we extend this well known classification method in order to be able to work with low quality data or imperfect information.

KNN is an instance-based method. To classify a new instance, KNN computes the k nearest neighbors and a class value from them is obtained. The KNN method is a predictive method that can infer both nominal attributes (the most common attribute value between the k nearest neighbors) and numerical attributes (the average of the values of the k nearest neighbors). We are going to consider imputation when the attribute to infer is not the class attribute and classification when the class is inferred. This method lacks learning phase. KNN method does not create explicit models as a decision tree or a set of rules, since the own dataset is used as a "vague model". Thus, the method can be adapted easily to predict any attribute.

However, KNN method suffers from some drawbacks such as the need of a large memory requirements to store all the examples forming the training set, the low efficiency during the working of the decision rule due to the large calculation of the distances between the test examples and training examples and the little tolerance to noise since all examples are used as relevant [10].

Nonetheless, the most important limitation of the three mentioned occurs when trying to find the nearest examples to the example to be inferred. In this case the issue is critical because the method must cover all the examples in the dataset and depending on the number of examples of this and the number of attributes in each example, the time taken may be excessive. Therefore, the dataset size is a problem. However, in literature there are methods that attempt to solve this limitation, creating reduced training sets only composed of prototype examples. A review of these methods can be found in [10].

The KNN method, like many conventional methods of Intelligent Data Analysis, does not consider potential sources of imperfect information that may affect the input data. As a result, incomplete, imprecise and uncertain data are usually discarded and ignored of the input dataset and subsequently in the imputation/classification process.

In this situation, it would be interesting to extend the KNN method in order to be able to work with

© 2015. The authors - Published by Atlantis Press

imperfect data or low quality data.

In this paper we are going to present our proposal to carry out the extension of the KNN method to impute/classify from low quality data (denoted by $\mathrm{KNN_{LQD}}$ method). Thus, in Section 2 the $\mathrm{KNN_{LQD}}$ method to impute/classify from a dataset with low quality values is exposed, describing all its components. In Section 3, some experiments are performed in order to measure the effectiveness of the proposal with a number of datasets with and without low quality values. In these experiments, we will focus on the classification task. Finally, the conclusions are presented.

2. $\mathrm{KNN_{LQD}}$: k-nearest neighbors method from low quality data

Suppose the set of examples E, described by n attributes $x = \{x_1, x_2, \ldots, x_n\}$ with domains $\Omega_{x_1}, \Omega_{x_2}, ..., \Omega_{x_n}$ that can be nominal and numerical, where we consider without loss of generality that n-th attribute, x_n, is the class attribute. The domain of this n-th attribute, Ω_{x_n}, is composed of I classes represented by $\{\omega_1, \omega_2, \ldots, \omega_I\}$.

The attributes of examples can be expressed using low quality values of different types:

- The numerical attributes may be defined by crisp, fuzzy and interval. To homogenize their representation, these values are described internally by trapezoidal fuzzy membership functions as it is shown in Figure 1. In this way, the method works in all cases with the four representative values of such functions (a, b, c, d) (lower limit a, upper limit d, lower support limit b, and upper support limit c) defined according to Figure 1.
- Nominal attributes can be defined by crisp values and crisp/fuzzy subsets. Again, to unify their representation in the method, all nominal values are represented internally by crisp sets $\{h_1, h_2, \ldots, h_n\}$ and fuzzy sets $\{\mu(h_1)/h_1, \mu(h_2)/h_2, \ldots, \mu(h_n)/h_n\}$ of domain values. Thus, a nominal crisp value is represented by a crisp set composed of a simple domain value.
- In addition, both numerical and nominal attributes allow missing values. These missing values are represented as follows: a) for numerical attributes, a missing value is represented as the trapezoidal membership function (min, min, max, max) (Figure 1) where min and max correspond to the minimum/maximum values for this attribute in the dataset when the example class is missing or correspond to the minimum/maximum values, for this attribute, in the examples with equal class to the class known of the example; b) for nominal attributes, a missing value is represented by a crisp set containing all values of the domain of that attribute.

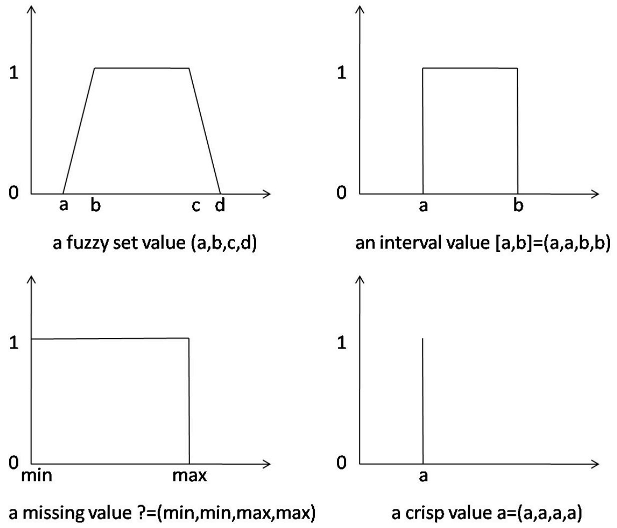
1
0
a
b
c
d
a fuzzy set value (a,b,c,d)
1
0
a
b
an interval value [a,b]=(a,a,b,b)
1
0
min
max
a missing value ?=(min,min,max,max)
1
0
a
a crisp value a=(a,a,a,a)

Figure 1: Numerical attributes as quadruples

As mentioned, the $\mathrm{KNN_{LQD}}$ method allows us to perform the imputation of missing values of a given example based on the nearest neighbors to that example. When the attribute to impute is the class attribute, we perform the classification task.

To perform the imputation/classification based on neighborhood, it is necessary to obtain the nearest neighbors using a distance function $d_{LQD}(\cdot, \cdot)$. The different characteristics and possible ways to define this function is discussed later in this section, although in a general way we must comment that this function will be able to work with numerical and nominal attributes defined by low quality values.

Using the $d_{LQD}(\cdot, \cdot)$ function we obtain the set of k nearest neighbors to a given example z. This set will be denoted by $KLQD$. From this set we perform the imputation of missing values in z or its classification. To impute attributes that are not considered the class attribute, as the class of z can be given with a low quality value defined by a crisp/fuzzy subset, we must define how that class value can be considered in the imputation.

In the case of classical KNN method from crisp data, the known class of z causes that in the imputation the only neighbors considered are those belong to that class. Now, in the $\mathrm{KNN_{LQD}}$ method, the contribution of each neighbor to the imputation is weighted in proportion to the similarity of their class to the class of z. The contribution of each neighbor $x \in KLQD$ is defined in $\mathrm{KNN_{LQD}}$ method by $p_1(x) = 1 - f_2(x_n, z_n)$, where $f_2(\cdot, \cdot)$ is a distance function between nominal attributes and it will be defined later. In this way, if $f_2(x_n, z_n) = 0$, the classes are equal and $p_1(x) = 1$ so the example x contributes with maximum weight in the imputation of values of z. However, if $f_2(x_n, z_n) = 1$ then $p_1(x) = 0$ and the example x is not considered in the imputation.

Furthermore, as $KLQD$ examples can be defined by low quality values, the method incorporates in the imputation/classification a second weighting of each neighbor based on the imperfection of each one, so that the more low quality a neighbor is, the less influence it has in the imputation/classification (this value is defined by $p_2(x)$). Fuzziness measures or fuzzy entropy are used as a measure of imperfection of an example. Such measures characterize the sharpness of the membership functions, [6], in order to obtain global measures of the indefiniteness described by fuzzy sets, [5]. Thus, the weighting of a example x, $p_2(x)$, is defined as $p_2(x) = 1 - entropy(x)$ and the definition of $entropy(x)$ will be discussed below.

To indicate the degree with which each of the above weights ($p_1(x)$, $p_2(x)$) are applied, parameters $0 \leq F_1 \leq 1$ and $0 \leq F_2 \leq 1$ are used, fulfilling that $F_1 + F_2 = 1$. When the method is used to classify, automatically these values are set as $F_1 = 0$ and $F_2 = 1$ since as it is not known class of examples of input, it can not be used as weighting the distance of classes of the nearest neighbors. In this case, the contribution of each neighbor, in the classification of a new example, will be proportional to their "perfection degree".

The $\mathrm{KNN_{LQD}}$ method allows us to define a similarity mean value to reach, defined by the threshold $0 \leq U_{sim} \leq 1$, to carry out the imputation in an example. Thus, if the average similarity between classes of the nearest neighbors and the example z is below this threshold, the imputation is not performed. This threshold can be interpreted as the minimum average grade of belief with which the user wants to perform imputations. When U_{sim} is defined as 0 all imputations will be performed regardless of belief that supports them, and when U_{sim} is 1, it is only carried out a imputation when it is based on examples with the same class value as the imputed example. In this last case, the method works as the classical method.

Likewise, the $\mathrm{KNN_{LQD}}$ method allows us to define a degree of mean maximum entropy (U_{ent}) between the nearest neighbors to perform the imputation/classification. Thus, if the imperfection degree of the nearest neighbors were very high, we could indicate that the imputation/classification was not carried out. When $U_{ent} = 0$ the imputation/classification is only carried out when the neighbors are defined by crisp values and when $U_{ent} = 1$ all imputations are carried out regardless of the quality of the neighbors.

The process of the $\mathrm{KNN_{LQD}}$ method is described in Algorithm 1.

2.1. Fuzzy distance measures

An important aspect in the definition of $\mathrm{KNN_{LQD}}$ method is the distance function, $d_{LQD}(\cdot,\cdot)$, between two examples x and y. We define the distance $d_{LQD}(x, y)$ as a heterogeneous distance function that can work with numerical and nominal attributes that could be defined by low quality values. The $d_{LQD}(\cdot,\cdot)$ function is defined as follows:

$$d_{LQD}(x, y) = \sqrt{\frac{\sum_{i=1}^{n-1} f(x_i, y_i)^2}{n-1}}$$

where function $f(\cdot,\cdot)$ is a heterogeneous measure of the distance between two attributes that could be defined by low quality values.

Therefore, function $f(\cdot,\cdot)$ is, in a general way, defined as:

$$f(x_i, y_i) = \begin{cases} f_1(x_i, y_i) & \text{if i is numerical} \\ \\ f_2(x_i, y_i) & \text{if i is nominal} \end{cases}$$

Among possible functions to calculate the distance between low quality values expressed by generalized trapezoidal functions as described above in the case of numerical attributes, we have considered the functions in Section 2.1.1. Therefore, any one of these functions could be part of the definition of the function f_1.

Among possible functions to calculate the distance between low quality values expressed by crisp/fuzzy subsets in the case of nominal attributes, we have considered the functions in Section 2.1.2. Therefore, any one of these functions could be part of the definition of the function f_2.

As an additional characteristic, we are interested in functions that are computationally efficient given the large number of distances to be calculated in this kind of methods.

2.1.1. Numerical attributes

Next, we describe some distance measures, which are used for expressing the distance of trapezoidal shaped fuzzy sets:

- The distance of Diamond [4] (denoted by NU_1) is defined between the numerical values of the i-th attribute of two examples x and y as:

$$NU_1(x_i, y_i) = \frac{\sqrt{\frac{(a-a')^2+(b-b')^2+(c-c')^2+(d-d')^2}{4}}}{max_i - min_i}$$

  where x_i and y_i are numerical values defined by quadruples (a,b,c,d) and (a',b',c',d') respectively and max_i, min_i are the maximum and minimum values of attribute i in the dataset.

- The distance of Hausdorff generalized to fuzzy sets by Ralescu and Ralescu [14] (NU_2) between the numerical values of two examples x and y is defined by:

$$NU_2(x_i, y_i) = \sum_{\alpha=0}^{1} max\{|x_{i,1} - y_{i,1}|, |x_{i,2} - y_{i,2}|\}$$

Algorithm 1 – KNN$_{\text{LQD}}$ - K Nearest neighbor applied to low quality data

```
Introduce $z$;
Introduce $k$, $1 \leq k \leq |E|$;
Introduce $F_1$, $F_2$, $U_{sim}$, $U_{ent}$;
Obtain the set $KLQD$ with the $k$ nearest examples to $z$ according to distance function $d_{LQD}(z,x)$;
if the class of $z$ is missing then
    $F_1 = 0$; $F_2 = 1$;
else
    for all $x$ in $KLQD$ do   $p_1(x) = 1 - f_2(x_n, z_n)$   end for
    $P1_{KLQD} = \sum_{x \in KLQD} p_1(x)$
end if
for all $x$ in $KLQD$ do   $p_2(x) = 1 - \mathit{entropy}(x)$   end for
$P2_{KLQD} = \sum_{x \in KLQD} p_2(x)$
$ENTROP_{av} = \frac{\sum_{x \in KLQD} entropy(x)}{k}$
$SIM_{av} = \frac{P1_{KLQD}}{k}$
if $(((SIM_{av} \geq U_{sim}$ and $P1_{KLQD} \neq 0)$ or $F_1 = 0)$ and $(ENTROP_{av} \leq U_{ent}$ and $P2_{KLQD} \neq 0)$ or $F_2 = 0))$
then
    for all $z_j$ missing in $z$ do
        if $z_j$ is numerical then
            $z_j = F_1 \cdot \frac{\sum_{x \in KLQD} p_1(x) \cdot x_j}{P1_{KLQD}} + F_2 \cdot \frac{\sum_{x \in KLQD} p_2(x) \cdot x_j}{P2_{KLQD}}$;
        else if $z_j$ is nominal then
            $z_j = \left\{ \frac{F_1 \cdot \sum_{x \in KLQD} p_1(x) \cdot \mu_{x_j}(h) + F_2 \cdot \sum_{x \in KLQD} p_2(x) \cdot \mu_{x_j}(h)}{k} / h \right\}$; $\forall h \in \Omega_j$.
        end if
    end for
else
    Imputation/classification is not realized;
end if
```

where $[x_{i,1}, x_{i,2}]$ and $[y_{i,1}, y_{i,2}]$ are the intervals defined by the α-cuts of x_i and y_i.

- The Dissemblance index to fuzzy sets defined by Kaufman and Gupta [13] (NU_3) that it is defined as:

$$NU_3(x_i, y_i) = \sum_{\alpha=0}^{1} \frac{(|x_{i,1} - y_{i,1}| + |x_{i,2} - y_{i,2}|)}{2(\beta_2 - \beta_1)}$$

where $[x_{i,1}, x_{i,2}]$ and $[y_{i,1}, y_{i,2}]$ are the intervals defined by the α-cuts of x_i and y_i, and β_1 and β_2 are the bound values of the interval covering the last two intervals.

2.1.2. Nominal Attributes

Now, we describe some distance measures for nominal attributes:

- The dissimilarity measure proposed by Dubois and Prade [7] (NO_1) is defined between two nominal values of two examples x and y as:

$$NO_1(x_i, y_i) = 1 - \frac{Card(x_i \bigcap y_i)}{Card(x_i \bigcup y_i)}$$

where x_i and y_i are fuzzy sets and $Card(x_i \bigcap y_i)$ and $Card(x_i \bigcup y_i)$ are defined as the cardinality of fuzzy sets resulting from the union and intersection of x_i and y_i respectively.

- The family of distance measures that generalizing the geometric distance models to fuzzy subsets [19] is defined as:

$$d_r(x_i, y_i) = \left(\sum_{h \in \Omega_i} |\mu_{x_i}(h) - \mu_{y_i}(h)|^r \right)^{1/r} \quad ; \; r \geq 1$$

The cases $r = 1$ (NO_2) and $r = 2$ (NO_3) were studied by Kaufmann [12]. Case d_2^2 (NO_4) was studied by Kacprzyk [11].

- Disconsistency Measure [18] between values x_i

and y_i (NO_5) is defined as:

$$NO_5(x_i, y_i) = 1 - \sup_{h \in \Omega_i} (\mu_{x_i \cap y_i}(h))$$

where $x_i \cap y_i$ is the min t-norm and $\mu_{x_i \cap y_i}(h) = min\{\mu_{x_i}(h), \mu_{y_i}(h)\}; \forall h \in \Omega_i$.

2.2. Fuzzy entropy functions

Another important aspect in the definition of KNN$_{\text{LQD}}$ method is the fuzzy entropy function, $entropy(x)$, of an example x. The $entropy(x)$ function is defined as follows:

$$entropy(x) = \frac{\sum_{i=1}^{n} g(x_i)}{n}$$

where function $g(\cdot)$ is a proper fuzzy entropy function for each attribute type x_i.

Among possible functions $g(\cdot)$ of fuzzy entropy we consider the following:

- Function of Termini and Luca [5] (ENT_1) that defines the entropy of a fuzzy set x_i as:

$$ENT_1(x_i) = -\sum_{h \in \Omega_i} (\; \mu_{x_i}(h) \cdot \log(\mu_{x_i}(h)) + \\ + (1 - \mu_{x_i}(h)) \cdot \log(1 - \mu_{x_i}(h)) \;)$$

- Function of Kaufmann based on the Hamming distance [12] (ENT_2):

$$ENT_2(x_i) = \frac{2}{|\Omega_i|} \sum_{h \in \Omega_i} |\mu_{x_i}(h) - \mu_{\frac{1}{2}, x_i}(h)|$$

where

$$\mu_{\frac{1}{2}, x_i}(h) = \left\{ \begin{array}{ll} 0 & \text{if } \mu_{x_i}(h) < \frac{1}{2} \\ 1 & \text{if } \mu_{x_i}(h) \geq \frac{1}{2} \end{array} \right.$$

- Funtion of Kaufmann based on the Euclidea distance [12] (ENT_3):

$$ENT_3(x_i) = \frac{2}{|\Omega_i|^{\frac{1}{2}}} \left\{ \sum_{h \in \Omega_i} (\mu_{x_i}(h) - \mu_{\frac{1}{2}, x_i}(h))^2 \right\}^{\frac{1}{2}}$$

2.3. Classification Accuracy

As we have commented above, KNN$_{\text{LQD}}$ method can provide low quality values as output in classification. To obtain the results of classification accuracy of KNN$_{\text{LQD}}$ method, we apply the decision process shown in Algorithm 2.

Since the class value assigned to an example z by the method can be a fuzzy set $\{\mu_{z_n}(\omega_1)/\omega_1, \mu_{z_n}(\omega_2)/\omega_2, ..., \mu_{z_n}(\omega_I)/\omega_I\}$, this value is transformed to a crisp set $\{\omega_1, ..., \omega_j\}$ as follows.

Algorithm 2 – Decision in classification

```
success=0;
error=0;
success_error;
for all $z$ in $E_{test}$ do
  if ($class(z) = class_{KNN_{LQD}}(z)$) then
    success=success+1;
    else
      if ($class(z) \bigcap class_{KNN_{LQD}}(z) \neq \emptyset$) then
        success_error=success_error+1;
      else
        error=error+1;
      end if
  end if
end for
```

Let ω_m be the class with the highest membership degree in the previous fuzzy set, the crisp set is obtained as:

$$class_{KNN_{LQD}}(z) =$$

$$= \{\omega_m\} \bigcup \left\{ \omega_c \mid \frac{\mu_{z_n}(\omega_m) - \mu_{z_n}(\omega_c)}{\mu_{z_n}(\omega_m)} \leq \gamma \right\}$$

where γ is an external parameter to indicate how close to the majority class of the output set, ω_m, should be a class ω_c to be considered a final class. With $\gamma = 0$ the final class tends to be a set composed of a single class, the majority class, but it can be a set composed by more than one class in case of a tie between majority classes. With $\gamma = 1$, the final class is the one obtained by the algorithm.

Therefore, to obtain the results of classification accuracy, we apply the decision process shown in Algorithm 2.

From Algorithm 2, the interval $[min, max]$ of classification accuracy is constructed, where min is calculated as the percentage of success considering only the variable $success$ and max is calculated as the percentage of success considering the sum of the variables $success + success\_error$. We can interpret the lower bound of the interval as a pessimistic percentage of accuracy and the upper bound as an optimistic percentage of classification accuracy considering those cases where the real class is contained in the inferred class as success.

3. Experimental Results

In this section we evaluate the performance of the proposed approach when it is applied to datasets without/with low quality values.

The datasets without low quality values used to test the proposed approach have been obtained of [2] and are summarized in Table 1, where to each dataset is shown the abbreviation (Abbr), the number of examples ($|E|$), the number of numerical attributes (Nu), nominal attributes (No), the number of classes (I) and if it contains missing values (M).

We have included explicitly low quality values in these datasets using the NIPip tool to management of low quality datasets [3]:

Datasets	Abbr	|E|	Nu	No	I	M
Australian	**AUS**	690	6	8	2	N
Credit Screen	**CRX**	690	6	9	2	N
Glass	**GLA**	214	9	0	6	N
Hepatitis	**HEP**	155	6	13	2	Y
Horse-colic	**HOR**	368	7	15	2	Y
Zoo	**ZOO**	101	1	16	7	N

Table 1: Datasets description

- 10%-20% of interval values,
- 10%-20% of fuzzy values,
- 10%-20% of crisp subsets,
- 10%-20% of fuzzy subsets, and
- 10%-20% in total divided between all the above types of low quality values.

These percentages do not affect the class attribute. In addition, some of these datasets contain missing values as it is shown in Table 1.

The Table 2 shows the interval of average classification accuracy percentage for a 10-fold cross-validation to datasets without low quality data. We use several distance measures to obtain the nearest neighbors:

- $d_{LQD}(x, y) = DM_1(x, y)$ is defined by:

  $$f_1(x, y) = NU_1(x, y) \text{ and } f_2(x, y) = NO_1(x, y)$$

- $d_{LQD}(x, y) = DM_2(x, y)$ is defined by:

  $$f_1(x, y) = NU_2(x, y) \text{ and } f_2(x, y) = NO_1(x, y)$$

- $d_{LQD}(x, y) = DM_3(x, y)$ is defined by:

  $$f_1(x, y) = NU_3(x, y) \text{ and } f_2(x, y) = NO_1(x, y)$$

- $d_{LQD}(x, y) = DM_4(x, y)$ is defined by:

  $$f_1(x, y) = NU_1(x, y) \text{ and } f_2(x, y) = NO_5(x, y)$$

- $d_{LQD}(x, y) = DM_5(x, y)$ is defined by:

  $$f_1(x, y) = NU_2(x, y) \text{ and } f_2(x, y) = NO_5(x, y)$$

- $d_{LQD}(x, y) = DM_6(x, y)$ is defined by:

  $$f_1(x, y) = NU_3(x, y) \text{ and } f_2(x, y) = NO_5(x, y)$$

To each measure, two columns show the classification accuracy when several values to k, $k = 1$ and $k > 1$, are used. In the latter case $k = \sqrt{|E|}$ for all datasets except ZOO where $k = 2$ and GLA where $k = 3$ due to these datasets have few examples and a greater number of classes so that the value of $k = \sqrt{|E|}$ is too large. Others parameters are defined in the experiment as follows:

- $\gamma = 0$;
- $F_1 = 0$, $F_2 = 1$;
- $U_{ent} = 0.2$;
- $g(x_i) = ENT_2(x_i)$ to define $entropy(x)$;

.

It is not necessary to define U_{sim}, since we perform classification and in this case, as discussed in Section 2, no weighting of similarity between classes is used. In case $k = 1$, the obtained class by the KNN_{LQD} method is a crisp value, so the interval extremes generated by the decision rule 2 match. When $k > 1$, the obtained class tends to be imprecise generating an accuracy interval. This interval, given that $\gamma = 0$, appears when the class obtained by the KNN_{LQD} method has several majority classes with the same degree of membership. For this reason, and according to the decision rule of Algorithm 2, columns corresponding to $k > 1$ of Table 2 show, on the one hand, a crisp accuracy when in the class inferred by KNN_{LQD} there is a majority class and on the other hand an interval accuracy when the class inferred by the KNN_{LQD} method has several majority classes with the same membership degree. With the value $\gamma = 0$, the class inferred by the method is less imprecise and a clearer comparison of the results obtained $k = 1$ and $k > 1$ can be carried out. On the other hand, experiments performed previously, show that with the entropy function $ENT_2(\cdot)$ with $U_{ent} = 0.2$ a good performance of the method is obtained, since the imperfection percentage in the data are not so high.

The accuracy results are expressed by intervals or by crisp values depending on the class inferred. To compare how good the accuracy results are the *hl* measure has been defined. This measure is directly proportional to the mean value of the interval and inversely proportional to the width of the interval. That is, we consider that the higher accuracy and the less imprecise the interval is, the better a result is. Therefore, *hl* measure is expressed as:

$$hl = \left(\frac{max + min}{2}\right) - (max - min)$$

where *min* and *max* are the extremes of intervals.

In Table 2 the best results for each dataset are indicated in bold. We use the distance *hl* when results are obtained in the form of intervals. In the last row the mean value of each column is shown. We can see that the best result is obtained with the distance defined by DM_4 and $k = 1$.

The results are competitive with those obtained in [15] where they carry out the classification of a set of datasets using the KNN method with $k = 1$ and using several heterogeneous distance measures. The only low quality values that contain datasets of [15] are missing values.

Next, with datasets of Table 1 with low quality values we obtain some results using the four best distance measures from the Table 2.

The Table 3 shows the average classification accuracy percentage for a 10-fold cross-validation to datasets with low quality data. The parameters used in the experiment are the following:

	DM_1		**D**M_2		**D**M_3		**D**M_4		**D**M_5		**D**M_6	
Datasets	**k** $= 1$	**k** > 1	**k** $= 1$	**k** > 1	**k** $= 1$	**k** > 1	**k** $= 1$	**k** > 1	**k** $= 1$	**k** > 1	**k** $= 1$	**k** > 1
AUS	81.4	**86.4**	65.4	68.3	74.6	82.9	81.4	**86.4**	65.4	68.3	74.6	82.9
CRX	81.4	**85.9**	63.9	67.4	68.6	78.8	81.2	**85.9**	63.9	67.4	68.4	79.7
GLA	71.9	[67.2,73.7]	**72.9**	[67.7,72.8]	53.1	[41.1,57.9]	71.9	[67.2,73.7]	**72.9**	[67.7,72.8]	53.1	[41.1,57.9]
HEP	**82.8**	[81.0,88.6]	64.2	[77.8,80.3]	68.3	79.7	81.6	[77.8,86.0]	64.2	[77.8,80.3]	73.0	79.7
HOR	76.9	**[81.3,83.4]**	58.2	[69.2,73.3]	51.4	[60.3,65.4]	79.6	[77.7,84.2]	58.2	[69.2,73.3]	50.8	[60.8,64.1]
ZOO	**97.0**	[91.2,97.0]	94.0	[90.1,95.0]	94.0	[90.1,95.0]	**97.0**	[91.2,97.0]	94.0	[90.1,95.0]	94.0	[90.1,95.0]
Average	81.85	80.3	69.8	72.0	68.3	69.9	**82.1**	78.8	69.8	72.0	69.0	70.3

Table 2: Results with datasets of Table 1 without low quality data

- $F_1 = 0$, $F_2 = 1$;
- $k = \sqrt{|E|}$;
- $U_{ent} = 0.2$;
- $g(x_i) = ENT_2(x_i)$ to define $entropy(x)$;
- $\gamma = 0.05$.

In Table 3 the best results for each dataset are indicated in bold. In the last column the best distance measure to each dataset is indicated. In the last row of the table, the best distance measure to the experiments with 10% and 20% of low quality values is indicated. Again, to measure the goodness of results expressed by intervals, we use the measure *hl*. The results of dataset GLA corresponding a Subset and Fuzzy Subset are not obtained because the dataset does not contain nominal attributes.

In general, distances $DM_2(\cdot,\cdot)$ and $DM_5(\cdot,\cdot)$ obtain bad results except for the GLA dataset which has the characteristic of being formed by numerical attributes. In this case, $DM_2(\cdot,\cdot)$ and $DM_5(\cdot,\cdot)$ are working in the same way. Therefore, the good performance is produced by numerical distance $NU_2(\cdot,\cdot)$ common to both distances.

In the remaining datasets where numeric and nominal attributes appear, the distances with best performance are $DM_1(\cdot,\cdot)$ and $DM_4(\cdot,\cdot)$, although overall $DM_1(\cdot,\cdot)$ has better performance. Therefore, the numerical distance $NU_1(\cdot,\cdot)$ and the nominal distance $NO_1(\cdot,\cdot)$ have the best performance.

According to *hl* measure, we can establish the ranking displayed in Table 4.

	Ranking	**hl**
1	$d_{LQD}(x,y) = DM_1(\cdot,\cdot)$	78.9
2	$d_{LQD}(x,y) = DM_4(\cdot,\cdot)$	78.2
3	$d_{LQD}(x,y) = DM_5(\cdot,\cdot)$	67.9
4	$d_{LQD}(x,y) = DM_2(\cdot,\cdot)$	67.8

Table 4: Ranking of the distance measures with low quality data using *hl* measure

Comparing the results obtained in the datasets with and without low quality values, we can conclude that the method has a robust performance, maintaining accurate results, somewhat smaller but very similar, when adding different types of low quality values.

4. Conclusions

In this paper we have presented an extension of classical KNN method so it can work with low quality values, leading to the KNN_{LQD} method. Thus we have a method to impute/classify datasets with missing, fuzzy and interval values in numerical attributes; and crisp/fuzzy subsets in nominal attributes. Experiments have been focused on the classification task using different distance measures. The results show that the method is competitive with the classic method when working with datasets without low quality values and is robust, keeping similar results when working with datasets with low quality values. As future works a comparison of the proposed method with other methods of literature that works with low quality data will be carried out. It is also necessary to carry out the extension of examples reduction methods to withstand treatment of low quality data to improve the efficiency of the method. Finally, a more detailed analysis of the different parameters of the method must be performed.

Acknowledgements - Supported by the project TIN2011-27696-C02-02 of the Ministry of Economy and Competitiveness of Spain.

References

[1] H. Ahn, H. Moon, J. Fazzari, N. Lim, J. Chen, R. Kodell, Classification by ensembles from random partitions of high dimensional data, *Computational Statistics & Data Analysis*, 51:6166–6179, 2007.

[2] A. Asuncion and D.J. Newman, *UCI Machine Learning Repository*, University of California, School of Information and Computer Science, http://www.ics.uci.edu/∼mlearn, 2007.

[3] J.M. Cadenas, M.C. Garrido and R. Martínez-España, *Software tool: NIP tool*, Universidad de Murcia, http://heurimind.inf.um.es, 2012.

[4] P. Diamon and P.Kloeden, Metric Spaces of Fuzzy Sets: Theory and Application, *World Scientific*, Singapore, 1994.

[5] A. DeLuca and Termini, A definition of a nonprobabilistic entropy in the setting of fuzzy sets theory, *Information and Control*, 301–312, 1972.

		10%				**20%**				
		DM_1	**D**M_2	**D**M_4	**D**M_5	**D**M_1	**D**M_2	**D**M_4	**D**M_5	
Interval	**AUS**	**86.2**	67.1	**86.2**	67.1	85.8	67.5	85.8	67.5	DM_1, DM_4
	CRX	85.7	67.0	**85.9**	67.0	85.7	67.0	85.7	67.0	DM_4
	GLA	[58.4,66.8]	[58.9,63.1]	[58.4,66.8]	[58.9,63.1]	**[61.7,66.9]**	[59.9,63.2]	**[61.7,66.9]**	[59.9,63.2]	DM_2, DM_5
	HEP	**[81.9,86.4]**	[77.3,79.3]	[81.3,85.8]	[77.3,79.3]	[81.2,84.5]	[78.0,79.3]	[80.6,83.8]	[78.0,79.3]	DM_1
	HOR	[81.3,84.5]	[69.1,73.4]	[78.9,85.1]	[69.1,73.4]	**[82.1,84.8]**	[68.8,73.4]	[78.9,84.6]	[68.8,73.4]	DM_1
	ZOO	**[89.2,92.2]**	[73.5,76.4]	**[89.2,92.2]**	[73.5,76.4]	[89.2,93.2]	[68.4,74.2]	**[89.2,93.2]**	[68.4,74.2]	DM_4
Fuzzy	**AUS**	86.2	67.1	86.2	67.1	**87.2**	66.7	**87.2**	66.7	DM_1, DM_4
	CRX	85.8	65.5	**86.1**	65.5	85.5	65.8	85.9	65.8	DM_4
	GLA	[58.4,63.5]	**[59.4,62.6]**	[58.4,63.5]	**[59.4,62.6]**	[57.0,63.1]	[59.8,64.6]	[57.0,63.1]	[59.8,64.6]	DM_2, DM_5
	HEP	**[81.3,85.8]**	[78.0,79.3]	[80.0,86.4]	[78.0,79.3]	[80.6,85.1]	[78.0,79.3]	[79.3,85.8]	[78.0,79.3]	DM_1
	HOR	[81.5,85.1]	[66.9,74.2]	[77.2,84.3]	[66.9,74.2]	**[81.5,84.8]**	[66.9,72.3]	[77.2,83.3]	[66.9,72.3]	DM_1
	ZOO	**[89.2,92.2]**	[73.5,76.4]	**[89.2,92.2]**	[73.5,76.4]	[89.2,93.2]	[68.4,74.2]	[89.2,93.2]	[68.4,74.2]	DM_1, DM_4
Subset	**AUS**	84.9	68.3	**85.2**	68.2	82.6	68.3	83.3	68.3	DM_4
	CRX	**84.5**	67.4	83.6	67.4	83.8	67.4	82.5	67.4	DM_1
	GLA	-	-	-	-	-	-	-	-	-
	HEP	[81.2,83.8]	[76.0,79.9]	[80.6,86.5]	[76.1,79.9]	**[83.2,87.1]**	[76.1,79.9]	[83.2,87.2]	[76.1,79.9]	DM_1
	HOR	**[75.9,80.5]**	[68.3,74.0]	[75.0,83.2]	[68.3,74.0]	[75.0,81.3]	[68.3,74.0]	[72.6,82.4]	[68.3,74.0]	DM_1
	ZOO	[86.2,91.2]	[76.4,77.4]	**[88.2,92.2]**	[76.4,77.4]	[70.4,78.4]	[62.5,73.4]	[72.4,79.4]	[67.5,71.4]	DM_4
Fuzzy subset	**AUS**	84.9	68.3	**85.8**	68.3	81.9	68.3	82.9	68.3	DM_4
	CRX	**84.3**	67.4	83.5	67.4	83.3	67.4	81.7	67.4	DM_1
	GLA	-	-	-	-	-	-	-	-	
	HEP	[81.2,86.4]	[76.0,79.9]	[81.9,86.5]	[76.0,79.9]	[79.2,83.8]	[76.1,79.9]	**[83.2,87.8]**	[76.1,79.9]	DM_4
	HOR	**[76.1,79.4]**	[68.3,74.0]	[76.4,82.4]	[68.3,74.0]	[73.7,80.5]	[68.3,74.0]	[75.4,82.1]	[68.3,74.0]	DM_4
	ZOO	[87.2,90.2]	[76.4,77.4]	**[89.2,93.2]**	[76.4,77.4]	[86.2,91.2]	[75.4,78.4]	[84.3,92.2]	[75.4,79.3]	DM_1
All	**AUS**	**84.6**	67.1	84.1	67.1	80.7	67.1	82.6	67.1	DM_4
	CRX	**84.2**	66.5	82.9	66.5	83.5	64.8	82.0	64.8	DM_1
	GLA	[59.4,65.9]	[59.8,63.1]	[59.4,65.9]	[59.8,63.1]	[58.9,65.0]	**[60.4,63.6]**	[58.9,65.0]	**[60.4,63.6]**	DM_2, DM_5
	HEP	**[82.0,85.9]**	[77.3,79.3]	[80.6,83.3]	[77.3,79.3]	[80.6,83.2]	[78.0,79.3]	[79.9,85.1]	[78.0,79.3]	DM_1
	HOR	**[79.1,81.3]**	[68.3,73.2]	[75.2,82.9]	[68.3,73.2]	[74.2,79.6]	[66.9,72.6]	[73.4,80.5]	[66.9,72.6]	DM_1
	ZOO	[87.2,90.2]	[73.5,75.4]	**[88.2,91.2]**	[73.5,75.4]	[86.3,91.2]	[67.5,71.4]	[86.3,91.2]	[68.4,72.3]	DM_4
		DM_1				DM_1				

Table 3: Results with datasets of Table 1 with low quality data

[6] J. Dombi and L. Porkolab, Measures of fuzziness, *Annales Universitasis Scientiarium Budapestinensis, Sectio Computatorica*, 12:69–78, 1991.

[7] D. Dubois and H. Prade, *Fuzzy Sets and System, Theory and Applications*, Academic Press, 1980.

[8] R. O. Duda, P. E. Hart and D.G. Stork, *Pattern Classification*, John Wiley and Sons, New York, 2001.

[9] A. Fernández, M.J. del Jesus, F. Herrera, Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets, *International Journal of Approximate Reasoning*, 50(3):561–577, 2009.

[10] S. García, J. Derrac, J.R. Cano and F. Herrera, Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study, *IEEE Transactions on pattern analysis and machine intelligence*, 34(3):417–435, 2012.

[11] J. Kacprzyk, Fuzzy set-theoretic approach to the optimal assignment of work places, *Large-Scale Systems Theory and Applications*, Italy, 123-131, 1976.

[12] A. Kaufmann, *Introduction to the Theory of Fuzzy Subsets vol. 1*, Academic Press, 1975.

[13] A. Kaufmann and M.M. Gupta, Introduction to Fuzzy Arithmetic, *Van Nostrand Reinhold*, New York, 1985.

[14] A.L. Ralescu and D.A. Ralescu, Probability and fuzziness, *Information Science*, 34:85–92, 1984.

[15] D.R. Wilson and T.R. Martinez, Improved Heterogeneous Distance Functions, *Journal of Artificial Intelligence Research*, 6:1–34, 1997.

[16] L. Sánchez , M.R. Suárez, J.R. Villar, I. Couso, Mutual information-based feature selection and partition design in fuzzy rulebased classifiers from vague data, *International Journal of Approximate Reasoning*, 49(3):607–622, 2008.

[17] L. Sánchez, I. Couso, J. Casillas, Genetic learning of fuzzy rules based on low quality data, *Fuzzy Sets and Systems*, 160:2524–2552, 2009.

[18] C.J. Zsolt and S. Kovàcs, Distance based similarity measures of fuzzy sets, *Proceedings of 3rd Symposium on Applied Machine Intelligence* (SAMI 2005), Slovakia, 2005.

[19] R. Zwick, E. Carlstein and D.V. Budescu, Measures of Similarity Among Fuzzy Concepts: A Comparative Analysis, *International Journal of Approximate Reasoning*, 1:221–242, 1987.