Journal of Ambient Intelligence and Smart Environments 1 (2017) 1–12
IOS Press

A k-Nearest Neighbors based Approach applied to more realistic Activity Recognition datasets

Jose M. Cadenas [a,*], M. Carmen Garrido [a], Raquel Martinez-España [b] and Andrés Muñoz [b]
[a] *Dept. of Information and Communications Engineering, University of Murcia, Murcia, Spain*
E-mails: jcadenas@um.es, carmengarrido@um.es
[b] *Dept. of Computer Engineering, Catholic University of Murcia, Murcia, Spain*
E-mails: rmartinez@ucam.edu, amunoz@ucam.edu

Abstract.
Due to the latest technological advances, the current society has the possibility to store large volumes of data in the majority of the problems of the daily life. These data are useless if there is not a set of techniques available to analyze them with the objective of obtaining knowledge that facilitates the problem resolution. This paper focuses on the techniques provided by data mining as a tool for intelligent data analysis in the field of human activity recognition, specifically in the application of two techniques of data mining capable of carrying out the extraction of knowledge from data that are not as accurate and exact as desirable. This type of data reflects the true nature of the information collected on a day-to-day basis. The proposed techniques allow us to perform a preprocessing of the data by means of an instance selection that improves the computational requirements of the system response, obtaining satisfactory accuracy results. Several experiments are carried out on a real world dataset and various datasets obtained from the previous one in a synthetic way to simulate more realistic datasets that illustrate the potential of the techniques proposed.

Keywords: Imperfect Information, Fuzzy Sets, Classification, Instance Selection, Data Mining, k-Nearest Neighbors

1. Introduction

At present, in the majority of domains that involve daily life problems, large amounts of data are available on them. This has been favored by the advancement of hardware technology and more specifically by the development of increasingly smaller sensors and with a decreasing price which allows to keep low implantation costs. These sensors are essential elements in the processes of monitoring, measurement and data collection. From these datasets, the Data Mining (DM) process allows us to obtain useful knowledge, that can be exploited in a large number of applications aimed to solve the problems of daily life. Thus we can find applications such as planning and managing forests, text categorization, climate prediction, network intrusion detection systems, recognition of human activity which is used in many innovative applications aimed at improving comfort, energy saving, security, elderly care, etc.

There are many DM techniques used in these applications but few of them take into account that the data provided by the sensors or other means may not be as accurate as it would be desirable. Thus, we could have data with noisy values, missing and often ambiguous values, which we name imperfect data.

Among the DM techniques used in applications to solve problems of daily life, we highlight the k-Nearest Neighbor (kNN) technique where k are the neighbors considered to classify a new instance [20]. In this work we focus on human activity recognition problems from sensor data using as recognition process

*Corresponding author.E-mail: jcadenas@um.es.

two approaches, a kNN and an instance selection processes. The kNN approach, named kNN$_{imp}$, supports training/test datasets with values expressed with different types of imperfection (missing, interval and fuzzy values) and the process of instance selection, named ISEL$_{imp}$, also works with imperfect data and allows to reduce the number of instances.

This paper is organized as follows: in Section 2 we present a review of related works with DM applied to human activity recognition. In Section 3, we briefly describe the kNN$_{imp}$ technique that allows the processing of imperfect data by incorporating the Fuzzy Sets Theory and that we will apply to human activity recognition. In order to deal with high volumes of data, in Section 4 we propose and design an instance selection technique that allows us the selection of instances from imperfect data. In Section 5 we apply the proposed techniques to the detection of falls from a dataset based on human activity recognition focusing on information that is possibly discarded because this information can not be manipulated with conventional DM techniques. In these situations, and without discarding the imperfection of the collected information, we simulate and construct datasets containing values that reflect the true nature of them. On these datasets, we apply the proposed techniques and analyze the obtained results. Finally, in Section 6 we make a synthesis of the main contributions provided in this paper.

2. Data mining applied to human activity recognition

The general procedure to perform a DM process for activity recognition consists of the following five steps: (1) acquiring sensor data of activities, including annotations of user's actions, (2) transforming the data into application-dependent features, e.g. by computing specific properties, eliminating noise, normalizing the data or reducing its dimensionality, (3) dividing the dataset into a training and a test set, (4) training the algorithm on the training set, and (5) testing the classification performance of the trained algorithm on the test set. Commonly, steps (3) to (5) are repeated with different partitioning into training and test sets, and the results are averaged, thus providing a better estimate of the generalization capability of the algorithm [6].

There are many DM techniques applied to human activity recognition problems. Without being exhaustive, in [2] the authors carry out the recognition of 20 daily activities from data obtained by means of five accelerometers and a digital watch that a subject wears. They obtain a dataset where the subjects perform a list of activities without supervision of the researcher together with data collection under somewhat more controlled conditions. The activity recognition is performed using C4.5 decision tree obtaining the best results compared to the other techniques used as kNN. In [25], the accuracy obtained in classification is evaluated when situating the wearable sensors in different body positions. The user's activity in real time is obtained from the information provided from multiple sensors and the classification is carried out during a day. To perform the activity recognition, they use the decision tree C4.5, kNN, Naive Bayes and Bayesian Network techniques. The good balance between accuracy and computational complexity is obtained by the decision tree. In [34] fast Decision Tree classifiers are used for the recognition of physical activities and their intensities from information provided by five triaxial wireless accelerometers and a wireless heart rate monitor. In [28] two algorithms based on decision tree (custom and automatically generated) are applied to automatically classify daily activities with the aim of finding how to recognize such activities, which sensors are useful and which signal processing and algorithms are necessary.

The kNN technique is used in [22] where authors employ the simple nearest neighbor method to find the subject's current location from the current estimated displacement of the same using wearable sensors data. In [14] the detection of falls is performed in two steps, 1) data is collected from sensor devices and 2) kNN technique is used to identify the fall patterns in the data.

Several studies use neural networks as the technique to classify. In [28] a multilayer perceptron is applied as classifier to the same problem discussed earlier. In [27], an activity recognition system is described. This system collects data by means of a wireless sensor network and the inertial system embedded in the smartphone. Then this collected data are forwarded to the computational module, where after preprocessing them, a recurrent neural network is used for the activity recognition process. In [19], the authors use a neural network to recognize elderly activities and identify the occurrence of falls. Particularly, they describe an Android-based application that detects the falls applying a trained multilayer perceptron neural network, that takes data from smartphone resources such as accelerometer sensor and GPS.

Other techniques used are clustering, support vector machine (SVM) and Bayesian network. Thus, in [21] clustering algorithm is used to detect four types of activities from sensors located in the environment. In [15] a SVM is used to monitor the activity of older people, distinguishing seven different activities and using wearable sensors and sensors located in the environment. In [31], the authors develop a two-phase system to perform the recognition of activities. In the first phase they use a probabilistic-SVM, that takes feature vectors as input for activity prediction. The output of this first phase is the input of the second phase, where filtering module is used to deal with transitions and fluctuations obtained by probabilistic-SVM. Another work where the SVM technique for the recognition of activities is used is presented in [7]. In this paper the authors describe how activities are represented by appearance and motion features using in the activity recognition process a bag of visual words representation in conjunction with a SVM classifier. In [5] through a Bayesian network, the interaction of patients from video and audio data is analyzed.

Other DM approaches based on meta-classifiers where a collection of weak classifiers is combined into a single and possibly very accurate classifier have been applied to activity recognition. Thereby in [30] activity recognition is performed by meta-level classifiers using strategies based on boosting, bagging, plurality voting and stacking obtaining better results than with the base classifiers. The authors use a single triaxial accelerometer. Among the base classifiers they use decision trees, kNN, SVM and Naive Bayes. The best results are obtained with plurality voting.

In all previous applications, although some authors indicate that imprecisions/inaccuracies occur in the measures, they do not incorporate in the available data their true nature. In some works as in [33] the authors try to perform some treatment of these imprecisions/inaccuracies incorporating in the models fuzzy logic to make decisions more flexible, avoiding the limitations in the representation of the human figures and smoothing the limits in the parameter assessment. However, fuzzy logic is not used in the representation of the data.

It would be interesting to incorporate in the data their true nature due to: 1) the absolute error committed by a sensor, 2) missing values produced by errors in communication at certain times and 3) to allow the classification of the instances used in the learning would be a soft classification, indicating that an instance does not fully belong to one class but it has different membership degrees to several classes. This soft classification can be incorporated not only in the class attribute but in any other nominal attribute. This is the potential of kNN_{imp} and ISEL_{imp} techniques we describe in the following sections.

As we will see next, when we address the various phases of the general procedure of DM for activity recognition (described at the beginning of this section) using kNN_{imp} and ISEL_{imp} approaches, the phase (4) is already eliminated because kNN technique lacks a training algorithm and phase (2) requires less transformation of the data since the technique is able to work with imperfect values (missing, intervals, fuzzy values). In this phase the instance reduction of training dataset using ISEL_{imp} is carried out which allows that the computational cost of phase (5) decreases and therefore improving the response in online applications.

3. Background and Preliminaries

3.1. Background of kNN technique and imperfect data

The kNN technique, where k is the number of neighbors considered, is a non-parametric method and it infers 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). The kNN technique is one of the important techniques in top 10 DM algorithms [1]. This technique has been analyzed extensively by the research community and many approaches have been proposed, for example, the computation of similarity measures, the optimum choice of the k parameter or the definition of weighting schemes for instances and attributes. Fuzzy sets theory has been the basis of a remarkable number of these approaches. All these approaches have been proposed with a clear objective: improving the accuracy of the kNN technique (in [9] different approaches of this technique are discussed).

Although this technique has been extensively analyzed, there are few extensions of the technique regarding the handling of imperfect data. Some extensions are aimed at dealing with multivalued class instances and others allow us some kind of uncertain data to some degree, [1, 10, 36]. In [3], the authors present an extension, called kNN_{imp}, which deal with the data in a more general way and allows us to interpret the imperfection expressed in the data, obtaining robust behavior without transforming the true nature of them. In the following section we briefly describe the kNN$_{imp}$ technique.

3.2. kNN$_{imp}$: a classifier that handles imperfect data

In [3], Cadenas et al. propose and design a new kNN approach that supports different types of imperfect data. Algorithm 1 describes the process of the this approach. Let us consider a set of instances E. Each instance $\mathbf{x}$ is characterized by a number of n attributes in a vector $(x_1, x_2, \ldots, x_n)$, where the n-th attribute represents the class. The domains of each attribute, $\Omega_{x_1}, \Omega_{x_2}, \ldots, \Omega_{x_{n-1}}$, can be numerical or nominal, while the domain of the class Ω_{x_n} can take the values $\{\omega_1, \omega_2, \ldots, \omega_I\}$.

kNN$_{imp}$ technique represents numerical attributes as fuzzy sets with a trapezoidal fuzzy membership function and nominal attributes as fuzzy subsets on the h_j values of its domain $\{\mu(h_1)/h_1, \ldots, \mu(h_s)/h_s\}$.

With this representation, kNN$_{imp}$ technique can deal with values in numerical attributes of type crisp, interval, fuzzy and missing, and with values in the nominal attributes of type crisp, crisp subset, fuzzy subset and missing.

Algorithm 1: kNN$_{imp}$ technique

```
Input Dataset E, Instance to classify z, Value k;
1 ⩽ k ⩽ |E|, Values U_D and U_I (U_D, U_I ∈ [0,1])
Let KIMP_z be the set of the k nearest instances of z
according to d_imp(·,·)
Calculate imperfection weight (p(x^j)) and distance
weight (q(x^j)) for all x^j ∈ KIMP_z
if (degree of imperfection of KIMP_z) ⩽ U_I then
    Aggregate the information of each neighbor in order to
    obtain possible class values for the instance z using
    AggreN and AggreF functions
    Calculate the set of output classes z_n using U_D
    Output z_n
else
    Output Classification is not performed
end if
```

As Algorithm 1 shows, kNN$_{imp}$ technique computes the set $KIMP_z$ that contains the k instances $\mathbf{x^j} \in E$ which are the nearest to z according to the measure $d_{imp}(\mathbf{x^j}, \mathbf{z})$. Then, for each instance $\mathbf{x^j} \in KIMP_z$, two weights are calculated depending on its degree of imperfection ($p(\cdot)$) and its distance to $\mathbf{z}$ ($q(\cdot)$). Furthermore, the overall degree of imperfection in $KIMP_z$ is measured, if it is too high, the classification is not performed. To establish the maximum degree of imperfection, kNN$_{imp}$ uses the parameter U_I. This parameter plays an important role when the dataset can contain instances with high degrees of imperfection.If $KIMP_z$ passes the imperfection check, the functions $AggreN$ and $AggreF$ obtain the set of possible weighted classes taking into account the k nearest neighbors. The class with the highest score is chosen as output, together with other classes whose score is similar to the highest. To assess if a class should be included in the final output, kNN$_{imp}$ uses the threshold U_D.

We briefly comment on some of the elements of kNN$_{imp}$ (for more depth, see [3]): Distance/dissimilarity measures, contribution of neighbors to the classification, controlling the similarity in the output classes, aggregation methods for classification and the process to obtain the accuracy.

3.2.1. Distance/dissimilarity measures

In order to calculate the nearest neighbors, the technique uses a measure (distance/dissimilarity) which computes the distance between two instances and can work with/without imperfect data coming from numerical and nominal attributes.

The measure is defined as

$$d_{imp}(\mathbf{x}, \mathbf{x}') = \sqrt{\frac{\sum_{i=1}^{n-1} f(x_i, x_i')^2}{n-1}}$$

where $f(x_i, x_i')$ is defined by two functions $f_1(x_i, x_i')$ and $f_2(x_i, x_i')$. $d_{imp}(\mathbf{x}, \mathbf{x}')$ is a heterogeneous function defined from different functions, $f_1(\cdot,\cdot)$ and $f_2(\cdot,\cdot)$, on different kinds of attributes (numerical and nominal respectively) where $f_1(\cdot,\cdot)$ and $f_2(\cdot,\cdot)$ are normalized fuzzy distance or dissimilarity measures.

3.2.2. Contribution of neighbors to the classification

- Weights based on distance:

$$q(\mathbf{x}) = 1 - d_{imp}(\mathbf{x}, \mathbf{z})$$

  with $\mathbf{x} \in KIMP_z$
- Weights based on imperfection:

$$p(\mathbf{x}) = 1 - imp(\mathbf{x})$$

  with $\mathbf{x} \in KIMP_z$ and $imp(\cdot): E \to [0,1]$ defined as $imp(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} g(x_i)$ where $g(\cdot) : \Omega_{x_i} \to [0,1]$ is a function defined for each attribute x_i and it measures the imperfection of the value in the attribute x_i.

3.2.3. Controlling the similarity in the output classes

The kNN$_{imp}$ technique exploits the definition of a similarity value between possible classes, defined as $sim(\omega_M, \omega_i) = \frac{\mu(\omega_M)-\mu(\omega_i)}{\mu(\omega_M)}$, to perform the classification of an instance. The minimum $sim(\omega_M, \omega_i)$ necessary to consider that the classes ω_M and ω_i are possible outputs is controlled by the threshold $0 \leqslant U_D \leqslant 1$. Thus, let us assume that ω_c is the class having the highest membership degree $\mu(\omega_c)$ to classify an instance. If there are other classes with very close membership degrees to $\mu(\omega_c)$, we could return all these classes as possible classification of the instance. The role of U_D threshold is to define how close to ω_c must be a class to be considered an output class. Thus, the threshold U_D allows the output of kNN$_{imp}$ technique is multivalued.

3.2.4. Aggregation methods for classification

The aggregation methods defined for kNN$_{imp}$ technique are composed of the two functions $AggreN(\cdot)$ and $AggreF(\cdot)$. These two functions provide high flexibility to this technique, allowing choose them according to the classification problem. In [3], different aggregation methods are defined. Next, we describe four of them that will be used later in the experiments. Methods $\mathbf{WM_{CV}}$ and $\mathbf{WM_{SV}}$ use the function $AggreN()$=$WCVEN()$ but different versions of $AggreF()$ ($CV()$ and $SV()$, respectively). Methods $\mathbf{SM_{CV}}$ and $\mathbf{SM_{SV}}$ use the function $AggreN()$=$SVEN()$ and different versions of $AggreF()$ ($CV()$ and $SV()$, respectively).

– $SVEN()$ returns a vote of 1 to the class of x^j with the highest membership degree and 0 to the other classes:

$$SVEN(i, x_n^j, p(\mathbf{x^j}), q(\mathbf{x^j})) = \begin{cases} 1 & \text{if cond} \\ 0 & \text{otherwise} \end{cases}$$

and cond is verified when $i = \arg \max\limits_{h=1,...,I} \mu^j(\omega_h)$.

– $WCVEN()$ returns the score assigned by a neighbor (x^j) to each class value ($i = 1, ..., I$) and is defined as follows:

$$WCVEN(i, x_n^j, p(\mathbf{x^j}), q(\mathbf{x^j})) = \mu^j(\omega_i)\cdot p(\mathbf{x^j})\cdot q(\mathbf{x^j})$$

that is, the weight assigned by a neighbor to each class value is determined by the weight of that value in x_n^j ($\mu^j(\omega_i)$), by the weight of x^j according to its distance ($q(\mathbf{x^j})$) and by the weight of x^j according to its imperfection ($p(\mathbf{x^j})$).

- $CV()$ provides as output the fuzzy set $\{\mu(\omega_i)/\omega_i\}$ composed of ω_i with $\mu(\omega_i) > 0$.

$$\mu(\omega_i) = \frac{\sum_{j=1}^{k} AggreN(i, x_n^j, p(\mathbf{x^j}), q(\mathbf{x^j}))}{\sum_{j=1}^{k}\sum_{i=1}^{I} AggreN(i, x_n^j, p(\mathbf{x^j}), q(\mathbf{x^j}))}$$

- $SV()$ provides as output the crisp set $\{1/\omega_h\}$ composed of w_h where

$$h = \arg \max_{i=1,...,I} \sum_{j=1}^{k} AggreN(i, x_n^j, p(\mathbf{x^j}), q(\mathbf{x^j}))$$

The set can be composed of more than one element if there are several values of class with maximum score.

3.2.5. Obtaining the accuracy of multivalued classes

Since the class inferred by kNN$_{imp}$ technique, using the aggregation methods defined above and the threshold U_D, can be a set, it is necessary to define how the accuracy in classification is measured. Algorithm 2 shows this process where the considered dataset is a subset of reserved instances as test dataset (E_{test}).

Algorithm 2: Classification Accuracy

Input Dataset E_{test}, Class value of **z** ($class(\mathbf{z})$), Class value inferred to **z** (($class_{kNN_{imp}}(\mathbf{z})$)
Suc, $SucErr$=0;
for all z in E_{test} **do**
  if $class_{kNN_{imp}}(\mathbf{z}) = class(\mathbf{z})$ **then** $Suc = Suc + 1$
  else
    if $(class_{kNN_{imp}}(\mathbf{z}) \bigcap class(\mathbf{z})) \neq \emptyset$ **then**
      $SucErr = SucErr + 1$
end for
$Acc_{min} = \frac{Suc}{|E_{test}|}$, $Acc_{max} = \frac{Suc+SucErr}{|E_{test}|}$
Output $[Acc_{min}, Acc_{max}]$

In the definition of the upper bound of this interval we consider as success those cases where the class value of a test instance is not the same but it is included in the inferred class value. Note that situations in which the two values of interval are equal will be denoted with a single value Acc.

4. Instance Selection from Imperfect Data

In general, the kNN technique suffers from several drawbacks. Among these drawbacks it is worth noting the need for a high memory requirement to store all the examples that make up the learning set, and the low efficiency during the operation of the decision rule due to the high comparison of similarities between the test examples and the learning examples [20]. To solve these problems, many proposals have been made in the literature, among which we can find the instance selection [16]. Once the instance selection is performed, the computational cost and memory requirements of the kNN technique (and of any other that uses the dataset) are decreased because they work with a reduced dataset.

We propose to perform an instance selection whose objective is to select those instances that are in the decision boundaries and that allow us to discriminate better between the different class values Our proposal allows us to carry out the selection from imperfect data. This will allow us to improve the real-time processing requirements of some applications along with the possibility of handling data sources with greater expressive richness. The instance selection technique that we propose together with the use of kNN$_{imp}$ technique allows us to perform a more realistic DM process.

4.1. ISEL$_{imp}$: An instance selection technique from imperfect data

We propose ISEL$_{imp}$, an instance selection technique based on Condensed Nearest Neighbor (CNN) technique [17], but with the appropriate extensions focused on taking into account that the input dataset may contain imperfect values. For this, we incorporate in the process suitable distance/dissimilarity measures for imperfect data and imperfection measures, aggregation methods, contribution of neighbors in the final decision and some control parameters.

In this way, the proposed technique starts from an empty set S to which randomly selected instances are added. Next, while S is modified with the incorporation of some new instance, the technique tries to classify the instances of E using only instances of S. If an instance is incorrectly classified, it is added to the set S and is deleted from set E. In order to classify an instance $\mathbf{x}$ of E, using the set S, the technique calculates for each instance in S its weight based on the imperfection and its weight based on distance to $\mathbf{x}$, so that an instance of S with less distance and less imperfection has a greater weight. The class values of k nearest neighbors to $\mathbf{x}$ are aggregated using an aggregation method as the ones described in Section 3.2.4. If the distance from the aggregated class to the real class is greater than a threshold U_c, the instance is not classified correctly and is added to the set S. The threshold U_c defines, with a value greater than 0, how much the user allows the aggregated and real class can differ to continue considering that the classification is correct. A value of $U_c = 0$ indicates that the classification is considered correct only when the aggregated class and the real class are the same.

The instance selection technique from imperfect data proposed is shown in Algorithm 3.

Algorithm 3: ISEL$_{imp}$ technique

```
Input Dataset E
S = ∅
Let IN be a set of k instances of E
Remove IN from E and add IN to S
while S is modified do
  for all x^j ∈ E do
    Let KIMP_{x^j} be the set of the k nearest instances x^j
    in S according to d_imp(·,·)
    Calculate imperfection weight (p(y^i)) and distance
    weight (q(y^i)) for all y^i ∈ KIMP_{x^j}
    Aggregate the information of each neighbor in order
    to obtain possible class values for the instance x^j
    using AggreN and AggreF functions.
    Let classAgg be the result of that aggregation
    if f_2(x_n^j, classAgg) > U_c then
      Add x^j to S
      Remove x^j from E
    end if
  end for
end while
Output S
```

As we can see in Algorithm 3, and with the aim of allowing the processing of imperfect data, the technique must use appropriate measures taking into account the inherent information of the data. Specifically, ISEL$_{imp}$ technique must use: a) a distance/dissimilarity measure $d_{imp}(\cdot)$ in order to obtain the distance between two instances with imperfect information in both numerical and nominal attributes (using the normalized fuzzy distance or dissimilarity measures $f_1(\cdot,\cdot)$, $f_2(\cdot,\cdot)$ for numerical and nominal attributes, respectively); and b) a measure $imp(\cdot)$ that obtains the degree of imperfection that contains an instance from the imperfect values of the different attributes that compose it.

From these measures, the ISEL$_{imp}$ technique uses two weights to the instances: the weight based on imperfection $p(\cdot) = 1 - imp(\cdot)$ and the weight based on distance $q(\cdot) = 1 - d_{imp}(\cdot, \cdot)$.

5. Localization Activity Dataset

In this section, we carry out several experiments oriented to the application focused on "Detection of falls". The objective of these experiments is to show the robustness of the proposed techniques handling the true nature of the available information, the relevance of the selected information and the possibility of obtaining multivalued classes.

5.1. Detection of falls

As indicated in [18], there are about tens of millions living people over the age of 65 who fall at least one time each year. Researchers have demonstrated that the risk of hospitalization and necessary post-fall care would be greatly reduced if caregivers were notified immediately after the fall. Therefore, the high percentage of falls and the high cost of the treatment of the injuries produced, make it important to have fall detection systems. Among the systems developed for this purpose are those using wearable sensors that allow measuring physical activity in real-life environments. Fall detection systems have been developed using systems based on DM trying to differentiate falls from activities of daily living (ADLs) and, in particular, in [14] the problem has been addressed through the use of kNN technique. However, in most of these systems the only information collected in the datasets comes from sensors, discarding valuable additional information that manufacturers of the devices provide about the measurement error committed by them.

It is interesting to incorporate into the datasets the real nature of the information, indicating that the real value of the sensor is represented by an interval rather than by a crisp value. In this case, the information has a greater richness and it is needed that the used DM technique will be able to work with such intervals. In addition, the malfunction of any sensor at any given time, can generate incomplete data with some missing value. These missing values do not have to be discarded if the technique is able to deal with them. This is the potential of the kNN$_{imp}$ and ISEL$_{imp}$ techniques.

In addition, the separation between the situations of fall and ADLs does not have to be always exclusive or differentiated. It may be important in this case the use of DM techniques that provide all the information obtained in the classification process allowing the external decision maker to decide when a final class can be discarded and when not. This can be done with kNN$_{imp}$ technique using the U_D threshold. Moreover, in many problems of this application domain, datasets are obtained by the performance of subjects in certain environments (houses, hospitals, etc) and in these datasets, there are real time self-annotations by subjects indicating the activity they are performing at certain times [35]. It is interesting to allow that these annotations can reflect the imprecision and ambiguity of human being allowing to use linguistic labels, or multivalued annotations, which can also be approached with the proposed techniques.

To carry out a simulation on this application and the potential of the techniques proposed in it, we perform a set of experiments on a dataset of the application domain and various modifications of it.

5.1.1. Datasets description

We apply the proposed techniques to "Localization Data for Person Activity DataSet" [23], which we have denoted as LocalAc$_{\text{ORG}}$ dataset. This dataset contains instances of five people performing different activities. Each person wore four sensors (ankle left, ankle right, belt and chest). The dataset is composed of 164860 instances and 6 attributes. Attribute 1 indicates the person, attribute 2 the sensor, attributes 3, 4 and 5 indicate the x, y, z coordinates of the location captured by the sensor, and attribute 6 is the class of the instance. The class attribute, which indicates the activity that the person is doing, has 11 class values (walking, falling, lying down, lying, sitting down, standing up from lying, on all fours, sitting on the ground, standing up from sitting, standing up from sitting on the ground). If we use a conventional DM technique on this dataset, we are discarding the true nature of the data and the classification is a hard classification (a disjoint separation of the different class values).

Applying the techniques proposed in this paper, we can handle and process imperfect data. For experiments, we make several changes on LocalAc$_{\text{ORG}}$ dataset to generate various synthetic datasets with imperfect information:

- We modify the values of the x, y, z coordinates (attributes 3, 4 and 5, respectively) obtained by the sensors to incorporate the error made by them according to the manufacturer's information. It would be more appropriate to express the measure obtained as $value \pm errorAbsolute$. Specifically, we incorporate *value* $\pm$ 0.1 into the dataset (the new dataset is denoted by $\text{LocalAc}_{\text{I}}$ dataset).

- We include on the original dataset a 10% of missing values without considering the class attribute ($\text{LocalAc}_{\text{M}}$).
- We generate a new dataset joining the imperfect values of the two previous datasets, that is, a dataset with the x, y, z coordinates expressed by *value* $\pm$ 0.1 and a 10% of missing values ($\text{LocalAc}_{\text{IM}}$).

The description of the datasets is shown in Table 1.

Table 1
Description of "LocalAc" datasets

Dataset	|E|	Nu	No	I	Imperfect values IV% – MV%
LocalAc$_{\text{ORG}}$	164860	3	3	11	0 – 0
LocalAc$_{\text{I}}$					50 – 0
LocalAc$_{\text{M}}$					0 – 10
LocalAc$_{\text{IM}}$					44 – 10

In Table 1, we show the number of instances |E|, the number of numerical (Nu) and nominal (No) attributes and the number of possible values for the class attribute (I). In the column "Imperfect values", we show the percentage of imperfect values contained in the dataset. Column IV% shows the percentage of interval values and column MV% shows the percentage of missing values. In total, the LocalAc_{ORG} dataset does not contains imperfect data, while the $\text{LocalAc}_{\text{I}}$ dataset contains 50% of imperfect values (interval values). $\text{LocalAc}_{\text{M}}$ contains 10% of missing values and $\text{LocalAc}_{\text{IM}}$ contains 54% of imperfect values (10% of missing values and 44% of interval values).

We perform different experiments aimed at checking the robustness and flexibility provided by the proposed techniques when carrying out data processing with greater expressive richness than conventional techniques. The testing process in all experiments is done by dividing the datasets into two sets: The first one is considered as train dataset (80% of the dataset) and the second one is considered as test dataset (20% of the dataset). For these datasets, both the train and the test datasets will contain imperfect data. This process is repeated five times with different partitions. All tables in the following sections show the averaged accuracy of the five replicates.

5.2. *Analysis and configuration of parameters*

We are going to carry out an analysis of the configurable parameters of the techniques taking as base the $\text{LocalAc}_{\text{ORG}}$ dataset. With the best parameter values we will perform the following experiments. In the analysis we consider all combinations of the different distance functions of Table 2, values k={1, 3, 5, 7, 9, 11, ..., 99}, values $U_D = \{0, 0.05, 0.1, 0.15, 0.2\}$ and the aggregation methods WM_{CV}, WM_{SV}, SM_{CV} and SM_{SV}. When carrying out the different executions, the results obtained with the different combinations of the distance functions considered do not show significant differences, so in Table 3 only those obtained with the Diamond distance [11] to $f_1(\cdot,\cdot)$ and Dubois and Prade measure [12] to $f_2(\cdot,\cdot)$ are shown.

Table 2
Distance/Dissimilarity/Similarity Measures

$$f_1(\cdot,\cdot) = \begin{cases} \text{Diamond distance [11]} \\ \text{Extended Hausdorff distance [29]} \\ \text{Similarity proposed by Chen [4]} \end{cases}$$

$$f_2(\cdot,\cdot) = \begin{cases} \text{Dubois and Prade dissimilarity measure [12]} \\ \text{Disconsistency index [13]} \\ \text{Similarity proposed by Santini [32]} \end{cases}$$

In Table 3 we highlight in bold the best results obtained with each aggregation method. We can see that the best results are obtained with $k = 5$ and that the accuracy decreases as the value of k increases (greater values for k have been considered although their results are not shown due to space limitations).

As discussed earlier, the technique can provide multivalued classes as output. That is, when we have to assign an output to an instance and we have several possible major classes with very similar membership degrees, we should not choose between those classes and provide as output a multivalued class (through the parameter U_D). For example, if we have a class "*a*" with membership degree $\mu(a)$=0.45 and a class "*b*" with membership degree $\mu(b)$=0.55, with the threshold U_D=0.2 the technique will return as output {a,b} since $\frac{0.55-0.45}{0.55}$=0.18 $\leqslant$ 0.2. By allowing a set of possible values in the class attribute (controlled by the threshold value U_D), the obtained results are improved as we can see in Table 3. The upper end of the interval indicates that the output class values contain the real class value giving better results which indicates that different instances have their real classes as the second can-

Table 3

Accuracy results when evaluating different parameters with LocalAc$_{ORG}$ dataset

Method, k	**U_D=0**	**U_D=0.05**	**U_D=0.1**	**U_D=0.15**	**U_D=0.2**
WM_{CV}, k=1	77.6	77.6	77.6	77.6	77.6
WM_{CV}, k=5	**79.0**	**[77.0,81.0]**	**[77.0,81.0]**	**[77.0,81.0]**	**[77.0,81.0]**
WM_{CV}, k=7	**79.0**	[77.2,80.0]	[77.2,80.0]	[77.2,80.0]	[77.2,80.0]
WM_{CV}, k=11	78.0	[76.6,79.0]	[76.6,79.0]	[76.6,79.0]	[75.0,80.8]
WM_{CV}, k=15	77.4	[76.2,78.0]	[76.2,78.0]	[75.0,79.2]	[74.0,80.0]
WM_{CV}, k=19	76.8	[75.4,77.0]	[75.4,77.2]	[74.0,78.6]	[73.8,79.0]
WM_{SV}, k=1	77.6	77.6	77.6	77.6	77.6
WM_{SV}, k=5	**79.0**	**79.0**	**79.0**	**79.0**	**79.0**
WM_{SV}, k=7	**79.0**	**79.0**	**79.0**	**79.0**	**79.0**
WM_{SV}, k=11	78.0	78.0	78.0	78.0	78.0
WM_{SV}, k=15	77.4	77.4	77.4	77.4	77.4
WM_{SV}, k=19	76.8	76.8	76.8	76.8	76.8
SM_{CV}, k=1	77.6	77.6	77.6	77.6	77.6
SM_{CV}, k=5	**[77.0,81.0]**	**[77.0,81.0]**	**[77.0,81.0]**	**[77.0,81.0]**	**[77.0,81.0]**
SM_{CV}, k=7	[77.2,80.0]	[77.2,80.0]	[77.2,80.0]	[77.2,80.0]	[77.2,80.0]
SM_{CV}, k=11	[76.6,79.0]	[76.6,79.0]	[76.6,79.0]	[76.6,79.0]	[74.2,81.0]
SM_{CV}, k=15	[76.2,78.0]	[76.2,78.0]	[76.2,78.0]	[75.0,79.2]	[74.0,80.0]
SM_{CV}, k=19	[75.4,77.0]	[75.4,77.0]	[75.4,77.0]	[74.0,78.6]	[73.8,79.0]
SM_{SV}, k=1	77.6	77.6	77.6	77.6	77.6
SM_{SV}, k=5	**[77.0,81.0]**	**[77.0,81.0]**	**[77.0,81.0]**	**[77.0,81.0]**	**[77.0,81.0]**
SM_{SV}, k=7	[77.2,80.0]	[77.2,80.0]	[77.2,80.0]	[77.2,80.0]	[77.2,80.0]
SM_{SV}, k=11	[76.6,79.0]	[76.6,79.0]	[76.6,79.0]	[76.6,79.0]	[76.6,79.0]
SM_{SV}, k=15	[76.2,78.0]	[76.2,78.0]	[76.2,78.0]	[76.2,78.0]	[76.2,78.0]
SM_{SV}, k=19	[75.4,77.0]	[75.4,77.0]	[75.4,77.0]	[75.4,77.0]	[75.4,77.0]

didate in the prediction. Therefore, allowing a multi-valued output class helps users' decision making.

Regarding the aggregation methods we can consider that the method $\mathbf{WM_{CV}}$ is the one that produces better results because it obtains the maximum precision with $U_D = 0$ and better interval for the remaining values of U_D. To compare the different intervals obtained, we used the measure $int([a,b]) = \frac{a+b}{2} - \frac{b-a}{width_{max}}$ where $width_{max}$ is the maximum amplitude of the intervals obtained.

With the results obtained with the LocalAc$_{ORG}$ dataset in Table 4, we show the common configuration of the parameters of the techniques used for the following experiments. Parameters U_I and U_c will be specified in the different experiments.

5.3. Analyzing robustness with imperfect data

The experiments in this section are performed to verify whether the use of imperfect information maintain satisfactory results. For this, we use the kNN$_{imp}$ technique with the parameter configuration shown in Table 4.

Table 4

Parameter Configuration

Parameter	**Configuration**
$f_1(\cdot,\cdot)$	Diamond distance [11]
$f_2(\cdot,\cdot)$	Dubois and Prade measure [12]
$g(\cdot)$	Power of fuzzy sets [8]
k	5
U_D	{0, 0.2}
Aggregation Method	WM_{CV}

Table 5 shows the averaged accuracy obtained when classifying the test dataset using the corresponding train dataset, both for original dataset and for datasets with imperfection. As the combination method WM$_{CV}$ is affected by the threshold U$_D$, in this table we show the results for different values of this parameter. The threshold U_I is set to 1 because the imperfection introduced in the datasets is synthetic and affects in a controlled way to all the instances, therefore, it will always be carried out the classification of the test instances. Moreover, we indicate with subscript the value of k used.

Table 5

Accuracy results when classifying the different LocalAc datasets using kNN$_{imp}$ technique

	U$_D$=0	**U$_D$=0.2**
LocalAc$_{ORG}$	79.0_5	$[77.0,81.0]_5$
LocalAc$_M$	76.0_5	$[73.2,78.0]_5$
LocalAc$_I$	79.0_5	$[77.0,81.0]_5$
LocalAc$_{IM}$	76.0_5	$[73.2,78.0]_5$

We can verify with the obtained results that the applied technique shows robustness, maintaining good results when we add to the dataset additional information about the problem (LocalAc$_I$). The averaged accuracy obtained with LocalAc$_M$ and LocalAc$_{IM}$ decreases due to we have deleted information from the dataset by entering missing values. However, we can see that the technique is robust in the treatment of imperfect data and therefore it allows to improve the data preprocessing phase in the DM process by not having to perform a transformation and elimination of these types of imperfect data.

5.4. *Instance Selection with imperfect data*

The following experiments are performed to asses the ISEL$_{imp}$ technique verifying whether the reduction made to the datasets maintain satisfactory results.

First, we select a subset of instances in the several datasets using the proposed ISEL$_{imp}$ technique. The results obtained from the process of instance selection on the four datasets are shown in Table 6. The instance selection process has been applied to each of the five train datasets (80% of the complete dataset). Table 6 shows the averaged values obtained with the five replicates. In addition, **#InsSel** and **RedRate** indicate the number of instances in the several reduced datasets and the percentage reduction respectively.

The results have been obtained with two different configurations for ISEL$_{imp}$: the first one (denoted by ISEL$_{imp}^{1-0}$) selects an instance using the nearest neighbor ($k = 1$) and the threshold $U_c = 0$. In this way any instance whose class is not equal to that of the nearest neighbor is selected and becomes part of S. In the second configuration (denoted by ISEL$_{imp}^{5-0.3}$) an instance is selected using the k nearest neighbors ($k = 5$) and the threshold $U_c = 0.3$. Thus, an example is selected, and becomes part of S, when the class of the 5 nearest neighbors designated by the WM$_{CV}$ aggregation method differs by an amount greater than 0.3 from its class ($f_2(\cdot,\cdot) > U_c$). The configuration $k = 1$, $U_c = 0.3$ has not been considered since with $k = 1$, the output class is always formed by a value and therefore $f_2(\cdot,\cdot) \in \{0, 1\}$ and the result of the configuration is always the same for all possible values of U_c. The configuration $k = 5$, $U_c = 0$ has also been checked, showing similar results to the ISEL$_{imp}^{5-0.3}$ configuration.

Table 6

Instance selection on the several datasets

	ISEL$_{imp}^{1-0}$		**ISEL$_{imp}^{5-0.3}$**	
	#InsSel	**RedRate**	**#InsSel**	**RedRate**
LocalAc$_{ORG}$	48802.8	62.99%	86804.2	34.18%
LocalAc$_M$	52692.8	60.05%	91269.4	30.80%
LocalAc$_I$	48810.0	62.99%	86809.4	34.18%
LocalAc$_{IM}$	52674.0	60.06%	91281.0	30.79%

As we can see in Table 6, the reduction obtained by the proposed technique is very significant for both the original datasets and the ones containing imperfect data. We denote the new datasets as LocalAc$_{ORG\text{-}R}$, LocalAc$_{I\text{-}R}$, LocalAc$_{M\text{-}R}$ and LocalAc$_{IM\text{-}R}$ respectively.

When we classify the corresponding test datasets using the reduced train datasets by kNN$_{imp}$ technique, we obtain the results shown in Table 7.

Table 7

Accuracy results when classifying the different reduced LocalAc datasets using kNN$_{imp}$ technique

Dataset		**U$_D$=0**	**U$_D$=0.2**
with ISEL$_{imp}^{1-0}$	**LocalAc$_{ORG\text{-}R}$**	77.0_5	$[70.0,80.0]_5$
	LocalAc$_{M\text{-}R}$	73.2_5	$[65.6,77.0]_5$
	LocalAc$_{I\text{-}R}$	77.0_5	$[70.0,80.0]_5$
	LocalAc$_{IM\text{-}R}$	73.0_5	$[65.6,77.0]_5$
with ISEL$_{imp}^{5-0.3}$	**LocalAc$_{ORG\text{-}R}$**	79.0_5	$[76.8,81.0]_5$
	LocalAc$_{M\text{-}R}$	76.0_5	$[73.0,78.0]_5$
	LocalAc$_{I\text{-}R}$	79.0_5	$[76.8,81.0]_5$
	LocalAc$_{IM\text{-}R}$	76.0_5	$[73.0,78.0]_5$

We can verify with the obtained results that the applied techniques show robustness, maintaining the results when we perform an instance selection that allows us to improve the computational cost and memory requirements of the algorithms. As we can observe with the results obtained in Tables 6 and 7, the configuration ISEL$_{imp}^{5-0.3}$ obtains the same accuracy as the datasets without reducing although the reduction is smaller than the one carried out with ISEL$_{imp}^{1-0}$. So, if the real-time response requirements are important we will choose the configuration ISEL_{imp}^{1-0}. In this configuration, with the parameter $U_D = 0.2$, we can observe that the upper ends of the intervals are still high, so the accuracy is still good allowing high-risk situations to be detected by a supervisor.

5.5. *Comparing with other instance selection techniques*

Finally, we are going to carry out the comparison of the results obtained with the techniques proposed in this work with the results obtained in [24]. In [24] a reduction of the $\text{LocalAc}_{\text{ORG}}$ dataset is realized by means of two instance selection processes: baseline process and ReDD. Both processes internally use the instance selection methods provided by GA, DROP3 and IB3 algorithms. Additionally ReDD process uses a RD (representative data) and URD (unrepresentative data) detector based on a classification technique. In particular they obtain results with a CART decision tree, k nearest neighbor and support vector machine.

In their experiments perform five partitions of the original datasets (80% train dataset and 20% test dataset) to repeat the experiment five times showing the averaged accuracy and averaged reduction rate.

In Table 8 the best results obtained in [24] and the best ones obtained with ISEL_{imp} and kNN_{imp} techniques are shown. The two configurations of ISEL_{imp} obtain better precision than the proposals of [24] with a larger reduction in the case of ISEL_{imp}^{1-0}.

Table 8
Comparative results of instance selection in $\text{LocalAc}_{\text{ORG}}$ dataset

Method	% Reduction	% Accuracy
Baseline-IB3$_{CART}$	18.54	62.46
Baseline-DROP3$_{kNN}$	44.89	69.14
Baseline-GA$_{kNN}$	59.46	62.97
ReDD-IB3$_{CART}$	18.54	62.65
ReDD-DROP3$_{kNN}$	44.89	64.67
ReDD-GA$_{kNN}$	59.46	62.35
ISEL$_{imp}^{1-0}$	62.99	77.00
ISEL$_{imp}^{5-0.3}$	34.18	79.00

6. Conclusion and Future Works

The results obtained in the experiments performed in this work lead us to think that the kNN_{imp} technique is robust when it is applied in real world problems and it reflects in a more natural and adequate way the data available in such applications decreasing the necessary data preprocessing previous to the DM phase. The proposed ISEL_{imp} technique to instance selection reduces the initial dataset by a high percentage, maintaining the classification results both by comparing them with the results obtained with the original dataset and with respect to the datasets with imperfection. This allows to improve the computational requirements of a response in real time maintaining the accuracy of the same. The results obtained with ISEL_{imp} technique have also been compared with those obtained with other instances selection techniques of literature, obtaining better results.

As future works, more combination methods should be studied in depth with both kNN_{imp} and ISEL_{imp} techniques. Moreover, new distance/dissimilarity measures can be implemented and tested with several datasets to analyzed in depth the robustness and flexibility of the techniques proposed.

Acknowledgement

Supported by the projects TIN2014-52099-R, TIN2014-56381-REDT and TIN2016-78799-P (AEI/FEDER, UE), granted by the Ministry of Economy and Competitiveness of Spain (including ERDF support).

References

[1] R. Agrawal, K-nearest neighbor for uncertain data, *Int. J. of Computer Applications* **105 (11)** (2014), 13–16.

[2] L. Bao 2004 and S. Intille, *Activity recognition from user-annotated acceleration data*, in: Int. Conf. on Pervasive Computing, 2004, pp. 1–17.

[3] J.M. Cadenas, M.C. Garrido, R. Martínez, E. Muñoz, and P.P. Bonissone, A fuzzy k-nearest neighbor classifier to deal with imperfect data, *Soft Computing* (2017), in press, doi:10.1007/s00500-017-2567-x.

[4] S. M. Chen, New methods for subjective mental workload assesment and fuzzy risk analysis, *Cybernetic and Systems* **27** (1996), 449–472.

[5] D. Chen, R. Malkin and J. Yang, *Multimodal Detection of Human Interaction Events in a Nursing Home Environment*, in: Proc. of the 6th Int. Conference on Multimodal Interfaces, 2004, pp. 82–89.

[6] L. Chen and C. Nugent, Ontology-based activity recognition in intelligent pervasive environments, *Int. J. of Web Information Systems* **5(4)** (2009), 410–430.

[7] S.L. Chua, S. Marsland and H. Guesgen, A supervised learning approach for behaviour recognition in smart homes, *J. of Ambient Intelligence and Smart Environments* **8(3)** (2016), 259–271.

[8] A. De Luca and S. Termini, A definition of a nonprobabilistic entropy in the setting of fuzzy sets theory, *Information and control* **20 (4)** (1972), 301–312.

[9] J. Derrac, S. García, and F. Herrera, Fuzzy nearest neighbor algorithms: Taxonomy, experimental analysis and prospects, *Information Sciences* **260** (2014), 98–119.

[10] S. Destercke, A k-nearest neighbours method based on imprecise probabilities, *Soft Computing* **16 (5)** (2012), 833–844.

[11] P. Diamond and P. Kloeden, *Metric spaces of fuzzy sets: theory and applications*, World scientific, 1994.

[12] D.J. Dubois, *Fuzzy sets and systems: theory and applications*, Academic Press, 1980.

[13] Y. Enta, *Fuzzy decision theory, in Applied Systems and Cybernetics*, in: Proc. of the Int. Congress on Applied Systems Research and Cybernetics (G. E. Lasker, Ed.), Acapulco, Mexico, 1980, pp. 2980-2990.

[14] S.Z. Erdogan and T.T. Bilgin, A data mining approach for fall detection by using k-nearest neighbour algorithm on wireless sensor network data, *IET Communications* **6 (18)** (2012), 3281–3287.

[15] A. Fleury, M. Vacher and N. Noury, SVM-Based Multimodal Classification of Activities of Daily Living in Health Smart Homes: Sensors, Algorithms, and First Experimental Result, *IEEE Trans. on Information Technology in Biomedicine* **14 (2)** (2010), 274–283.

[16] S. Garcia, J. Derrac, J. Cano, and F. Herrera, Prototype selection for nearest neighbor classification: Taxonomy and empirical study, *IEEE Trans. on Pattern Analysis and Machine Intelligence* **34 (3)** (2012), 417–435.

[17] P.E. Hart, The condensed nearest neighbour rule, *IEEE Trans. on Information Theory* **18 (5)** (2016), 515–516.

[18] J. He, C. Hu, and X. Wang, A smart device enabled system for autonomous fall detection and alert, *Int. J. of Distributed Sensor Networks* (2016), 1–10.

[19] H. Kerdegari, S. Mokaram, K. Samsudin and A.R. Ramli, A pervasive neural network based fall detection system on smart phone, *J. of Ambient Intelligence and Smart Environments* **7(2)** (2015), 221–230.

[20] I. Kononenko and M. Kukar, *Machine learning and data mining: introduction to principles and algorithms*. Horwood Publishing, 2007.

[21] B. Krose, T. van Kasteren, C. Gibson and T. van den Dool, CARE: Context Awareness in Residences for Elderly, *Gerontechnology* **7** (2008), 101–105.

[22] S.W. Lee and K. Mase, Activity and location recognition using wearable sensors, *IEEE Pervasive Computing* **1(3)** (2002), 24–32.

[23] M. Lichman, *Uci machine learning repository* (http://archive.ics. uci.edu/ml), University of california, irvine, *School of Information and Computer Sciences, Irvine, CA, 2013.*

[24] W-C. Lin, C-F. Tsai, S-W. Ke, C-W. Hung and W. Eberle, Learning to Detect Representative Data for Large Scale Instance Selection, *J. of Systems and Software* **106** (2015), 1–8.

[25] U. Maurer, A. Smailagic, D.P. Siewiorek and M. Deisher, *Activity Recognition and Monitoring Using Multiple Sensors on Different Body Positions*, in: Proc. of the Int. WS on Wearable and Implantable Body Sensor Networks, 2006, pp. 1–4.

[26] R.E. McRoberts, Using satellite imagery and the k-nearest neighbors technique as a bridge between strategic and management forest inventories, *Remote Sensing of Environment* **112 (5)** (2008), 2212–2221.

[27] F. Palumbo, C. Gallicchio, R. Pucci and A. Micheli, Human activity recognition using multisensor data fusion based on reservoir computing, *J. of Ambient Intelligence and Smart Environments* **8(2)** (2016), 87–107.

[28] J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola J and I. Korhonen, Activity Classification Using Realistic Data From Wearable Sensors, *IEEE Trans. on Information Technology in Biomedicine* **10 (1)** (2006), 119–128.

[29] A.L. Ralescu and D.A. Ralescu, Probability and fuzziness, *Information Sciences* **34** (1984), 85-92.

[30] N. Ravi, N. Dandekar, P. Mysore, and M. Littman, *Activity recognition from accelerometer data*, in: Proc. of the 17th Conf. on Innovative Applications of Artificial Intelligence, 2005, pp. 1541–1546.

[31] J.L. Reyes-Ortiz, L. Oneto, A. Sama, X. Parra and D. Anguita, Transition-aware human activity recognition using smartphones, *Neurocomputing* **171** (2016), 754–767.

[32] S. Santini, R. Jain, Similarity is a geometer, *Multimedia Tools and Applications* **5 (3)** (1997), 277–306.

[33] M.V. Sokolova, J. Serrano-Cuerda, J.C. Castillo and A. Fernández-Caballero, A fuzzy model for human fall detection in infrared video, *Journal of Intelligent & Fuzzy Systems* **24(2)** (2013), 215–228.

[34] E.M. Tapia, S.S. Intille, W. Haskell, K. Larson, J.Wright, A. King and R. Friedman *Real-Time Recognition of Physical Activities and Their Intensities Using Wireless Accelerometers and a Heart Rate Monitor*, in: 11th IEEE Int. Symposium on Wearable Computers, 2007, pp. 1–4.

[35] R. Velik, A Brain-Inspired Multimodal Data Mining Approach for Human Activity Recognition in Elderly Homes, *J. of Ambient Intelligence and Smart Environments* **6 (4)** (2014), 447–468.

[36] Z. Younes, F. Abdallah, and T. Denœux, *Fuzzy multi-label learning under veristic variables*, in: Proc. of the IEEE Int. Conference on Fuzzy Systems, 2010, pp. 1–8.