A Novel Classification by Dual Regression Algorithm for Machine Learning over Time Series in Human Activity Recognition

Tengyue Li
University of Macau
Taipa, Macau SAR
yb97475@umac.mo

Simon Fong
University of Macau
Taipa, Macau SAR
ccfong@umac.mo

Antonio J. Tallón-Ballesteros
University of Huelva
Huelva, Spain
antonio.tallon@diesia.uhu.es

ABSTRACT

Human activity recognition (HAR) is a practical research area focused on recognizing specific action or movement a person from sensor data stream. In a general design, the data feeds from sensors are channeled quickly as data streams, to some data analysis module where real-time decisions are to be made. The latency between the sensors and the brain of the HAR, end-to-end, shall not be slower than the human bodily movement. This is especially true in crucial application scenarios such as IoT enabled hospitalization, machine vision for rescue operations, security surveillance by AI, sports analytics and object recognition in high-speed manufacturing etc. In the past decade, designs of high-speed IoT applications received a great deal of research attention. Many papers are published in the aspects of data modeling, big data processing, decision support system and implementation. Relatively there are less works on the fast data analytics for supporting HAR in real-time environments. The research work that is reported in this article sheds light into this particular area. A novel algorithm suitable for classifying recorded data from sensors into one of the predefined labels of activities is proposed. It is called classification by dual regression. Experiment results show that classification by dual regression achieves superior analytics performance for HAR.

CCS CONCEPTS

•Applied computing~Operations research~Forecasting

KEYWORDS

Human activity recognition, IoT data analysis, Forecasting, Regression

1 Introduction

HAR [1] is a modern field of computing which is essentially about recognizing what the activity a human or a group of people are doing automatically. It works by either directly programming a set of predicate rules which scrutinize the values of the sensor data for concluding a labelled activity, or by inducing up a predictive model for teaching a computer to recognize the patterns from the attributes of the input data, which resemble a predefined human activity like gestures, postures and actions. The analytics part of the HAR process is like an engine under the hood which decides the highest recognition accuracy the HAR system is able to achieve.

Traditionally data mining techniques have been deployed as the data analytic engines for building predictive models. Those techniques have a long history that can be traced back to the 60's where they were not mean for data of complex data attributes. For instance, in the context of Internet-of-things (IoT) [2], the data feeds are characterized as sequence of multi-dimensional data units or a collection of multiple time-series where the data are ordered chronologically by timestamps. Such data can be found in almost all IoT-related applications where sensors recorded data samples at certain frequency, and the data are relayed as a data stream at regularly paced intervals to the sink.

HAR applications broadly can be categorized as HAR by wearable sensors [3] and HAR by remote sensing [4]. The former type of applications relies on sensors that are attached on the body of the user, usually measures the vital signals and/or cartesian data of movement caused by the limbs or body of the user. Usually the data by wearable sensors are limited in attributes, with only one dimension of data (e.g. step counter, bloody pressure or heart pulse monitor) or several Cartesian attributes measuring the displacement of bodily actions in vectors of (x, y, z). The latter type of sensors includes but not limited to Microsoft Kinect camera which uses structured light and record depth images which are in relatively huge amount of point clouds; the programming functions by SDK development kit are able to record 60 joint points of information from a skeletal image. So, the movement of a human body is represented by a sequence of multi-dimensional data of 60 variables as data vector. Others are surveillance camera which captures human activities on video or animation, which is a sequence of images. Video analytics often require transforming the pixels into some simpler and representative image digest, suitable for further computer processing. What these HAR systems have in common are that the data are time-ordered, running as continuous data stream that potentially could amount to infinity. Different from traditional structured data where the data are in a two-dimensional matrix, such HAR data feeds in the format of time-series or multiple time-series running in synchronization poses certain computational challenges in machine learning. In this article, popular techniques for HAR ranging from data transformation to modification are introduced and reviewed. Specifically, a new type of machine learning algorithm called Classification by Dual Regression (CDR) is proposed – it is shown to be simpler than the prior-arts, yet it

yields profound performance as a result in classifying human activities from a multi-dimensional data stream generated from a wearable sensor. CDR is a simple approach in converting the attributes values pertaining to a specific class of activity into extra attributes to be added into training data. The additional attributes are the statistical representation of the time-series pattern taken from that attribute whose sequential values within that class label indeed form a time-series. The reminder of the article is structured as follow. Section 2 surveys about related techniques on formatting time-series and supporting machine learning over time-series. The CDR formulation is presented in Section 3. Experimentation is described in Section 4, followed by analyzing the results. Section 4 concludes.

2 Time Series Data Mining

The literature of Dynamic Time Wrap (DTW) is reviewed here, as DTW is one of the pioneer approaches in data mining on time-series. Then, state-of-the-art subsequence matching approaches based on DTW are reviewed and thus the motivation of proposing FPNS and NSPRING are introduced. Next, novel pattern discovery methods are reviewed and the motivation of proposing NCM is demonstrated. Furthermore, time-series mining approaches employed fuzzy logic are reviewed and the motivation of proposing IFSM-R* is illustrated. Lastly, several deep learning approaches are reviewed and the motivation of proposing DTWNet is presented.

DTW is a well-known distance measure with good accuracy but slow for time-series data mining. In recent years, a number of methods were proposed on accelerating DTW. These methods include NSPRING and UCR-DTW. Rakthan-manon et al. [5] claims that the DTW is the best distance measure. Itakura et al. [6] and Sakoe et al. [7] proposed Warping Window Constraint to constrain the path of DTW, which are called Sakoe-Chiba band and Itakura parallelo-gram respectively. The constraints can reduce the calculation by constraining the path of DTW. Keogh et al. [8] introduced a kind of Lower Bounding called LB Keogh for DTW, which accelerates the calculation of DTW. Keogh et al. [9] also introduced Early Abandoning to eliminate unnecessary calculations in DTW. Their approach accelerates DTW by stopping the distance calculation when the remaining calculations are deemed unnecessary. Rakthanmanon et al. [5] proposed UCR-DTW to combine several heuristics together to accelerate DTW.

A number of techniques were also proposed for subsequence matching. Majority of these approaches are based on DTW. Wong et al. [10] introduced a sliding window approach to index all possible prefixes with a spatial access method for subsequence monitoring on stored sequences. Sakurai et al. [11] introduced SPRING algorithm for subsequence monitoring. Both time and space complexity of the algorithm are $O(n)$. Peng et al. [12] proposed Fast Subsequence Matching (FSM) technique for accelerating the speed of SPRING. Niennattrakul et al. [13] proposed an approach to improve the accuracy of SPRING. Zhou et al. [14] also proposed cone indexing for subsequence monitoring and the complexity of the algorithm in best-case scenario is $O(nm)$. Papapetrou et al. [15] proposed an embedding-based framework to improve the efficiency of subsequence monitoring. However, all the above approaches do not consider normalization. There are situations where the data sets from some of the applications can only be processed after normalization. Therefore, Rakthanmanon et al. [5] proposed the UCR-DTW algorithm for subsequence monitoring with normalization. The proposed UCR-DTW is a fast method that can execute sequence of length one trillion in hours. Yet, UCR-DTW is complicated to implement and slower than SPRING. Therefore, we proposed NSPRING [16] which modifies SPRING to support normalization. In addition, FPNS [17] is proposed to further accelerate the speed of NSPRING.

For pattern discovery, efficiency is essential as it is normally computationally expensive. Although DTW is effective, ED has the advantage of efficiency. Accordingly, a number of pattern discovery approaches based on ED have been proposed in the literature. Chiu et al. [18] first proposed the motif (pattern) discovery approach. The smart brute-force (SBF) approach [19] was then proposed to calculate smartly the distance between neighboring subsequences to reduce the computational time. Mueen et al. [20] proposed the Mueen-Keogh (MK) approach, which uses a reference index to prune unnecessary calculations. However, the objective of motif discovery is slightly different from that of pattern discovery. Specifically, the goal of motif discovery is to find the most similar subsequence pair between two time series, whereas the aim of pattern discovery is to find all of the similar subsequence pairs between two time series. In addition, motif discovery requires the length of the motif to be fixed while pattern discovery allows the length of the pattern to be within an available range. Therefore, as the focus of motif discovery is different from that of pattern discovery, we do not compare our approach with the motif discovery approaches. The other branch of pattern discovery is the DTW-based approaches. Toyoda et al. [21] first tried to discover the patterns between two time series by only calculating a matrix. Toyoda et al. [22] then proposed a cross-similarity method for the pattern discovery. Later, the CM approach [23] was proposed to effectively and efficiently discover the patterns on time-series. Yet, none of these approaches consider normalization. In contrast, all of the ED-based motif discovery approaches consider normalization. Therefore, in this paper, we propose the NCM [24] approach to make the pattern discovery problem more applicable. Besides traditional time-series data mining researches, we reviewed many researches on time-series fuzzy data mining. However, there is no research tries to solve fuzzy subsequence matching problem. For fuzzy clustering, Anand et al. [25] extended template-based fuzzy clustering algorithm to incorporate functional annotation information available for genes. Izakian et al. [26] proposed three DTW-based fuzzy clustering algorithms, which are DTW-based Fuzzy C-Means, DTW-based Fuzzy C-Medoids and Hybrid of them. Besides, it is claimed in the paper hybrid clustering outperforms the other two. Maharaj et al. [27] transferred time-series from time domain to frequency domain and do fuzzy clustering approach based on the estimated cepstrum. For fuzzy

classification, Raviku-mar et al. [28] proposed a one- nearest-neighbor-Euclidean distance (1NN-ED) based classifier to fuzzy classify time-series. Based on the reviewed researches, we decided to develop novel fuzzy subsequence matching algorithms. After that, UCRSuite proposed by Rakthanmanon [5] gave us inspiration to accelerate the execution time of naive algorithm. In addition, we found that spatial indexing approach well fits the fuzzy subsequence matching problem, which is able to further accelerate the execution time by indexing sequences. Thus, two kinds of spatial indexing algorithms, i.e. kd-tree [28] and R*-tree [29, 30, 31], were used in our algorithms. Finally, IFSM-R* and IFSM-kd are proposed and IFSM-R* outperforms IFSM-kd.

3 Classification by Dual Regression Algorithm

Our proposed model CDR is extended from Classification by Regression (CR). CR was first invented by Ian Witten et al in 1998 which is a powerful classification model that classifies using linear regression methods. Witten has shown that classification problem can be efficiently solved by using simple linear regression especially if most or all the attributes in the dataset are numeric. Class is binarized and one regression model is constructed for each class value. When the linear regression applies by using the attributes values to formulate up a regression equation, the output of the regression equation outputs a numeric value between 0 and 1 for each instance. The predicted number is then used as a split point on deciding whether a testing data instance should belong to class 0 or class 1, judging from how much the predicted value is inclined towards class 0 or class 1. Setting a threshold for predicting class 0 or 1, the method is simple and efficient. Checking upon the regression line, for most 0 instances they will have a low value close to zero; and for most 1 instance, they have a larger value over the threshold. If the predicted value is less than the threshold, the model predicts to class 0; if it is greater, it predicts to class 1. For multi-class classification, where there is more than one class value, the model is generalized to more than two classes using a separate regression for each class. The output is predicted to 1 for data instances that belong to that class, and 0 for instances that do not. A separate regression line is inferred for each class, and given an unknown test data instances, the model will choose a class with the largest output. The CR model hence will have r regressions for a classification problem where there are r unique classes. Alternatively, pairwise regression could be used for discriminating the test samples into different classes. Pairwise regression takes every possible pair of classes by combination – that is $r^2/2$ of them. Then the CR uses a linear regression line for each pair of classes, discriminating an instance in one class of that pair from the other class of that pair. This simple CR method is chosen here for the sake of speed and accuracy in HAR which is one of the common extreme automation problems.

Typically, in HAR, a predictive model is built based on some training data that are described by a set of variables or attributes. In the simplest case of HAR by wearable sensors, the data are characterized by three Cartesian coordinates and a position or part of the body where the sensor is worn. So, there are three numeric attributes and one nominal attribute, and six unique activity types in the class called activity. The training dataset is provided by recording the input attributes values in relation to the known activities which a participant performed in advance. In HAR, the training process is equivalent to some calibration stage where a predictive model is induced into maturity. Then the model will be applied to real-time HAR [32] for predicting which activity a person is doing from the input of testing data. In the literature, linear regression is not very much used in HAR domain; rather complex models such as deep learning and neural networks are popular choices of machine learning in tackling the complex data patterns to be learnt with respect to the class. It is mainly because the maximum accuracy by linear regression is limited to a mediocre level, as the goodness of fit is often approximated by a straight line and the total residual is high. In this article, an alternative but quick and accurate approach called CDR is put forward.

The core of linear regression which is the most popular and simplest regression model, is a function $y=c_0=c_1x$ which maps the input data points as a vector $x_i = (x^1_i, x^2_i, \ldots x^m_i)$ by a total of m attributes at the i^{th} data instance to a predicted outcome y_i. c_0 and c_1 are coefficients of the regression model to be worked out by the data mining algorithm. A measure of goodness of fit, which is how well the regression line $y=c_0=c_1x$ predicts the activity y is the magnitude of the residual e_i at each of the n training data instances. Usually CR attempts to minimize the overall magnitude of the residual e^i over all i where $e_i=y_i-(a_0+a_1x_i)$. Therefore, minimization of the total residual is an objective of computing the suitable regression coefficients. If all the residuals ei are minimized to zero, an ideal equation is obtained in which all the data points will fit neatly along the regression line. For fast computation, a simple statistical measure called least square method is often used. It ensures the estimates of the coefficients of the CR are selected such that the sum of the squared residuals is kept at minimum, by optimizing $\sum_{i=1}^{n}(e_i)^2$. This simple regression operation is performed on our HAR case that has four variables – sensor type, x, y and z. For four input variables the equation takes the form: $y=c_0+c_1x_1+c_2x_2+ c_3x_3+c_4x_4$. CR computes the four coefficient values for $c_1..c_4$ and one constant c_0 that collectively minimize squares of the differences between the actual and predicted y values over n instances.

Figure 1 shows a standard CR time-series data mining procedure. Data stream of training data in vectors of <*sensor_type*, *x*, *y*, *z*, *class*=activity> arrive continuously into a regression model building process. As a standard supervised data mining process, CR receives a train of input training data, mines and learns over the relations between the attribute's values and the corresponding class label. The regression equation is inferred when the coefficients are computed with the objective of minimizing the chi-squared error. When the equation is ready, as well as the classification model matures, it can be used for testing. New testing data stream in with new attribute values which are

unseen by the model. The model predicts an outcome by using the regression equation by fitting in the attribution values.

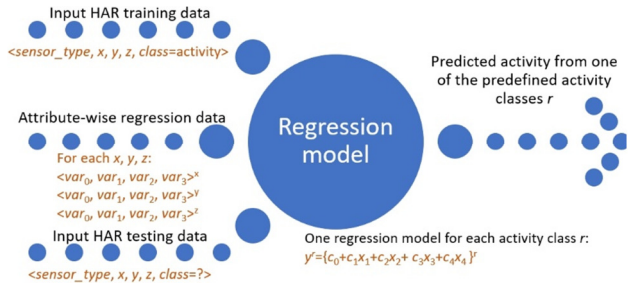


Figure 1: The data mining model for Classification by Dual Regression

The standard CR model is limited by the predictive performance since only a simple linear regression equation is used. Although it is fast and efficient suitable for meeting the real-time requirement of HAR, there are room for improvement in enhancing its performance. One simple approach which is proposed in this article is adding extra attributes which represent the time-series at the original attributes. Given a wearable sensor dataset as an example, the original spatial attributes are in fact in the form of 1-D time-series which collectively map to a certain activity label. Extending from CR, our new model namely CDR applies linear regression again on each of the time-series from the original attributes, creating a statistical digest. A quick transformation is adopted by using Dummy Variable Regression Analysis (DVRA) [33]. The time-series of the three variables (x, y, z) are visualized as an illustration in Figure 2. In this case, the variable values (x, y, z) are no longer treated as individual records of a tri-valued tuple at i^{th} time; but from a longitudinal view, the variable values are considered and viewed as three time-series whose patterns can characterize a corresponding activity label.

Similarly, other statistical components that are derived from the time-series of the three variables could be used as the extra attributes. For example, time-series could be transformed from temporal domain to frequency domain as Haar wavelets. Other option would be shapelets, subsequence and/or segment characters of a sequence and even classical statistical properties such as median, standard deviation, kurtosis, skewness could be used. Figure 3 shows an option of transforming the time-series into histograms.

A dummy variable in DVRA is an artificial variable generated to represent a new attribute that is dichotomous. The new variable is coded to represent the original attribute with a higher level of measurement. Dummy variables are often used in multiple linear regression – in our case, we have three time-series to be loaded into a multiple linear regression at the same time. Except for the constant and the residual, each of the terms in the function is a product of a regression coefficient and a variable. By choosing this additive form, we make the assumption that the 'effect' of one independent variable on the dependent variable is measured by the size of its own coefficient, and that this 'effect' is independent of the other variables and coefficients. The independent variables may still affect each other, but this does not preclude us from assuming that the effect of an independent variable on the dependent variable is unaffected by the other independent variables. Let us suppose we are concerned with the regression of a numerically scaled dependent variable Y, on a set of numerical independent variables X_1, X_2, etc.; furthermore, the population is partitioned into mutually exclusive classes, and we know to which class each item of the sample belongs. We want to study not only the influence of X_1, X_2, etc. on Y but also the effect of class membership. The goal of the CDR is to learn the relationship between the time-series by the original x, y, z attributes and the activity class. For this purpose, suppose we have six classification classes of activities. Since activity level is not a conventionally scaled attribute, we must somehow supply it with numerical values if we are to introduce it into a regression equation. To do this we define two dummy variables, var_0 and var_1 with the property that var_0=1 if the instance belongs to the current activity class; otherwise var_0=0 means the instance may belong to other class. These variables may then be substituted in the regression as equation variables in good standing provided the proper steps are taken to ensure that the solution of the normal equations will be determinate. In the general case we have multi-class classification where the number of different time-series attribute is m, of which the i^{th} instance contains k_i mutually exclusive classes. We define m sets of dummy variables one for each attribute, R_{ij}(i=1,2, …, m; j=1,2, …, k_i) so that R_{ij}=1 if the item belongs to the j^{th} class of the i^{th} regression model (one for each time-series variable); in all other cases R_{ij}=0. The generation of the regression equation is then: $Y = aX + \sum_{i=1}^{i=m} \sum_{j=1}^{j=ki} b_{ij} R_{ij} + c + \mu$. It is apparent that for any set of constants b_i*(i=1,2, …, m) and c* such that $\sum_{i=1}^{i=m} b_i^* + c^* = 0$, Y is identically unaffected by the substitution of b_{ij}+bi* and c+c* in place of b_{ij} and c, and determinate results again require a system of constraints. This is most easily arranged by leaving out one dummy variable from each set; i.e. select a j_i for each system of classes i, and pre-assign b_{ij}=0 where i=1,2, …, m.

Visually the transformation of a time-series of a variable to the dummy variables pertaining to a specific class activity is shown in Figure 4. The four dummy variables for each x, y, and z original attributed are added in the coloured columns representing the regression curve behavior of the corresponding class. Once the dummy-variable-regression curves for each of x, y and z are established, the statistical digests of the curves are added into the training data. Note that for each activity class, separate sets of dummy-variable-regression curves for x, y, z values are created independently from those of other classes. The digests are added too to the training dataset. An illustration of the dummy-variable-regression curves are shown in Figure 5, having one curve representing a particular class activity and the other curve representing all the other activities.

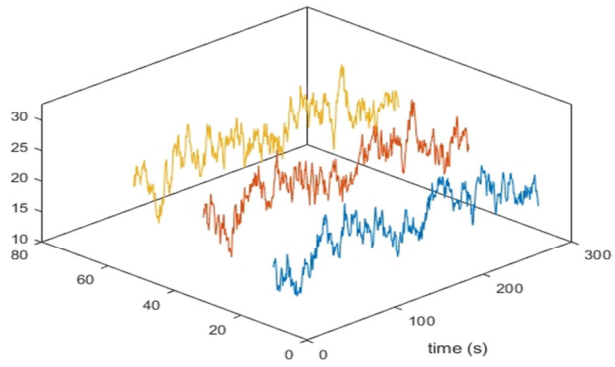


Figure 2: Time series as variable values in the case of HAR dataset

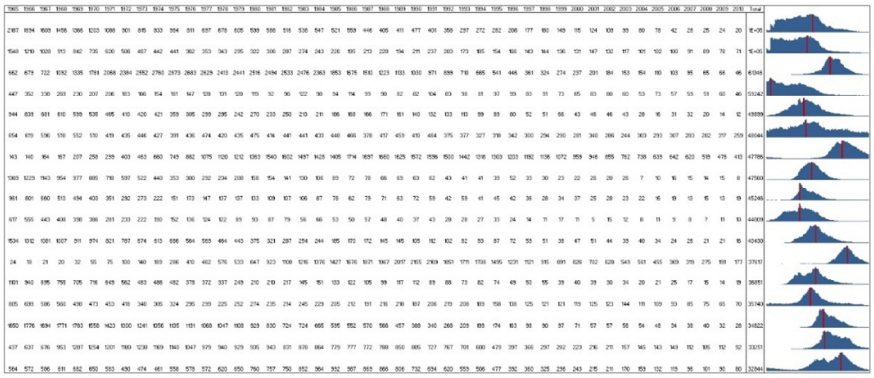
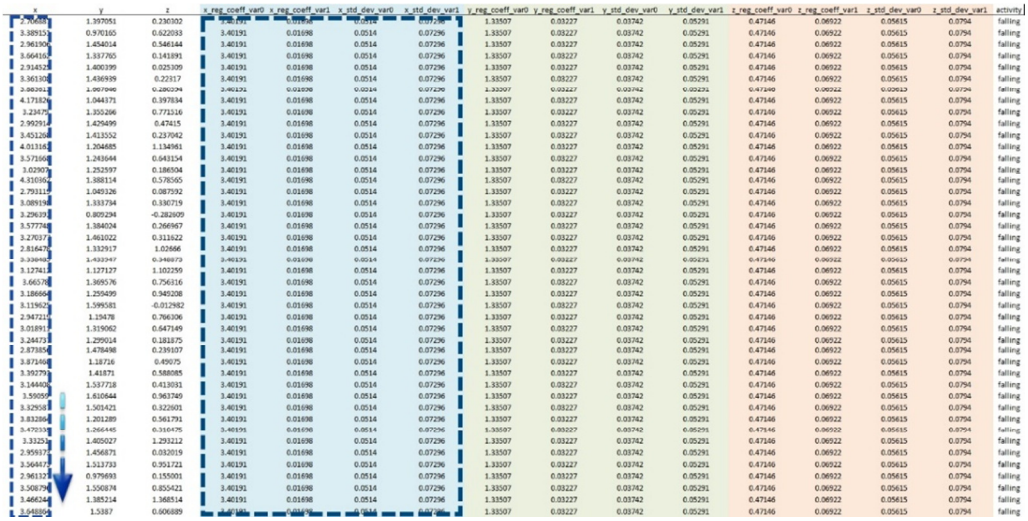


Figure 3: Transformation of time-series into histograms



x	y	z	x_reg_coeff_var0	x_reg_coeff_var1	x_std_dev_var0	x_std_dev_var1	y_reg_coeff_var0	y_reg_coeff_var1	y_std_dev_var0	y_std_dev_var1	z_reg_coeff_var0	z_reg_coeff_var1	z_std_dev_var0	z_std_dev_var1	activity
2.96088	1.397051	0.230302	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.38915	0.970165	0.622033	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
2.96190	1.454014	0.546144	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.66416	1.337765	0.141891	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
2.91452	1.400399	0.025309	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.36130	1.436939	0.23917	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
2.86261	1.097046	0.280239	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
4.17182	1.044371	0.397834	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.23475	1.350266	0.771516	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
2.99291	1.429499	0.47415	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.45126	1.413552	0.237042	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
4.01316	1.204685	1.134961	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.57166	1.243644	0.643154	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.02907	1.252597	0.186504	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
4.31036	1.388114	0.578565	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
2.75311	1.049326	0.087592	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.08918	1.333734	0.330719	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.29639	0.809294	-0.282609	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.57774	1.384024	0.266967	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.27037	1.461022	0.311622	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
2.81647	1.332917	1.02666	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.26968	1.493247	0.348873	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.12741	1.127127	1.102259	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.66578	1.369576	0.756316	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.18666	1.259499	0.949208	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.11962	1.599581	-0.012982	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
2.94721	1.19478	0.796106	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.01891	1.319062	0.647149	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.24473	1.299014	0.181875	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
2.87385	1.478498	0.235107	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.87166	1.18716	0.49075	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.39279	1.41871	0.588085	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.14405	1.537718	0.413031	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.59059	1.610644	0.963749	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.32958	1.501421	0.322601	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.83286	1.201289	0.561791	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.47233	1.266445	0.316075	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.33252	1.405027	1.293212	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
2.95917	1.456871	0.032019	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.56447	1.513733	0.951721	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
2.96132	0.975693	0.155001	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.50879	1.550874	0.855421	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.46624	1.385214	1.368514	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling
3.54899	1.5387	0.606889	3.40191	0.01698	0.0514	0.07296	1.33507	0.03227	0.03742	0.05291	0.47146	0.06922	0.05615	0.0794	falling

Figure 4: A snapshot of how the time-series under a spatial variable x being transformed into four dummy variables by DVRA

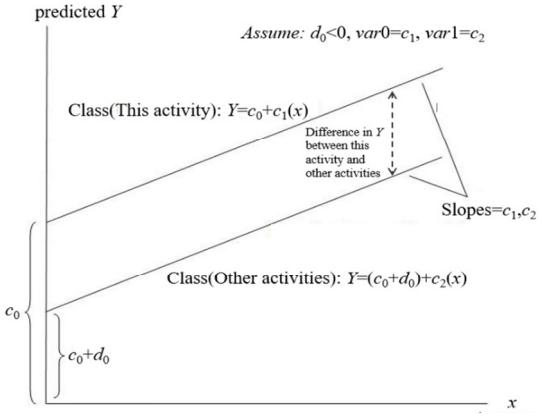


Figure 5: Regression curves of dummy variables

The steps of CDR operation are described as follow:

1. Normalize the original attributes *x*, *y*, *z* in the training dataset
2. For each of the x, y, z attributes, apply DVRA to obtain *var*0, *var*1, *var*2 and *var*3 where *var*0 and *var*2 are the gradients of the dummy-variable-regression-curves respectively, *var*2 and *var*3 are the standard deviations of the regression curves. This step repeats for each activity class.
3. The expanded training dataset with newly added attributes are used to build a classification model for HAR
4. During testing, steps 1-3 are repeated on the testing data. Since a-priori information is unavailable about the activity classes, either markers between different activities or clustering is required to bound the *x*, *y*, *z* attribute values for framing up the respective time-series.

4 Experiment

For validating the efficacy of the proposed CDR model, an empirical dataset that representing a classical scenario of HAR is used in the data mining experiment. The dataset which represents a case of extreme automation in HAR is donated by Mitja, Bostjan, Rok, Jana and Vedrana from Jozef Stefan Institute, Ljubljana, Slovenija. The wearable sensor data are recorded by people who were wearing four tags on various parts of the body - left and right ankles, belt and chest. Each data instance is a localization data for one of the tags. The data instances are ordered sequentially by timestamps. There are 10 different activities as target class for prediction, namely "walking, falling, 'lying down', lying, 'sitting down', sitting, 'standing up from lying', 'on all fours', 'sitting on the ground', 'standing up from sitting', 'standing up from sitting on the ground". A total of 8000 data instances are taken as sampled in the experimentation, with the objective of classifying the x, y, z values of each sensor into one of the 10 different activity classes. During the experiment, the ratio of the training and testing is changing, from 5% to 95%. This represents the situations where different proportions of data are used for training. At the extreme end where only 5% of the data is used for training, 95% of the data would be used for testing. The machine learning algorithm adopted here is random forest, which is known to be one of the most powerful predictive modeling algorithms. It is well-known for its ability to try and pick a most suitable decision tree with a compact size that avoids overfitting and underfitting. Dimension reduction is not applied in this experiment, as there are relatively few attributes, although each *x*, *y*, *z* time-series attributes are expanded four-folds. For other HAR data analytics such as Kinect with depth data, the overwhelmingly large number of features should undergo feature reduction especially our CDR technique will multiply the feature amount four times. This dataset can be freely accessed at the UCI repository upon citation of: http://archive.ics.uci.edu/ml/datasets/Localization+Data+for+Person+Activity. In terms of performance indicators, Accuracy, Kappa statistics, True Positive and False Positive rates are used. Accuracy is simply the percentage of correctly classified data instances into the activity classes from the total number of testing data. Kappa statistics measures inter-rater agreement for qualitative (categorical) items. It is generally thought to be a more robust measure than simple percent agreement calculation, as Kappa takes into account the possibility of the agreement occurring by chance. True positive rate measures the proportion of actual positives that are correctly identified as such. The false positive rate (or "false alarm rate") usually refers to the expectancy of the false positive ratio. A false positive occurrence happened when it is predicted a sample to be positive but actually it is not so. The samples are put into two group – one group is treated with CDR which have been pre-processed by the transformation, the other group is the control data without any intervention applied. The results with respect to these four performance indicators are charted in Figures 6 and 7 respectively.

Figure 6: Accuracy and Kappa performance of time-series datamining with and without CDR

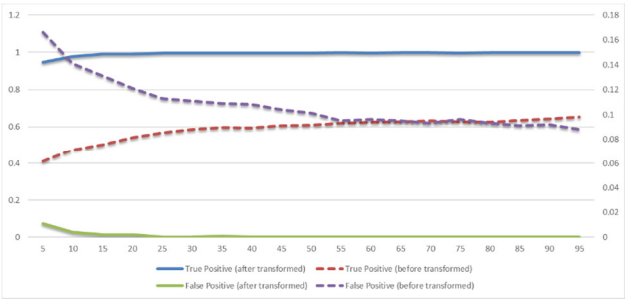


Figure 7: True-positive-rate and false-positive-rate performance of time-series datamining with and without CDR

In both charts, the dashed lines are the performance of the random forest HAR model that was built from time-series data before CDR transformation. They are generally limited in performance. For instance, in Figure 6 the accuracy ranges between 52% and less than 70% even when 95% training data was made available for well training. The Kappa is ranging from 0 to slightly over 0.5 that means the model is totally useless when only a small amount of training data was made available (5%); even when 95% of training data was provided, the model reliability is very mediocre at approximately 0.5. Based only on the original dataset it is inherently difficult to map the individual data instances per row to the corresponding class activity during machine learning. However, with CDR transformation applied, instead of individual data records row by row, the regression digest by CDR can be used to map a relation to the class activity. Therefore, both accuracy and Kappa achieve a near perfect score in Figure 6.

In Figure 7, it is ideal to have the true-positive rate at 1 while keeping the false alarm rate at zero. This is the case for the predictive model when CDR is used to transform the data, as shown by the solid curves. However, the original curves are attaining a poor accuracy level (0.4) and unacceptably high false alarm level (0.16) at 5% of available training data. It can be observed from both charts that after CDR is applied the performance of the HAR predictive model is superior.

5 Conclusion

One important practical research is human activity recognition which is characterized by the challenging real-time data mining requirements as well as the demand of highest possible accuracy especially for critical security applications. In the past decade, as sensor and camera technologies evolved, massive amount of data are collected for surveillance, monitoring and behaviour analysis. Relatively, the so-called AI part of the application which is often powered by fast data mining is neglected. Using empirical testing dataset for HAR experimentation, it is shown that when the original data are used for building a well-known powerful data mining model, by random forest, the performance is very limited. Observing that such data streams generated from body movements, picked up by sensors and feeding to the analytics process, they are indeed time-series which exhibit certain patterns when different activity is performance by the user. In light of this time-series property, a novel transformation is proposed in this article called Classification by Dual Regression (CDR). CDR features a double step of applying regression: one at the data transformation process and the other one at the machine learning process. The time-series of the original variables are converted into statistical digest from a regression curve that is induced based on the original time-series. Such statistical digest is added to the training dataset, enabling the data mining model to learn better than using only the individual data records per instance when the data is streaming in. At the prediction, regression is again used to predict a class based on a simple regression equation whose coefficient values are induced during the learning process. It is found that the results of the CDR are superior to the original data training without CDR. Furthermore, as future work, two challenges need to be further investigated. One is how to incremental infer a dynamic regression curve, adaptive to the incoming data from the data stream. In real life, incoming data are arriving as high-speed data feeds rather than batch of datasets from which a regression model is to be built. The other challenge is on marking the bounds of each activity automatically on the testing data, either by clustering or other similarity comparison approach. CDR contributes to enhancing the performance of data analytics for HAR, making HAR more suitable for extreme automaton where accuracy and analytics speed are of high priority.

ACKNOWLEDGMENTS

TBA.

REFERENCES

[1] Y. Feng, C. Chang, H. Ming, 2018, Engaging Mobile Data to Improve Human Well-being: the ADL Recognition Approach, *IT Professional*, (Early Access) Pages: 1 - 1

[2] R. D. Sriram, A. Sheth, 2015, Internet of Things Perspectives, *IT Professional*, Volume: 17, Issue: 3, Pages: 60 - 63

[3] H. Mizuno; H. Nagai; K. Sasaki; H. Hosaka; C. Sugimoto; K. Khalil; S. Tatsuta, 2007, Wearable Sensor System for Human Behavior Recognition (First Report: Basic Architecture and Behavior Prediction Method), *TRANSDUCERS 2007 - 2007 International Solid-State Sensors, Actuators and Microsystems Conference*, Pages: 435 - 438

[4] A. Jalal, Y. Kim, S. Kamal, A. Farooq and D. Kim, 2015, Human daily activity recognition with joints plus body features representation using Kinect sensor, *2015 International Conference on Informatics, Electronics & Vision (ICIEV)*, Pages: 1 - 6

[5] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B.Westover, Q. Zhu, J. Zakaria, E. Keogh, 2012, Searching and mining trillions of time series subsequences under dynamic time warping, in: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 262-270

[6] F. Itakura, Minimum prediction residual principle applied to speech recognition, 1975, *IEEE Transactions on Acoustics, Speech and Signal Processing* 23 (1) 67-72

[7] H. Sakoe, S. Chiba, 1978, Dynamic programming algorithm optimization for spoken word recognition, *IEEE Transactions on Acoustics, Speech and Signal Processing* 26 (1) 43-49.

[8] E. Keogh, C. A. Ratanamahatana, 2005, Exact indexing of dynamic time warping, *Knowledge and information systems* 7 (3) 358-386.

[9] E. Keogh, L. Wei, X. Xi, M. Vlachos, S.-H. Lee, P. Protopapas, 2009, Supporting exact indexing of arbitrarily rotated shapes and periodic time series under euclidean and warping distance measures, *The VLDB Journal-The International Journal on Very Large Data Bases* 18 (3) 611-630

[10] T. S. F. Wong, M. H. Wong, 2003, Efficient subsequence matching for sequences databases under time warping, in: *Proceedings. Seventh International Database Engineering and Applications Symposium*, pp. 139-148.
[11] Y. Sakurai, C. Faloutsos, M. Yamamuro, 2007, Stream monitoring under the time warping distance*, in: Proceedings of the 23rd International Conference on Data Engineering*, pp. 1046-1055.
[12] Z. Peng, S. Liang, J. Yan, H. W. Hong, Y. S. Qiang, 2008, Fast similarity matching on data stream with noise, *in: IEEE 24th International Conference on Data Engineering Workshop (ICDEW)*, pp. 194-199.
[13] V. Niennattrakul, D. Wanichsan, C. A. Ratanamahatana, 2010, Accurate subsequence matching on data stream under time warping distance, *in: New Frontiers in Applied Data Mining*, pp. 156-167.
[14] M. Zhou, M. H.Wong, Efficient online subsequence searching in data streams under dynamic time warping distance, 2008, *IEEE 24th International Conference on Data Engineering (ICDE)*, pp. 686-695.
[15] P. Papapetrou, V. Athitsos, M. Potamias, G. Kollios, D. Gunopulos, 2011, Embedding-based subsequence matching in time-series databases, *ACM Transactions on Database Systems (TODS)* 36 (3) 17.
[16] X. Gong, S. Fong, J. H. Chan, S. Mohammed, 2015, Nspring: the spring extension for subsequence matching of time series supporting normalization, *The Journal of Supercomputing* 1-25.
[17] X. Gong, S. Fong, Y.-W. Si, 2018, Fast multi-subsequence monitoring on streaming time-series based on forward-propagation, *Information Sciences* 450. pp.73- 88.
[18] B. Chiu, E. Keogh, S. Lonardi, Probabilistic discovery of time series motifs, 2003, *in: Proceedings of the 9th international conference on Knowledge discovery and data mining (SIGKDD)*, pp. 493-498.
[19] A. Mueen, Enumeration of time series motifs of all lengths, 2013*, in: IEEE 13th International Conference on Data Mining (ICDM)*, pp. 547-556.
[20] A. Mueen, E. J. Keogh, Q. Zhu, S. Cash, M. B. Westover, 2009, Exact discovery of time series motifs., *in: Proceedings of the 2015 SIAM International Conference on Data Mining*, Vol. 9, pp. 473-484.
[21] M. Toyoda, Y. Sakurai, T. Ichikawa, 2008, Identifying similar subsequences in data streams, *in: Database and Expert Systems Applications*, pp. 210-224.
[22] M. Toyoda, Y. Sakurai, Discovery of cross-similarity in data streams, 2010, *in: IEEE 26th International Conference on Data Engineering (ICDE)*, pp.101-104.
[23] M. Toyoda, Y. Sakurai, Y. Ishikawa, 2013, Pattern discovery in data streams under the time warping distance, *The VLDB Journal* 22 (3) 295-318.
[24] X. Gong, S. Fong, R. K. Wong, S. Mohammed, J. Fiaidhi, A. V. Vasilakos, 2016, Discovering sub-patterns from time series using a normalized cross-match algorithm, *The Journal of Supercomputing* 72 (10) 3850-3867.
[25] A. Anand, N. R. Pal, P. N. Suganthan, 2010, Integration of functional information of genes in fuzzy clustering of short time series gene expression data, *in: Proceedings of the IEEE Congress on Evolutionary Computation*, pp.1-8.
[26] H. Izakian, W. Pedrycz, I. Jamal, 2015, Fuzzy clustering of time series data using dynamic time warping distance, *Engineering Applications of Artificial Intelligence* 39, 235-244.
[27] E. A. Maharaj, P. D' Urso, 2011, Fuzzy clustering of time series in the frequency domain, *Information Sciences* 181 (7) 1187-1211.
[28] P. Ravikumar, V. S. Devi, Fuzzy classification of time series data, 2013, *in: IEEE International Conference on Fuzzy Systems*, pp. 1-6.
[29] J. L. Bentley, 1975, Multidimensional binary search trees used for associative searching, *Communications of the ACM* 18 (9) 509-517.
[30] C. Faloutsos, M. Ranganathan, Y. Manolopoulos, 1994, Fast subsequence matching in time-series databases, *in: Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data*, Minneapolis, Minnesota, May 24-27, 1994, pp. 419-429.
[31] N. Beckmann, H.-P. Kriegel, R. Schneider, B. Seeger, 1990, The r*-tree: An efficient and robust access method for points and rectangles, *in: Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data, SIGMOD '90*, pp. 322-331.
[32] Y. Han ; S.-L. Chung, J.-S. Yeh ; Q.-J. Chen, 2013, Real-time skeleton-based indoor activity recognition, *Proceedings of the 32nd Chinese Control Conference*, Pages: 3965 - 3970
[33] G. Damodar, 2003. *Basic econometrics*. McGraw Hill. p. 1002. ISBN 0-07-233542-4.