

Function to Flatten Gesture Data for Specific Feature Selection Methods to Improve Classification

Marilú Cervantes Salgado*, Raúl Pinto Elías, Andrea Magadán Salazar

Department of Computer Science, National Center for Research and Technological Development, Cuernavaca 62490, Morelos, Mexico

Corresponding Author Email: marilu.cersa@cenidet.edu.mx

https://doi.org/10.18280/ts.380402

ABSTRACT

Received: 19 May 2021 Accepted: 25 July 2021

Keywords:

Bayesian networks, chronologically linked data, feature selection methods, gesture classification, Logical Combinatorial to Pattern Recognition, Markov blanket Gestures are pieces of information with characteristics such as: multiple and chronologically linked samples with different length. The gesture characteristics mentioned before make classification, of this type of data, a challenging task. We studied the effects of flattening gesture data. We proposed a function to represent gestures in a flat format taking in consideration the evolution sense they possess. The function's main goal is to compare gestures intra class to spot differences. This function is described step by step and then its outcome is used as input to two feature selection methods (Bayesian network / Markov blanket and Logical Combinatorial to Pattern Recognition). After, with the subsets obtained, we trained Hidden Markov Models machines. We found that applying our methodology to gesture data, the subset of attributes obtained (feature selection) were able to classify with accuracies of 0.88 and 0.87 of a maximum of 0.90. The maximum accuracy was obtained from an exhaustive classification exercise we performed in order to compare our results. These findings suggest that our methodology can be applied over raw data (gesture data or any chronologically linked data) without the need of experts to transform data (i.e. feature extraction).

1. INTRODUCTION

Today, not touching is the new normality; for this reason, classifying gestures or any human movement is a great deal [1]. In order to classify gestures, models have to take in consideration that gestures are not executed in the same way per one person to another. Adding to that, gestures have an evolutionary property [2]. This property links the data within the gesture in a chronological order.

There are methods for gesture classification, among them are: training a machine with all attributes, feature extraction [3-5], feature selection using some statistical approach [6], and we could include time series studies [7].

Training a learning machine with all data's attributes leads us to an apparently simple solution, there is not preprocessing task to do. However, there are attributes that are by nature irrelevant or redundant [8], this can confuse the algorithm and the results could be poor.

Using a feature extraction approach allows to incorporate the evolution property of gestures to the transformation. For this, the presence of experts in the field is required and studies are specific to the data universe [9-11].

Another explored option has been time series studies but, gestures cannot be treated as time series for the following reasons: gestures examine a full description of the event within the observational time; thus past and future are not relevant to describe them. Also, time series objects have components, e.g. trend, seasonal, cyclical and random variations, which could or could not be present in gestures, human movements or objects of interest (video surveillance). Before examine feature selection for gesture data, let us describe the characteristics that make gesture data a complex problem. Gestures are chronologically linked multi-sample pieces of information that we will call objects. These objects could have different length within the same class, but the number of attributes is the same for all of them. In Figure 1, we illustrate four different gestures; in this: gesture a, Figure 1(a), is described in eight time units; gesture b, Figure 1(b) takes four time units to be completed, gesture c and d, Figure 1(c) and 1(d), take six time units.

As Figure 1 illustrates, two gestures of the same class can have different length (Figure 1(a) and 1(b)). It is also shown that the elements of two different gestures classes (Figure 1(a) and 1(c)) could be described with the same values, but different chronological order. Finally, in the same figure, we can notice that the median of the attribute Hx in gestures Figure 1(c) and 1(d) is the same but they belong to different classes.

Gesture data can be summarized as:

- Multi-sample objects;
- Objects with different length;
- Chronologically linked.

We have to remark the importance of the evolution sense of each gesture as part of the richness of this data and part of the problem to solve.

We see three options for performing feature selection over this type of data: 1) look for a selection technique that takes care of gesture data characteristics; 2) come out with a new feature selection technique that can handle this data; or, 3) find a way to adapt data to a classical feature technique. Any of the options has to ensure that the richness of the data is preserved.



Figure 1. Illustration of four gestures and their information coordinates. (a) and (b) Hand up class. (c) Hand down class. (d) Neutral, no gesture

For the first option, we looked into Group Lasso and multi view techniques since they were pointed out to handle structure and heterogeneous data [12]. Group Lasso [13-15] can handle chronologically linked data; however, it cannot differentiate between the movements of sitting on a chair and standing from it. Techniques for supervised multi-view feature selection [16, 17] are able to model multi sample data but these do not take care of the evolution sense of gestures.

Second option would be to come out with a new feature selection technique, which is outside the scope of this work.

Third option, here is where our work takes place; it is to find a way to capture the characteristics of gesture data into a flat representation to be used with a classical feature selection technique. We tackled this problem by proposing a function that converts gesture data into a single representation considering the evolution sense, the chronological linkage and the difference of length of each gesture (object).

We are proposing a function that compacts multi-sample gesture objects into a single value. Single representation of gesture data has a main advantage: the chance to be feed into classical feature selection methods that have demonstrated efficiency.

The feature selection methods chosen for this study were: Bayesian networks and Logical Combinatorial to Pattern Recognition (LCPR) [18]. The decision was made based on the theoretical model on which they work.

Since Bayesian networks are based in the notion of conditional independence; which defines that given a set of variables, if Pr(A = a | X = x, B = b) = Pr(A = a | X = x) then *B* gives none extra information about *A*, making *A* and *B* conditionally independent [19]. Our perspective focuses on finding the extra information (intra class differences) to construct a Bayesian network. Then, the feature selection is done using the Markov blanket concept.

On the other hand, for a LCPR approach, we applied the one based on representative sets as described by Martínez-Trinidad et al. [20]. A general idea of this approach can be seen in Figure 2.

Preliminary results, of classifying the variables subsets, show that our perspective can express gesture data into flat representation with error rate of 0.1178 percent of a minimum possible of 0.094.



Figure 2. General idea of representative sets for LCPR based on [20]

With the preliminary results we see improvement in classification rates using feature selection over using all attributes. With this, we conclude that our flattening function is achieving its purpose: it is compressing the evolution of the gesture, in a single representation, to be use in specific feature selection methods.

2. METHODOLOGY

Several methods integrate the methodology used in this study. In Figure 3, we present a diagram that depicts the entire methodology that was followed. For this study we use raw gesture data; raw data is fed to the flattening function in order to obtain a difference matrix; the difference matrix is then fed into the feature selection methods which output the most relevant feature of the dataset, finally the relevant features are utilized to train a Hidden Markov Models (HMM) machine.

The center explanation is on the flattening function that we are proposing and is given step by step.

In this methodology section, we start describing the difference matrix, which gives a general idea of the implications and considerations to be taken in the flattening function.



Figure 3. The methodology used in this study

Next our proposed *Flattening Function* is described step by step (which is the contribution to this research work). Immediately after, we describe the approach followed to set the threshold for deciding on the similarity of two objects.

Then, in section Materials, the datasets for experimentation are described, as well as the tools for the two feature selection methods. Also, in the same section, we point out why the HMM is used to classify.

2.1 Difference matrix

The proposed function (section 2.2) was developed with a result on mind: to find a single representation that signifies the comparison of two objects of different length; this all done respecting the chronological order of elements in each of the objects. For this, the comparison between objects was done in

order to obtain a difference matrix. This difference matrix was composed of values that describe either or not two objects of the same class are different. A difference matrix can be visually described with Figure 4 where *object1* and *object2* belong to the same class. If the two objects are considered similar, we place a cero in the difference matrix. On the other hand, an identifier different from zero is written.

The number of elements (rows) on the difference matrix is given by,

$$DM_{rows} = \sum_{i=1}^{c} \frac{K_i!}{2!(K_i - 2)!}$$
(1)

where, K_i is the number of objects in class *i*, 2 is the number of elements in each subset to be combined and *c* the total number of classes in the sample. In other words, *DM* is the sum of binomial coefficient for the number of objects in each class choose by two.

There are two objects in Figure 4, both with different length. Our perspective aims to compare two objects of different length (most of the time), element by element in chronological order.

2.2 Proposed flattening function

The contribution part of the methodology is presented in the following list of steps. This list of steps shows how to obtain a difference matrix from gesture datasets.

2.2.1 Step 1: Objects sense

This step aimed to detect the evolution sense of objects in every variable for later comparison. Since objects are real number sequences, the evolution was captured as,

$$EvO_{i} = \begin{cases} +1 \ IF \ O_{i_{r}} < O_{i_{r+1}} \\ 0 \ IF \ O_{i_{r}} = O_{i_{r+1}} \\ -1 \ otherwise \end{cases}$$
(2)

where, *r* represents the *r*th element of object O_i . The object sense (EvO_i) will be later used to compare (in step 5), at once, the real values sequences and the evolution sequences of two objects.



Figure 4. Visual description of how the structure of a difference matrix is obtain

2.2.2 Step 2: Normalization

Each object was transferred to start in cero using,

$$TO_{i_r} = O_{i_r} - O_{i_1}$$
 (3)

2.2.3 Step 3: Scaling

The scaling approach taken was,

$$SO_{i_r} = \frac{O_{i_r}}{\min O_i - \max O_i} \tag{4}$$

2.2.4 Step 4: Setting a comparison window

Since the sense of each object was captured with step 1, we could compare two objects element by element (taking in consideration both: the sense and the actual value). However, since the values are real numbers, comparing two elements of two objects could result inefficient. That is why a range of comparison was set. This range was established with the slope value of the regression analysis of each object. After obtaining the slope of each object, we set the range of comparison as given in Eq. (5),

$$Range Q_{i_{x}} = \pm \beta \tag{5}$$

where, β is the slope in the lineal regression analysis performed to each object.

2.2.5 Step 5: Longest Common Substring (LCS)

LCS algorithm [21] is used to identify how many elements of two objects are similar. We adapted LCS algorithm to perform over objects that have been processed with step two and three of this list; then, the comparison was designed to run over the range set in step four and the sense obtained in step one, at the same time.

2.2.6 Step 6: Decision

The decision of whether or not two objects are similar was taken based on the result of step 5 and the length of the two objects. The length of the observed objects matters because we had to evaluate what percentage of similarity was found in step 5. For this, the following strategy was taken:

• First, it was found which of the two objects is the shortest and which is the largest,

$$shO = \min\left(O_i, O_j\right) \tag{6}$$

$$lO = \max(O_i, O_j) \tag{7}$$

where, O_i and O_j are two objects being compared and r is the length of each.

• Second, a threshold was established as follows,

$$threshold = \begin{cases} \% shO_1 & IF shO \ge \% lO \\ \% shO_2 & otherwise \end{cases}$$
(8)

• Finally, the decision was set to be,

$$d = \begin{cases} 0 & IF \quad result > threshold \\ C & otherwise \end{cases}$$
(9)

where, *result* is the outcome of step 5 and C is the identifier to specify that the two objects are different. For the Bayesian Network approach, C is the number of each class, since the mathematical model for learning Bayesian nets requires to find extra information by classes. For the Logical Combinational approach, C is number one, since the algorithm does not require class information. In this way d is the outcome for comparing two objects. All the comparisons marks form up the difference matrix.

2.3 Threshold for decision

The decision whether two objects are different is based on: the result of the CLS algorithm and the length of the two objects being compared, Eq. (9).

If the result of the CSL algorithm exceeds a threshold, the objects are considered as similar.

The threshold is set based on the length of the two observed objects, Eq. (6) and Eq. (7). If the length of the shortest object is greater than or equal to, let say, 14 percent of the longest object, the threshold is set as 39%, otherwise the threshold is set as 51%. These percentages are the number of elements, of the shortest object, to be taken into account in the decision, Eq. (8).

In order to set the percentages for the threshold, we experimented with a range for each of them. The ranges in the experiments, as these were the best results in a semiexhaustive search, were set as:

- a) 5 to 20 percent. As the range we explored in order to establish how much of the shortest object has to be consider in comparison of the largest,
- b) 30 to 60 percent. As the range when the shortest object meet the length marked by percentage in a,
- c) 30 to 54 percent. As the range when the shortest object does not meet the length marked by percentage in a.

The experiments with the threshold for the decision were done using the Bayesian Network feature selection approach. We utilized the maximum classification accuracy (of the eight learning algorithms described in section 2.4.1) to find out where the highest accuracy was obtained.

2.4 Materials

2.4.1 Feature selection methods

As said before, the comparison of objects produces a difference matrix. Then, the difference matrix was fed to two classical feature selection methods: 1) a Bayesian network was learned using the difference matrix, after, a Markov blanket was obtained (which signifies the feature selection); and 2) we fed the difference matrix to the logical combinatory approach in order to obtain the *typical testors* which are consider subsets with the most relevant features.

Code was developed using R environment [22] to build Bayesian networks and their corresponding Markov blanket. From the *bnlearn* library [23] eight algorithms were applied to the difference matrix: PC, Grow-Shrink (GS), Incremental Association Markov Blanket (IAMB), Interleaved Incremental Association (Inter-IAMB), Tabu Search (Tabu), Max-Min Hill Climbing (MMHC), General 2-Phase Restricted Maximization (RSMAX2) and Hybrid HPC (H2PC); we chose these algorithms, from *bnlearn* library, because they are a mixture of constraint-based and score-based methods which are more appropriate to discover the independence relations between the variables in the model [24].

The Logical Combinatory approach was developed according to theory [25] and following the structure in Figure 2, where the block *similarity measures* is replaced with the proposed function in section 2.2.

2.4.2 HMM classification machine

Having the feature selection results, we trained a Markov Hidden Model machine in R environment using *depmixS4* library [26] with parameter *family* = *gaussian()* for all of the variables involved. HMM is one of the most utilized classification machine for gesture data [1].

2.4.3 Data

The gesture dataset full description as well as its past usage and details can be found at Refs. [27, 28]. Here, we describe some aspects that are important for this study: the *Gesture Phase Segmentation* dataset was obtained using a Microsoft Kinect sensor to get the position of left hand, right hand, left wrist, right wrist, head and spine in a Cartesian coordinate system for a three-dimensional space of three persons. Each person had to gesticulate while telling a comic story. Each person's file is listed with A, B or C identifiers; as well, each comic story is listed with numbers 1, 2 or 3 to differentiate between files. Other characteristics are shown in Table 1.

Notice that, the classification goal for the present study is intended by gestures, not by persons.

Table 1. Description of gesture phase segmentation dataset

Information	on Description			
Dataset type	chronologically linked			
Associated tasks	classification, clustering			
Number of samples	 A1 - 1747 frames, A2 - 1264 frames, A3 - 1834 frames. B1 - 1073 frames, B3 - 1423 frames. C1 - 1111 frames, C3 - 1448 frames 			
Number of attributes (variables)	raw information files: 18 attributes (double): <i>lhx, lhy, lhz, rhx, rhy, rhz, hx, hy, hz, sx, sy, sz,</i> <i>lwx, lwy, lwz, rwx, rwy, rwz</i> ; a timestamp attribute (integer) which was omitted for the present study; and one class attribute (factor): <i>phase</i>			
Classes	Rest, Preparation, Stroke, Hold, Retraction			
Donated by	University of Sao Pablo, Brazil			
Donation date	18th June 2014			
Location	https://archive.ics.uci.edu/ml/datasets/gesture+ phase+segmentation			

Another important aspect for the present study was to classify the gestures datasets as they are presented: in their raw

format; this includes the classification with five classes. However, as studies have been made with the same dataset [3, 27] separating the classes into *Rest* and *No Rest*, we decided to study the dataset in the same way.

2.5 Exhaustive classification for gesture phase segmentation dataset

In order to assess our modeling gestures perspective, we did an exhaustive classification of the 18 attributes in the *Gesture Phase Segmentation* dataset with files A1 and A2 for two and five classes, B1 and B2 for 5 classes and C1 and C3 for five classes. The exhaustive classification was done using HMM machines.

This exhaustive classification is the power set minus one (empty set is not relevant), which we divided in possible combinations by number of variables. With this, we found the highest classification accuracy possible for the datasets.

3. RESULTS AND DISCUSSION

3.1 Results

Having the results of the exhaustive classification as a point of comparison (section 2.5), we applied the methodology to the *Gesture Phase Segmentation* datasets. We performed the methodology to A1, B1 and C1 files for two and five classes. Results for two classes are presented in Table 2. It also contains the results of the exhaustive classification and the classification with all variables.

When using the flattening function to feed the Bayesian network / Markov blanket (BN / MB) and LCRP methods the accuracy shows a major increment over the accuracy found using all variables (marked as *None* in Table 2 and Table 3).

In Table 3, we show the experiments using five classes. In this, we see a major increment in the accuracy using the methodology of this paper over the accuracy result of classifying with all variables.

Even though the results for classification with five classes (Exhaustive best) show low accuracy (Table 3), we used these results as they are, just to assess how well our perspective does against the maximum possible accuracy in the dataset.

Table 4 shows the threshold's parameters used in each experiment as established in section 2.3.

The highest accuracy result found involves the LCRP method and C1/C3 files for two classes; for that, in Table 5, we show important rates obtained for it. Also, we show rates for A1/A2 files for two classes as highest results for BN / MB method.

Table 2. Results found ex	perimenting with datase	t gesture phase se	egmentation in two	classes format
		<u> </u>	0	

Fasture Solastian method	# of variables	# Classes	Files		A	
Feature Selection method			Train	Test	Accuracy	
None	18	2	Al	A2	0.131	
Exhaustive (best)	6	2	Al	A2	0.905	
BN / MB	4	2	Al	A2	0.882	
LCRP	6	2	Al	A2	0.877	
None	18	2	B1	<i>B3</i>	0.136	
BN / MB	3	2	B1	B3	0.864	
LCRP	2	2	B1	<i>B3</i>	0.863	
None	18	2	Cl	<i>C3</i>	0.250	
BN / MB	5	2	Cl	<i>C3</i>	0.901	
LCRP	4	2	CI	<i>C3</i>	0.977	

Table 3. Results found experimenting with dataset gesture phase segmentation in five classes format

Fasterne Salastien method	# of moriables	orighlag #Classon		es		
Feature Selection method	# of variables # Classes		Train	Test	Accuracy	
None	18	5	Al	A2	0.0585	
Exhaustive (best)	9	5	Al	A2	0.5308	
BN / MB	8	5	Al	A2	0.4438	
LCRP	6	5	Al	A2	0.4304	
None	18	5	B1	<i>B3</i>	0.1411	
Exhaustive (best)	7	5	B1	<i>B3</i>	0.4058	
BN / MB	4	5	B1	B3	0.3294	
LCRP	8	5	B1	<i>B3</i>	0.3294	
None	18	5	CI	<i>C3</i>	0.0994	
Exhaustive (best)	8	5	Cl	<i>C3</i>	0.5303	
BN / MB	8	5	CI	<i>C3</i>	0.1989	
LCRP	3	5	C1	СЗ	0.52	

Table 4. Threshold parameter for each experiment

Files involved		Classes	Threshold	
Train	Test	Classes	parameters	
Al	A2	2	a)14;b)39;c)51	
B1	B3	2	a)30;b)50;c)14	
Cl	<i>C3</i>	2	a)61;b)8;c)60	
Al	A2	5	a)51;b)20;c)76	
B1	B3	5	a)31;b)15;c)5	
Cl	<i>C3</i>	5	a)31;b)86;c)28	
Al	A2	2	a)14;b)39;c)51	
B1	<i>B3</i>	2	a)80;b)20;c)80	
Cl	<i>C3</i>	2	a)61;b)15;c)20	
Al	A2	5	a)51;b)20;c)76	
B1	<i>B3</i>	5	a)51;b)30;c)70	
Cl	С3	5	a)51;b)30;c)70	
	Files Train A1 B1 C1 A1 C1 A1 C1	Files involved Train Test A1 A2 B1 B3 C1 C3 A1 A2 B1 B3 C1 C3	Files involved Classes Train Test Classes A1 A2 2 B1 B3 2 C1 C3 2 A1 A2 5 B1 B3 5 C1 C3 5 C1 C3 5 C1 C3 5 A1 A2 2 B1 B3 2 C1 C3 2 B1 B3 2 C1 C3 2 B1 B3 2 C1 C3 2 A1 A2 5 B1 B3 5 C1 C3 5 C1 C3 5	

 Table 5. Rates summary for C1/C3 and A1/A2 files

File		Accuracy	Sensitivity	Precision	
Train	Test	-	Kest	Rest	NoRest
Cl	<i>C3</i>	0.977	0.961	0.948	0.987
Al	A2	0.882	0.9148	0.808	0.940

3.2 Discussion

Experiments with 2 and 5 classes have considerable increases in accuracy, when using the methodology here proposed, compared with the classification done using all the variables (Table 2 and Table 3). Accuracies obtained using feature selection do not go less of 10 percent points from the maximum obtained for a specific dataset (Exhaustive best) (Table 1 and Table 2). Rates for files C1/C3, show a real possibility of success in flattening gesture data for feature selection.

Madeo et al. [27] related in feature extraction in order to segment (classify) gestures. Their results, for files A1/A2 (training and testing) with two classes (Rest and No Rest) show the sensitivity value of 0.893 (for *Rest* class); while our best results for the same experiment are 0.914.

Threshold parameters impact in the final result. The threshold had to be adjusted for each of the feature selection methods and for each file (datasets) (Table 3). Threshold settings were not exhaustive search.

Half of the datasets used in the experiments are not balance (five classes format), nonetheless the results of an exhaustive classification give us a benchmark to compare the results.

4. CONCLUSIONS

The main goal of this study is to encapsulate the evolution sense of gestures into a flat representation in order to perform feature selection, while maintaining the chronological property in data. We found that applying our proposed function, the outcome subsets of attributes reach accuracies of 0.88 and 0.87 of a maximum possible of 0.90 (A1/A2 files two classes). Rates for files C1/C3, show a real possibility of success in flattening gesture data for feature selection.

We conclude that our methodology can be applied on raw data, without the intervention of experts to transform the attributes; and still take in consideration the richness of gesture data.

The search for the decision threshold could be seen as a lineal problem, which would represent a smaller search than an exhaustive classification for datasets with large number of attributes. Thus, the results here presented could improve with a better parameter tuning in the decision threshold. The decision threshold parameters tuning is not a trivial problem, yet it is beyond the scope of this study. This issue being a problem for future work.

ACKNOWLEDGMENT

Authors thank Dr. J. Ochoa for sharing code that helped performing the LCPR experiments.

The work of Marilú Cervantes Salgado was supported by Consejo Nacional de Ciencia y Tecnologia (CONACyT) under the concept of doctoral scholarship.

REFERENCES

- Liu, H., Wang, L. (2018). Gesture recognition for human-robot collaboration: A review. International Journal of Industrial Ergonomics, 68: 355-367. https://doi.org/10.1016/j.ergon.2017.02.004
- [2] Viblis, M.K., Kyriakopoulos, K.J. (2000). Gesture recognition: The gesture segmentation problem. Journal of Intelligent and Robotic Systems, 28(1): 151-158. https://doi.org/10.1023/A:1008101200733
- [3] Wagner, P.K., Peres, S.M., Madeo, R.C.B., Lima, C.A.M., Freitas, F.A. (2014). Gesture unit segmentation using spatial-temporal information and machine learning. Presented at Florida Artificial Intelligence Research Society Conference.
- [4] Islam, M.R., Mitu, U.K., Bhuiyan, R.A., Shin, J. (2018). Hand gesture feature extraction using deep convolutional neural network for recognizing American sign language. 4th International Conference on Frontiers of Signal Processing (ICFSP), pp. 115-119. https://doi.org/10.1109/ICFSP.2018.8552044
- [5] Mahmood, M.R., Abdulazeez, A.M. (2019) Different model for hand gesture recognition with a novel line feature extraction. International Conference on Advanced Science and Engineering (ICOASE), pp. 52-57. https://doi.org/10.1109/ICOASE.2019.8723731
- [6] Ochoa, J., Pinto, R. (2016). Identification of relevant features based on the variation of chrono-valued descriptors. IEEE Latin America Transactions, 14(2): 936-942. https://doi.org/10.1109/TLA.2016.7437243
- Yoon, H., Yang, K., Shahabi, C. (2005). Feature subset selection and feature ranking for multivariate time series. IEEE Transactions on Knowledge and Data Engineering, 17(9): 1186-1198.

https://doi.org/10.1109/TKDE.2005.144

- [8] Koller, D., Sahami, M. (1996). Toward Optimal Feature Selection. Presented at ICML.
- Shanableh, T., Assaleh, K., Al-Rousan, M. (2007). Spatio-temporal feature-extraction techniques for isolated gesture recognition in Arabic sign language. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 37(3): 641-650. https://doi.org/10.1109/TSMCB.2006.889630
- [10] Li, G., Zhang, R., Ritchie, M., Griffiths, H. (2017). Sparsity-driven micro-Doppler feature extraction for dynamic hand gesture recognition. IEEE Transactions on Aerospace and Electronic Systems, 54(2): 655-665. https://doi.org/10.1109/TAES.2017.2761229
- [11] Gupta, S., Jaafar, J., Ahmad, W.W., Bansal, A. (2013). Feature extraction using MFCC. Signal & Image Processing: An International Journal, 4(4): 101-108. https://doi.org/10.5121/sipij.2013.4408
- [12] Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R.P., Tang, J., Liu, H. (2017). Feature selection: A data perspective. ACM Computing Surveys (CSUR), 50(6): 1-45. https://doi.org/10.1145/3136625
- [13] Samadani, A. (2018). EMG channel selection for improved hand gesture classification. 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 4297-4300. https://doi.org/10.1109/EMBC.2018.8513395
- [14] Zhang, H., Wang, J., Sun, Z., Zurada, J.M., Pal, N.R.

(2019). Feature selection for neural networks using group lasso regularization. IEEE Transactions on Knowledge and Data Engineering, 32(4): 659-673. https://doi.org/10.1109/TKDE.2019.2893266

- [15] Yuan, L., Liu, J., Ye, J. (2013). Efficient methods for overlapping group lasso. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(9): 2104-2116. https://doi.org/10.1.1.230.8277
- [16] Cherrington, M., Lu, J., Airehrour, D., Thabtah, F., Xu, Q., Madanian, S. (2019). Feature selection: Multi-source and multi-view data limitations, capabilities and potentials. Presented at 29th International Telecommunication Networks and Applications Conference.
- [17] Zhang, R., Nie, F., Li, X., Wei, X. (2019). Feature selection with multi-view data: A survey. Information Fusion, 50: 158-167. https://doi.org/10.1016/j.inffus.2018.11.019
- [18] Ruiz-Shulcloper, J., Abidi, M.A. (2002). Logical combinatorial pattern recognition: A review. In: Pandalai, S.G. (ed.) Recent Research Developments in Pattern Recognition, Transworld Research Networks, Kerala, India.
- [19] Koller, D., Friedman, N. (2009). The Bayesian Network Representation, in Probabilistic Graphical Models: Principles and Techniques. MIT press, Massachusetts, US.
- [20] Martínez-Trinidad, J.F., Guzmán-Arenas, A. (2001). The logical combinatorial approach to pattern recognition, an overview through selected works. Pattern Recognition, 34(4): 741-751. https://doi.org/10.1016/S0031-3203(00)00027-3
- [21] Gong, R., Chan, T.K. (2006). Syllable alignment: A novel model for phonetic string search. IEICE Transactions on Information and Systems, 89(1): 332-339.
- [22] Team, R.C. (2018). R: A Language and Environment for Statistical Computing. Available: https://www.rproject.org.
- [23] Scutari, M. (2010). Learning Bayesian networks with the bnlearn R Package. Journal of Statistical Software, 35(3): 1-22.
- [24] Scanagatta, M., Salmerón, A., Stella, F. (2019). A survey on Bayesian network structure learning from data. Progress in Artifical Intelligence, 8: 425-439. https://doi.org/10.1007/s13748-019-00194-y
- [25] Ruiz-Shulcloper, J., Guzmán-Arenas, A., Martinez-Trinidad, J.F. (1999). Feature Selection, in Logical Combinatorial Approach to Pattern -Recognition, Mexico, Politécnico press, Mexico City.
- [26] Visser, I., Speekenbrink, M. (2010). depmixS4: An R Package for Hidden Markov Models. J. Stat. Softw., 36(7): 1-21.
- [27] Madeo, R.C.B., Lima, C.A.M., Peres, S.M. (2013). Gesture unit segmentation using support vector machines: Segmenting gestures from rest positions. In Proc. SAC'13. Coimbra, Portugal, 2013: 46-52.
- [28] University of California, School of Information and Computer Science. (2017). UCI machine learning repository. https://archive.ics.uci.edu/ml/citation policy.html.