A Machine Learning based Approach for Gesture Recognition from Inertial Measurements

Giuseppe Belgioioso, Angelo Cenedese, Giuseppe Ilario Cirillo, Francesco Fraccaroli, Gian Antonio Susto

Abstract

The interaction based on gestures has become a prominent approach to interact with electronic devices. In this paper a Machine Learning (ML) based approach to gesture recognition (GR) is illustrated; the proposed tool is freestanding from user, device and device orientation. The tool has been tested on a heterogeneous dataset representative of a typical application of gesture recognition. In the present work two novel ML algorithms based on Sparse Bayesian Learning are tested versus other classification approaches already employed in literature (Support Vector Machine, Relevance Vector Machine, k-Nearest Neighbor, Discriminant Analysis). A second element of novelty is represented by a Principal Component Analysis-based approach, called *Pre-PCA*, that is shown to enhance gesture recognition with heterogeneous working conditions. Feature extraction techniques are also investigated: a Principal Component Analysis based approach is compared to Frame-Based Description methods.

I. Introduction

Last decade's advances in electronics integration and information technologies have revolutionize our capability of interacting with the daily environments, introducing tools and devices that have pervasively become usual instruments to understand and control the human space. Smart appliances have been immersed into the ambient assisted living paradigm, providing solutions for a wide range of scenarios from the home automation for entertainment and comfort, to the assistive domotics for the elderly and the impaired, to the ergonomics management to increase productivity. In this context, the human-machine interaction has concurrently emerged as a relevant issue, to understand the appropriate ways to communicate with the device as much intuitively and naturally as possible [5]. With regards to this, gesture interfaces represent an appealing as well as valid alternative to other more conventional and traditional contact-based modalities.

This problem has been addressed by the scientific community [8] and in the specialized literature many studies can be found that refer in particular to video-based hand gesture recognition [18] and glove-based sensing [3], [14], which are the two technologies that have pioneered the research in the field. More recently, the rapid development of Micro Electrical Mechanical Systems (MEMS) has driven to the production of cheap and compact devices, pushing forward the approach to gesture recognition from inertial measurements provided by accelerometers and/or gyroscopes, which well fits the requirements given by smart environments and the modern trends of ubiquitous computing [7], [17], [19]. Indeed, the gesture recognition techniques based on inertial measurements have reached a certain level of maturity in the video game industry with the design of dedicated controllers but also in the natural gesture interpretation of basic functionalities for smartphones and tablets, and henceforth they can open up novel perspectives in controlling devices for the *smart home*. On the one hand, thanks to the employment of MEMS components, the resulting systems are cheaper than the vision-based ones, they do not need the setup of a dedicated environment, and result often more robust since they are not affected by environmental nuisances (e.g. light changes). On the other hand, though, the gesture description is somehow implicitly encoded in the measurement signals and there may not be a univocal link between the gesture motion and the measured quantities, thus calling for a machine learning (ML) approach for the recognition and classification phase.

Given this context, the problem of gesture recognition can be described as follows: we suppose to perform a sequence of arm movements and to measure the continuous acceleration signal at the hand as a three-dimensional time-series. We want to design a procedure that is able first to detect the beginning of the interesting motion and isolate the gesture instances (*Event Identification phase*), then to cluster together gestures of the same kind and characterize them with a specific signature (*Feature Extraction phase*), and finally to classify a newly measured gesture according to the so-built gesture dictionary (*Classification phase*).

One crucial issue in such a framework is related to the high variability of the input data, in terms of different people performing the action of interest, different device used for measurement, different modality of completing the action task (e.g. body initial position, motion direction, measuring device orientation or reference frame) and indeed the contribution of this work is the development of a gesture recognition algorithm in a context that is *user independent*, *device independent*, and *device orientation independent*. To this aim, classic ML techniques have been studied, such as *k-Nearest Neighbor* (k-NN), *Discriminant Analysis* (DA), *Support Vector Machine* (SVM), *Relevance Vector Machine* (RVM), and more novel approaches based on *Sparse Bayesian Learning* (SBL) are introduced. Moreover, to model the gestures both feature extraction and the analysis in the time domain through signal warping have been considered.

G. Belgioioso, A. Cenedese, G.I. Cirillo, F. Fraccaroli, and G.A. Susto are with the Department of Information Engineering, University of Padova, via Gradenigo 6/B, 35131 Padova, Italy. Corresponding author: G.A. Susto gianantonio.susto@gmail.com.

The remainder of the paper is organized as follows: in Sec. II an overview on the main ML techniques is provided and in Sec. III the SBL methods are introduced and discussed. Then, Sec. IV deals with the Gesture Recognition algorithm that is presented in its main phases, while its experimental validation is provided in Sec. V. Finally, in Sec. VI some concluding remarks are drawn.

II. Classification

Machine Learning (ML) techniques exploit organized data to create mathematical representations (*models*) of an observable phenomenon [6]. It is then possible to rely on such a model to provide predictions for unobserved data. In mathematical terms, let[1]

$$\mathcal{S} = \left\{ x_i \in \mathbb{R}^{1\times p},\ y_i \in \mathbb{D} \right\}_{i=1}^{n} \tag{1}$$

be a *training dataset* [11]. In this formalism, n observations of a certain phenomenon are available: the i-th observation (or *example*) is characterized by p input features, constituting the vector x_i, and a scalar target value y_i that belongs to a certain domain $\mathbb{D}$.

Depending on the domain $\mathbb{D}$, two types of ML problem are identified:

- *regression* - if the output takes continuous values;
- *classification* - if the output space is a set of classes or categories.

In the problem at hand, where we want to discriminate which class a gesture belongs to starting from inertial data, the output belongs to the space of the known gestures: we are therefore dealing with a classification problem. In the following we will indicate the number of different classes in the dataset with the notation N.

In the literature, known classifications algorithms have been adopted to GR problem [17], among which the most promising are those referring to Support Vector Machines (SVMs); in this work we have employed classification techniques (like SVM) that represent the state of the art in GR classification problems and compared these with novel, Sparse Bayesian Learning techniques.

A. Classification Techniques

For the problem at hand the following known classification algorithms have been compared:

1) k-Nearest Neighbors (kNN): probably the simplest approach to classification as it requires only computation of distances between samples. In the k-NN procedure each point of the input space is labeled according to the labels of its k closest neighboring samples (where distances are computed according to a given metric, that is often the Euclidean norm). The only parameter that requires tuning in k-NN is k, the value of the number of samples in the considered neighborhood: the choice of k is usually data-driven (often decided though cross-validation). Larger values of k reduce the noise effect on the classification, but make decision boundaries between classes less distinct [20].

2) Discriminant Analysis (DA): assuming that the conditional probability distributions $P(x|y = j)$ are Gaussian, the DA provides linear (LDA) or quadratic (QDA) combination of features that separates the N classes [6].

3) Support Vector Machines (SVM): SVMs are probably the most popular approach to classification, thanks to their high classification accuracy, even for non-linear problems, and to the availability of optimized algorithms for their computation. SVM are binary classifiers that identify the classification decision boundaries by maximizing the distance, called *margin*, between the two classes. SVM can be considered as a special case of Tikhonov regularization [4]: the objective function of SVM is a trade-off (governed by a parameter C) between margin maximization and penalty on errors in the classification.

Through the *kernel trick* [12] non-linear (in the features space) decision boundaries can be obtained: the most popular choice for kernel transformation is Radial Basis Function (RBF), where the dot products between samples x_i, $< x, x_i >$ in the SVM procedure are replaced by

$$K_{\text{RBF}}(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}, \tag{2}$$

where $\gamma \in \mathbb{R}^+$.

4) Relevance Vector Machines (RVM): the RVM has an identical functional form to the SVM, but provides probabilistic classification. It should be noted that RVM solutions are more parsimonious than SVM ones and therefore are more adapt to be embedded in devices with limited memory capabilities; on the other hand, the training phase for RVM is usually slower and less numerically stable w.r.t. the SVM. In particular, [15] exploits Bayesian inference to obtain sparse solutions.

B. Tuning and Cross-Validation

Generally, ML algorithms require the tuning of parameters or, more generally of some criteria, upon which the model accuracy strongly depends, e.g. k-NN performance strongly depends on the choice of k, the neighbourhood size. The list of the 'degrees of freedom' (DoFs) in design of the considered classification algorithms is contained in Table I; regarding the

[1]The formulation in Eq. (1) describes ML *supervised problem*, given the availability of labeled data, i.e. presence of output in the training data.

Algorithm	Design DoFs
k-NN	k Metric Distance Distance
DA	Type (Linear or Quadratic)
SVM	Regularization Parameter C γ (RBF Kernel)
RVM	γ (RBF Kernel)

TABLE I

EMPLOYED CLASSIFICATION APPROACHES AND DESIGN DOFS

design of k-NN, in this work the Euclidean norm has been employed in the distance calculation, while a complete search algorithm has been employed for the neighbours search.

To find the optimal values for the parameters listed in Table I it has been tested the performance of the algorithms at the changes of the parameters: in the case of discrete parameters, like k in k-NN, a complete search is feasible, while in the case of continuous values an *iterative grid search* has been employed: the approach consists on computing algorithm performance on a grid of values, then repeating the computation on a new grid where the best performance has been obtained on the previous search; this procedure is iterated until the performance improvement is not significative anymore.

In order to assess classification performance in an unbiased way, two *cross-validation* approaches have been employed

1) $CV sim$-fold: : the n samples are randomly partitioned into n equal size subsets; one of the subset is used for validation and the other $k-1$ for training. The cross-validation process is then repeated k times, with each of the k subsets used exactly once for validation.

2) Repeated Random Sub-Sampling Validation: [10], also known as *Monte Carlo crossvalidation (MCCV)*: Q simulations are done by randomly splitting the n observations into a training dataset of $n_{\text{TR}} = \lfloor nq \rfloor$ maintenance cycles and a validation dataset of $n_{\text{VL}} = \lceil n(1-q) \rceil$ maintenance cycle, with $0 < q < 1$. It has been shown [13] that MCCV is asymptotically consistent resulting in more pessimistic predictions of the test data compared with full crossvalidation.

In this work we have employed *stratified* crossvalidation: training and validation sets have the same proportion of data from the same class to avoid skewness problem in the classification.

The performance indicator employed in crossvalidation is the Missclassification Rate (MCR):

$$\text{MCR}[\%] = \text{Percentage of missclassified samples.}$$

III. RVM-BASED METHODS

In this Section we present two novel classification algorithms based on Sparse Bayesian Learning (SBL) and RVM that aims at providing parsimonious solutions to meet the requirement of the application in exam.[2]

A. Frame-based Descriptor Multi Kernel RVM (RVM-MK)

The SBL approach presented in this subsection is inspired by *Frame-based Descriptor* (FD) presented in [17] for dealing with GR with input data streams and exploits SBL to provide parsimonious solutions; the proposed approach can be considered a sort of *Multi Kernel RVM* [16].

We detail in the following the FD approach for GR; data streams X related to the three Cartesian frame acceleration axes (a_x, a_y and a_z) are divided into $M+1$ contiguous portions of equal length[3]; every contiguous couple of data portions, for a total of M, constitutes a *frame* and for each j-th frame the following quantities are computed:

- μ_j: the continuous component of the Discrete Fourier Transform (DFT), representing the mean;
- ε_j: energy (without the contribution of the DFT continuous component);
- δ_j: entropy (without the contribution of the DFT continuous component);
- σ_j: standard deviation;
- ρ_j: axis correlation.

The quantities μ, ε, δ and σ are computed along each of the 3 axis, while the last for each axis couple, for a total of $4 * 3 + 3 = 15$ features per frame. The M number is a design parameter and it is determined from cross-validation procedure. The features are grouped in the vector $x = [\mu\ \varepsilon\ \delta\ \sigma\ \rho]$, where μ is a vector containing the values μ_j for the M frames and similar definition stand for the other elements of x.

[2]For more details on SBL the interested readers are referred to [2].

[3]In the case data streams length is not a multiple of $M+1$, the last portions have a sample more than the first ones in order to exploit all the available data

In SBL, the candidate function to be estimated with respect to the input signal x may be expressed (as in the SVM approach) as follows [2]

$$z = \sum_{i=1}^{n} w_i \phi_i(x), \tag{3}$$

where w_i are coefficients estimated through Bayesian inference from the data; the *basis functions* ϕ_i are chosen a-priori; in our approach (3) is defined as

$$z = \sum_{i=1}^{n} \sum_{j=1}^{5} w_{i,j} e^{-\gamma \|P_j(x-x_i)\|^2} + w_0, \tag{4}$$

where $\gamma \in \mathbb{R}^+$ and the term w_0 is an additional bias included to help the classification; moreover,

$$P_1 = \text{diag}(I, 0, 0, 0, 0) \tag{5}$$

and analogously P_j has just the j-th diagonal block non-zero and equals to the identity. The features are clustered in groups of the same nature, with the possibility to give different weights to the various statistics: this can lead to the discard of groups of irrelevant statistics or provides hints on the most significative statistics.

B. RVM con Dynamic Time Warping (RVM-DTW)

The second novel approach, described in this subsection, is inspired by classic RBF kernel (2), that may be exploited as a tool to normalize (between 0 and 1) the Euclidean distance, making it a similarity index: with this approach, without requiring features, a norm between signals is considered instead.

In this work, we have exploited *Dynamic Time Warping* (DTW), that, even if it is not a mathematical norm, but a similarity factor for signals, it is widely adopted in vocal and gesture recognition problems, used together with ML algorithms and by itself [1], [7], [9]. Briefly, DTW is a technique to find an optimal alignment between two sequences of data, and in the following an introduction is provided.

Given two temporal sequences $u = (u_1, u_2, \ldots, u_U)$ and $v = (v_1, v_2, \ldots, v_U)$, we introduce a cost function on the single couples of samples $c(u_i, v_j)$; it is defined as a *Warping Path* (WP) a sequence of indices couples $p_l = (i_l, j_l)$ with $i_l \in [1 : U]$, $j_l \in [1 : V]$ e $l \in [1 : L]$, subject to the following conditions:

1) *Boundary conditions*:
   $p_1 = (1, 1)$ e $p_L = (U, V)$,
2) *Step-size conditions*:
   $p_{l+1} - p_l \in \{(1, 0), (0, 1), (1, 1)\}$.

The cost of a WP p is defined as

$$C_p(u, v) = \sum_{l=1}^{L} c(u_{i_l}, v_{j_l}), \tag{6}$$

while the DTW distance of two temporal sequences u and u in a WP p is defined as:

$$D_{\text{DTW}}(u, u) = min\{C_p(u, v)\}. \tag{7}$$

In the RVM-DTW approach, equation (3) becomes:

$$z = \sum_{i=1}^{n} w_i e^{-\gamma D_{\text{DTW}}(X, X_i)} + w_0, \tag{8}$$

where X is the (original) measured signal.

IV. GR Algorithm

As introduced in the Section I, the GR algorithm consists of 3 main phases: Event Identification, Data Pre-processing/Feature Extraction and Classification; in the following the first two phases are discussed and the PrePCA filtering is presented.

A. Events Identification

For event identification, a threshold approach has been employed: the data stream is divided into fixed length windows, where each window shares with the contiguous a certain amount of samples; for each window and for each axis the following parameter is computed

(i) sum of the sample variance,

while these are computed on the module

(ii) sample variance,

(iii) energy of frequencies band

$$\varepsilon = \frac{1}{T}\sum_{i=h}^{h+t} |d_i|^2, \tag{9}$$

(iv) spectral entropy

$$\delta = -\sum_{i=h}^{h+t} b_i \log(b_i) \qquad \text{with } b_i = \frac{|d_i|}{\sum\limits_{j=h}^{h+t} |d_j|}, \tag{10}$$

where d_i are the DFT coefficients[4], T is the temporal window length with h and t selecting the interest band in a way that continuous component and eventual noise at high frequencies are discarded. The parameters listed above are monitored: the start of a gesture is flagged when a certain condition on the parameters and on pre-defined thresholds is met; at the same way, the gesture is considered finished when the above condition is no longer met.

In the problem at hand, several experiments have lead to choose $T = 20$ with an overlap of 10 with the contiguous windows while the best choice for event detection condition has been a logical OR between the passing of two thresholds on the energy and the sum of sample variance; the thresholds values were respectively 0.002 for the sample variance sum and 0.025 for ε: the tuning has of course shown an important dependency of these values to the device in exam.

B. Data Pre-processing/Feature Extraction

Before any preprocessing operations, every sample in the data streams is replaced with the mean of the two adjacent samples to remove noise and movements involuntarily committed by the user; the described operation is absolutely feasible in GR application: in the case at hand we have employed a 60 ms window that corresponds to a 20 ms delay, which is negligible in practical usages.

As seen in Section II the various classification approaches (besides the RVM-DTW) require a fixed amount of features to represent the gestures: this is quite tricky in the case of streams of data (motivating the RVM-DTW approach) where events have generally different lengths. Moreover the raw data is much more informative than what is required by the classification to discriminate clusters of gestures: this is not only a computational issue, but also affects the accuracy of the classification, since excessive unnecessary information may mask the informative content of the signals. For these reasons it is more convenient to create a reduced set of features, possibly relevant to the classification problem. In this respect, two techniques for features extraction have been employed: Principal Component Analysis (PCA) and FD (with the same approach and features described in III-A).

PCA is a geometrical approach that allows to find an orthonormal basis for a matrix D, in which the components are ordered in descendant order of variability explained of the original matrix: the elements of the new computed basis are in fact eigenvectors for the covariance matrix of D and the diagonal elements of the new basis are ordered for decreasing magnitude order. With this transformation the first Principal Components (PCs), the eigenvectors, are those with more informative contents; the PCA-based feature extraction exploits this principle: the original signal is substituted with its projection along the first l PCs to focus on the most informative part of the signal. While several approaches may be adopted in the PCA design, in this work we have treated the number l of PCs included in the projection as a design parameter.

C. Pre-PCA and device orientation independency

One of the main goal of this work is to provide a tool that is device orientation-independent; in order to get rid of orientation related differences in the extracted features, the inertial signals should be projected onto the same coordinate system but with the available information this is not feasible.

For this reason, we introduce a procedure, called *PrePCA*, that considers a common 'basis' that is defined just by the signal itself; the method is based on the non-centered version of PCA, where before estimating the covariance matrix, the mean is not subtracted, which corresponds to considering the signal energy instead of the variance. As it will be shown in the Experimental Section, through the PrePCA the classification performances are strongly enhanced: it has been observed

[4]DFT coefficients are computed in our work with a Fast Fourier Transform of 32 samples, employing a Hamming window

(a) o_n (b) o_a

Fig. 1. Normal device orientation (o_n) with the display pointing upwards and alternative orientation (o_a) with the display pointing pointing to the side (both seen from above).

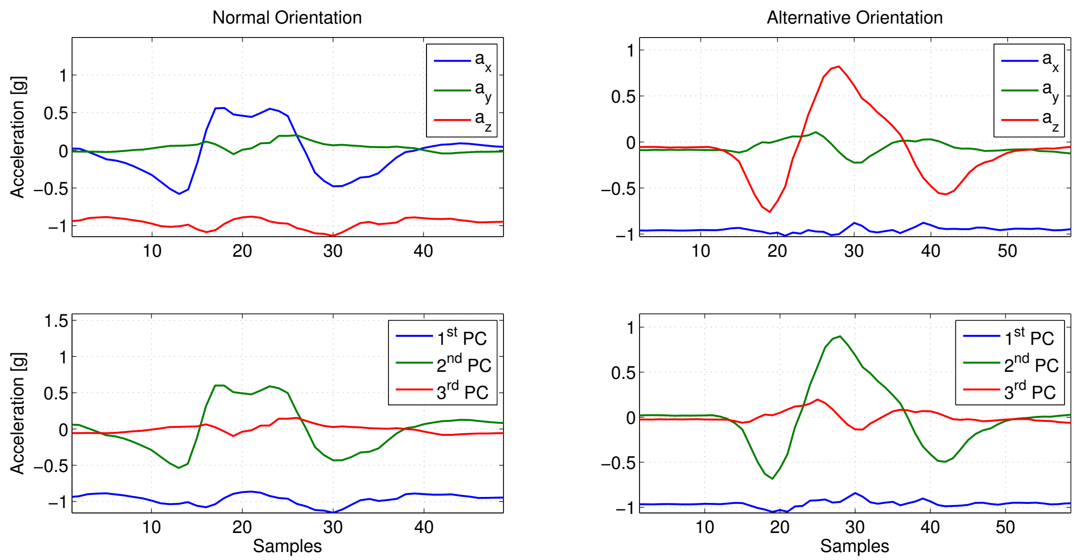

Fig. 2. First 3 PrePCA projections (lower panels) for two examples of gestures taken with different device orientation (top panels).

that generally the first PC is aligned with the gravity component of the signal, by proving the robustness of the PrePCA to device orientation.

In Fig. 2 an example of the PrePCA basis projection with different device orientation is reported: normal (o_n) or alternative (o_a) as depicted in Fig. 1; it can be clearly appreciated how the PrePCA can facilitate the classification problem in case of different orientation settings.

Since a 3-axis system generates 8 possible bases, we impose some conditions on the PrePCA: we assume that the first PC captures the gravity effect and we impose that the gravity effect is always negative; a second assumption is that the resulting coordinate system is right-handed. The imposed assumptions make two possible bases available: we consider both and we indicate them with *PrePCA v1* and *PrePCA v2*.

V. Experimental Results

A. Dataset Description

For the data collection two different devices have been employed: an Ipod-touch and a HTC-Explorer smartphone, each one equipped with a different inertial data retrieval application, respectively Sensor Stream and Accelerometer Monitor, both freely downloadable. For both devices the sampling frequency has been set to 50Hz and the gravity has not been compensated.

The dataset employed has been constructed with the help of 31 persons and the collected gestures are $n = 550$ divided into $N = 4$ types of gestures: verticals (v), horizontals (h), circles (c) and eights (e) as depicted in Fig. 3. As discussed in Section IV-C two different device orientation have been considered (normal o_n and alternative o_a, see Fig. 1). Each person has performed the gestures being free to choose the device orientation, direction of the gesture and starting point to have a representative dataset of real GR application.

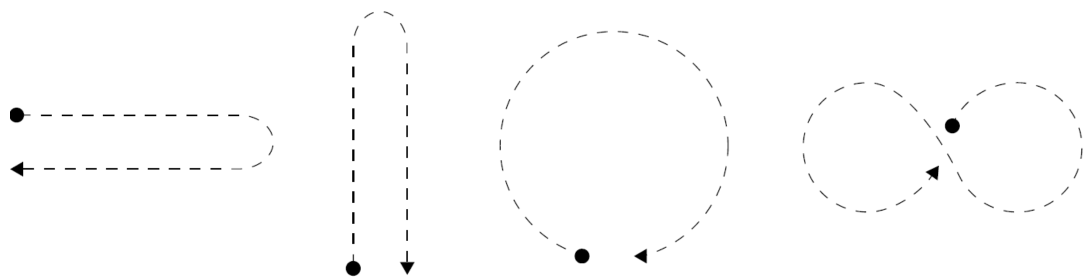

Fig. 3. Frontal view of the $N = 4$ gestures types collected in the dataset: horizontal (h), vertical (v), circles (c) and eight (e).

Algorithm	Feature Extraction	Optimal Values
k-NN	PCA	$k = 1$, $l = 14$
	FD	$k = 1$, $M = 2$
DA	PCA	Pseudo Quadratic DA, $l = 14$
	FD	Linear DA, $M = 2$
SVM	PCA	$\gamma = 2.2313$, $C = 5.5546$, $l = 16$
	FD	$\gamma = 0.25$, $C = 11.7579$, $M = 2$
RVM	PCA	$\gamma = 3.8715$, $l = 16$
	FD	$\gamma = 0.46294$, $M = 2$

TABLE II

EXPERIMENT A: TUNING OPTIMAL VALUES

B. Experiment A: PCA vs FD

In this first experiment we test the two feature extraction approaches: PCA and FD; this has been done on a reduced dataset regarding just the iPod Touch data with normal device orientation for a total of 145 observations.

In Table II are reported the optimal values for DA, k-NN, SVM and RVM obtained with the tuning procedure: the results reported in the following are obtained with the reported values. Fig. 4 reports the correct classification percentages (100% minus the MCR) obtained by the classifiers with the two features extraction approaches (averaged results over 5-fold CV); it can be appreciated how the FD based approach is consistently better than the PCA based: following these results, in the following experiments FD has been employed for features extraction.

C. Experiment B: PrePCA

Then, the full dataset have been considered (both devices with the two orientations). The goal of this experiment is to test the efficiency of the PrePCA filtering (in the two versions) and to compare the classification algorithms performances.

Classifiers have been tuned again and the optimal values have been reported in Table III: it can be appreciated how these optimal values are quite similar to those obtained in Experiment A when PrePCA is employed, while more significant variations on the values are in place when the PrePCA is not employed; this suggests how PrePCA may actually help to cope with different device orientations.

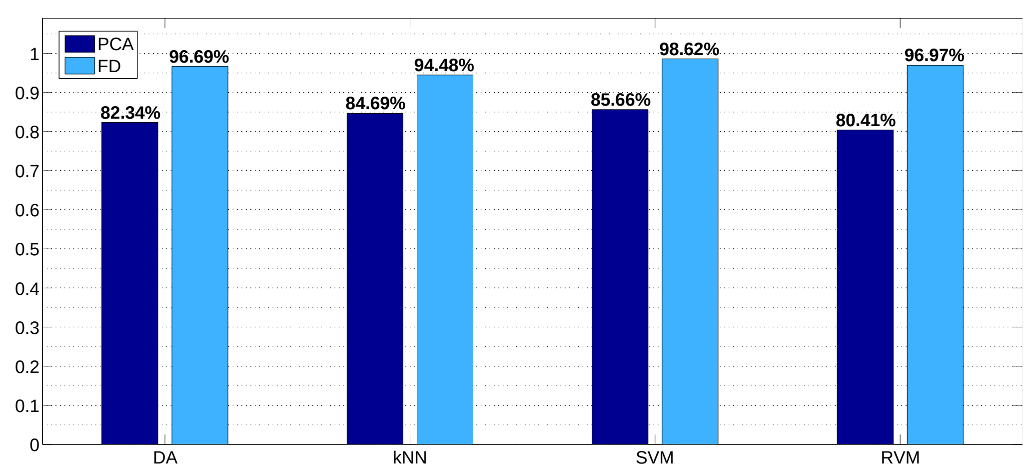

Fig. 4. Correct classifications (percentage values) for DA, k-NN, SVM and RVM with PCA and FD based features extraction FD (averaged results over 5-fold cross-validation (CV)).

Algorithm	PrePCA	Optimal Values
k-NN	no PrePCA	$k = 1$, $M = 4$
	PrePCA v1	$k = 1$, $M = 5$
	PrePCA v2	$k = 1$, $M = 5$
DA	no PrePCA	Pseudo Quadratic DA, $M = 4$
	PrePCA v1	Pseudo Linear, $M = 8$
	PrePCA v2	Pseudo Linear, $M = 5$
SVM	no PrePCA	$\gamma = 0.7423$, $C = 6.4831$, $M = 4$
	PrePCA v1	$\gamma = 0.2199$, $C = 7.6649$, $M = 4$
	PrePCA v2	$\gamma = 0.2143$, $C = 5.3508$, $M = 5$
RVM	no PrePCA	$\gamma = 0.63$, $M = 4$
	PrePCA v1	$\gamma = 0.3402$, $M = 3$
	PrePCA v2	$\gamma = 0.1118$, $M = 4$
RVM-MK	no PrePCA	$\gamma = 11.3622$, $M = 3$
	PrePCA v1	$\gamma = 0.3276$, $M = 4$
	PrePCA v2	$\gamma = 0.3177$, $M = 4$
RVM-DTW	no PrePCA	$\gamma = 0.0625$
	PrePCA v1	$\gamma = 0.0412$
	PrePCA v2	$\gamma = 0.04$

TABLE III
EXPERIMENT B: TUNING OPTIMAL VALUES

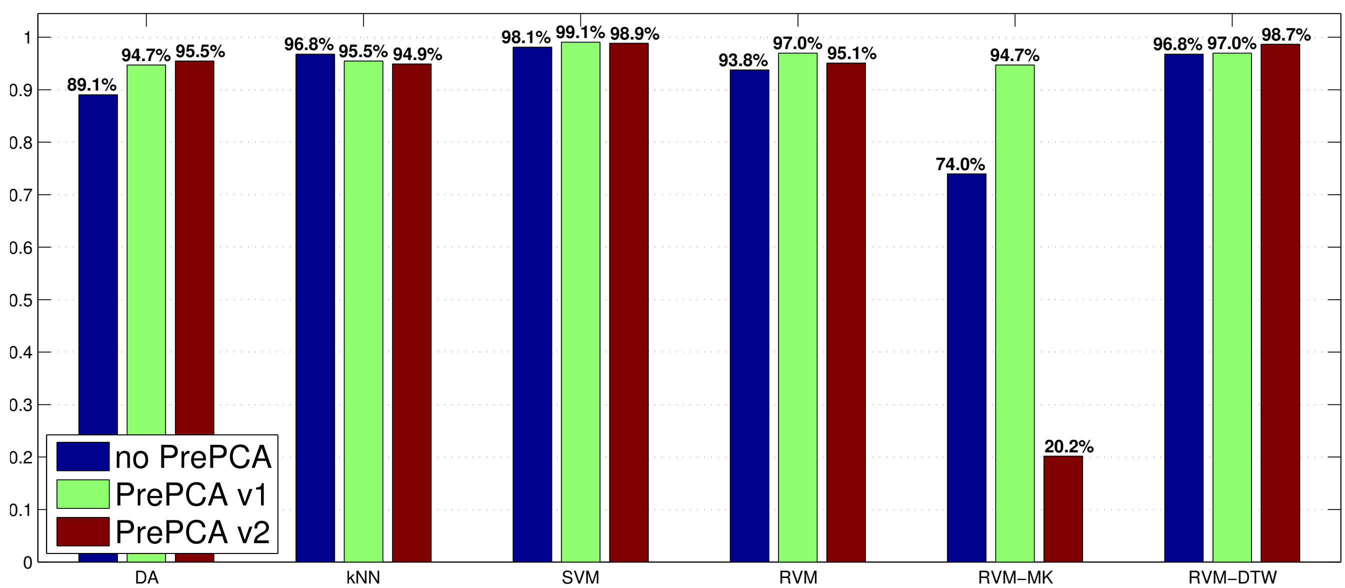

Fig. 5. Correct classifications (percentage values) for the classifiers with no PrePCA, PrePCA v1, PrePCA v2 (averaged results over 4-fold cross-validation (CV)) FD.

Once the parameters have been tuned, the classifiers performances are computed through 4-fold CV: the results are reported in Fig. 5. It can be highlighted how the employment of PrePCA (in both versions) improves the classifiers performances, beside for k-NN: it is particularly interesting to see how PrePCA v1 improves RVM-MK of over 20%, while in the PrePCA v2 version the results are even worse than the one obtained with no PrePCA.

VI. CONCLUSIONS

In this paper, a tool for GR based on ML algorithms has been presented. The tool has been tested on an heterogeneous dataset composed of gestures done by many users, with 2 different devices and different device orientations.

The first issue faced in this work was the detection of gestures within streams of data: the proposed solution based on thresholds should be set based on the device in exam; the second issue was to have a tool independent from user, device and device orientation: the filtering procedure proposed, called PrePCA, has been proved to be particularly effective in improving classifications performances.

From the analysis of the various algorithm accuracy, SVM has proved to be the approach that better performs in all considered scenarios; on the other hand, SBL based approaches present comparable classification performances with less computational cost required, which represents an extremely important quality for the application at hand.

References

[1] A. Akl, C. Feng, and S. Valaee. A novel accelerometer-based gesture recognition system. *Signal Processing, IEEE Transactions on*, 59(12):6197–6205, 2011.

[2] C.M. Bishop and M.E. Tipping. Bayesian regression and classification. *Nato Science Series Sub Series III Computer and Systems Sciences*, 190:267–288, 2003.

[3] L. Dipietro, A.M. Sabatini, and P. Dario. A survey of glove-based systems and their applications. *Systems, Man, and Cybernetics, Part C, IEEE Transactions on*, 38(4):461–482, 2008.

[4] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13(1):1–50, 2000.

[5] L. Gamberini, A. Spagnolli, L. Prontu, S. Furlan, F. Martino, B.R. Solaz, M. Alca niz, and J.A. Lozano. How natural is a natural interface? An evaluation procedure based on action breakdowns. *Personal Ubiquitous Comput.*, 17(1):69–79, 2013.

[6] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learnin*. New York: Springer, 2009.

[7] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan. uwave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing*, 5(6):657–675, 2009.

[8] S. Mitra and T. Acharya. Gesture recognition: A survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(3):311–324, 2007.

[9] M. Müller. *Information retrieval for music and motion*. 2007.

[10] R.R. Picard and R.D. Cook. Cross-validation of regression models. *Journal of the American Statistical Association*, 79:575–583, 1984.

[11] A. Schirru, G.A. Susto, S. Pampuri, and S. McLoone. Learning from time series: Supervised aggregative feature extraction. In *Decision and Control (CDC), IEEE Conference on*, pages 5254–5259. IEEE, 2012.

[12] B. Scholkopf. The kernel trick for distances. *Advances in neural information processing systems*, 13:301–307, 2001.

[13] J. Shao. Linear model selection by cross-validation. *Journal of the American Statistical Association*, 88:486–494, 1993.

[14] D. Sturman and D. Zeltzer. A survey of glove-based input. *IEEE Comput. Graph. Appl.*, 14(1):30–39, January 1994.

[15] M.E. Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

[16] D.G. Tzikas, L. Wei, A. Likas, Y. Yang, and N.P. Galatsanos. A tutorial on relevance vector machines for regression and classification with applications. *EURASIP News Letter, 17, 2, 4*, 23:1–25, 2006.

[17] J. Wu, G. Pan, D. Zhang, G. Qi, and S. Li. Gesture recognition with a 3-d accelerometer. In *Ubiquitous intelligence and computing*, pages 25–38. Springer, 2009.

[18] Y. Wu and T.S. Huang. Vision-based gesture recognition: A review. In *International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction*, 1999.

[19] R. Xu, S. Zhou, and W.J. Li. Mems accelerometer based nonspecific-user hand gesture recognition. *Sensors Journal, IEEE*, 12(5):1166–1173, May 2012.

[20] M.-L. Zhang and Z.-H. Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7):2038–2048, 2007.