A GRAPH BASED METHOD FOR TIMED UP & GO TEST QUALIFICATION USING INERTIAL SENSORS

Pierre Jallon, Benjamin Dupre, Michel Antonakios
CEA LETI - MINATEC Campus
Grenoble, France
contact: pierre.jallon@cea.fr

ABSTRACT
A graph based classifier is proposed to recognize the different time phases of the up & go test based on signals collected by an inertial sensor set on a person chest. This test being a sequential set of actions, a graph is used to model it and enforce the classification algorithm to estimate a solution with this constraint. The graph is described by a Markov chain \( A(m) \). Based on the hidden Markov model theoretical framework which by construction fits with this kind of modelling, the proposed method extends this framework to other classifiers: Bayes, LDA and SVM are discussed in this paper. These classifiers and their graph enforced versions are applied and their results compared to the analysis of the timed up & go test to recognize its different phases.

Index Terms— Classifiers, graph based method, HMM, LDA, SVM

1. INTRODUCTION
Inertial MEMs-based sensors (accelerometers, magnetometers, ...) have met a large success for motion related biomedical applications recently. Physical activity estimation for people suffering of obesity and related diseases and sportman monitoring themselves are probably the most studied applications [1]. [2] describes a method to assess gait activity of hemiplegic patients during ecological conditions; [3, 4] have developed systems for epileptic seizures characterization and detection, etc...

In this paper, the application of interest is the timed up & go (TUG) test for elderly persons. This test is a very simple technique for evaluating a person’s functional mobility, in particular to predict its ability to go outside alone safely [5]. Sit on a chair at 3 meters of a wall at the beginning of the test, the person has to do the following actions: 1. Stand up, 2. Walk to the wall, 3. Turn around, 4. Walk back to the chair, 5. Turn around, 6. Sit down. The total time to complete all these steps is used to predict the patient’s risk of falling. However, each activity has some interest from a medical point of view. The evaluation is currently done by a therapist by sight since no system is available to split and analyze the different phases of the test. Such systems are currently under development: in the CIU Sante project which funds this work or by [6] which addresses the signal processing problem with a heuristic approach.

In this paper, a graph-enforced classification algorithm of inertial sensors signals is presented to estimate the different phases of the TUG test. If the discrete variable \( A(m) \) represents the practiced activity at time index \( m \) (i.e. \( A(m) = i \) if the activity \( #i \) is practiced at time index \( m \)), a classifier without graph enforcement estimates \( \forall m \in \{0, M - 1\}, A(m) \) in an independent way based on the sensor signals and on the classification technique. Using a graph, \( A(m) \) is no more modelled as an independent sequence but as a Markov chain to set some constraints between activities of interest. These connections are referred in the following as the graph.

For the TUG test, the graph is used to describe the time structure of the test and to perform a classification which fits it. This kind of method can be extended to any test based on a protocol which imposes a time structure to the activities of interest. It can also be extended to applications for which activities are not as constrained, but are time stable. For these latter applications, the graph is used to stabilize the classification result setting up the probability to remain into a given state close to 1: [7] used it for physical activity estimation, [2] for gait activity assessment.

Hidden Markov model (HMM) [8] offers an adapted theoretical framework for graph based classifiers. The sequence of interest \( A(m) \) is the unobserved sequence, and if the signals characteristics used to perform classification are concatenated into a vector \( O(m) \) for the time index \( m \), link to the activities with probability density function \( p(O(m)|A(m)) \), under some assumptions about the statistical properties of \( O(m) \) detailed in section 2, the couple \( \{A, O\} \) is a so-called HMM. Starting from this theoretical framework, an extension to other classifiers is proposed in this paper and applied to Bayesian, linear discriminant analysis (LDA) [9] and support...
vector machines (SVM) [10] classifiers.

The paper is structured as follow: in section 2, the theoretical framework of graph based method is presented. HMM are first introduced and the graph based classification algorithm is developed. Bayesian, LDA and SVM classifiers adaptation are discussed in section 3. Finally, the TUG analysis details and results are presented in section 4.

2. HIDDEN MARKOV MODELS AND GRAPH BASED METHOD

2.1. Hidden Markov models

A hidden Markov model is a bivariate stochastic process. The first process \( A(m) \) is a Markov chain and is not observed. This process is fully described by the hidden variable \( A(m) \) and initialisation probabilities \( \pi_i = p(A(0) = i) \). For graph based method, this process is used to model the activities of interest and their dependencies. In particular, if an activity \( i \) can not occur just after another activity \( j \), it can be described in the algorithm setting \( a_{i,j} = 0 \). If an activity \( i \) is stable in time, the probability \( a_{i,i} \) can be set close to one: \( a_{i,i} = 1 - \epsilon \).

The second process \( O(m) \) is the so-called observation process. In general, \( O(m) \) is the concatenation of characteristics extracted from sensors signals. This process is assumed to be independent conditionally to \( A(m) \) and defined by the following probability density functions:

\[
\forall i, p(O(m) | A(m) = i) .
\]

In a classification context where the class of interest are represented by the value taken by the hidden variable \( A(m) \) and given an observation sequence \( O(0 : M - 1) \) of time length \( M \), the classification algorithm objective is to estimate the corresponding most likely sequence of activities \( A(0 : M - 1) \):

\[
\arg \max_{A(0:M-1)} p(A(0 : M - 1) | O(0 : M - 1))
\]

\[
= \arg \max_{A(0:M-1)} p(A(0 : M - 1), O(0 : M - 1)) \quad (1)
\]

This optimisation problem is well known to be solved using the Viterbi algorithm [11]. HMM are hence naturally adapted to classification problem with graph enforcement. In the next section, the extension to other classifiers is discussed.

2.2. Graph based methods

The r.h.s. of (1), \( p(A(0 : M - 1), O(0 : M - 1)) \) rewrites using the properties of HMM:

\[
p(A(0)) \prod_{m=1}^{M-1} p(A(m) | A(m-1)) \times \prod_{m=0}^{M-1} p(O(m) | A(m))
\]

The first r.h.s. of this equation refers to the graph and the second one links observation and graph. This second term can hence be replaced by any function \( \phi(O(0 : M - 1)|A(m)) \) which is a measurement of belief of the different activities of interest, i.e. with the following properties:

A1. \( \forall i, \phi(O(0 : M - 1)|A(m) = i) \) is a positive function

A2. \( \forall i, \phi(O(0 : M - 1)|A(m) = i) \) grows with the belief in the assumption \( A(m) = i \)

The sequence of activities is then estimated as:

\[
\hat{A}(0 : M - 1) = \arg \max_{A(0:M-1)} \Phi(A(0 : M - 1), O(0 : M - 1))
\]

where \( \Phi(A(0 : M - 1), O(0 : M - 1)) \) is defined as:

\[
p(A(0)) \prod_{m=1}^{M-1} p(A(m) | A(m-1)) \times \prod_{m=0}^{M-1} \phi(O(0 : M - 1)|A(m))
\]

The adaptation of Viterbi’s algorithm to this optimisation problem is straightforward: \( p(O(m)|A(m)) \) is simply replaced by \( \phi(O(0 : M - 1)|A(m)) \).

In the next section, example of functions \( \phi \) are derived.

3. CHOICE OF \( \phi \) FUNCTIONS

3.1. Bayes based \( \phi \) functions

The first kind of \( \phi \) functions are Bayesian ones: observations are linked to activities through probability density functions \( \forall i, p(O(m)|A = i) \). The simplest case is \( \phi(O(0 : M - 1)|A(m)) = p(O(m)|A(m) = i) \) and coincides with HMM. However, a Bayesian classifier using only 1 sample is known to have poor performances and generally a time window of analysis is used. For this case, the following classification function can be used:

\[
\phi(O|A(m)) = \phi^{(2T+1)}_{Bayes}(O|A(m)) = \left( \prod_{t=-T}^{T} p(O(m+t)|A) \right)^{1/2T+1}
\]

It is straightforward to check that this function satisfies both assumptions A1 and A2.

3.2. LDA and SVM based \( \phi \) functions

To construct \( \phi \) functions for LDA and SVM classifiers, binary classification problem is first considered. Extension to multi-class problem is then discussed. For the binary classification problem, both methods LDA and SVM split the two classes
of interest with a hyperplane, which equation is estimated according to the used technique. If \( \mathbf{w} \) is the hyperplane normal with unit norm, \( b \) an offset, and \( O(m) \) the observation vector value to classify, the decision is done according the sign of the following criterion:

\[
J(m) = \mathbf{w}^T O(m) + b
\]

where \( ^T \) stands for the transpose operator. For kernel-based SVM, the scalar product is replaced by a kernel function \([10]\).

The sign of \( J(m) \) indicates which of the two classes to select, and \( |J(m)| \) the belief in the decision. If \( J(m) > 0 \) for assumption \( \#0 \), the following function \( \phi_{0,1} \) can be used for the binary case:

\[
\begin{align*}
\phi_{0,1}(O(m)|A = 0) &= 1/2(\tanh(1/2J(m)) + 1) \\
\phi_{0,1}(O(m)|A = 1) &= 1/2(\tanh(-1/2J(m)) + 1)
\end{align*}
\]

\( \phi_{0,1} \) satisfies the following properties:

- \( \phi_{0,1}(O(m)|A = 1) = 1 - \phi_{0,1}(O(m)|A = 0) \)
- \( 0 \leq \phi_{0,1}(O(m)|A = 0) \leq 1 \)
- \( \phi_{0,1}(O(m)|A) \) grows with the belief in \( A \). In particular if \( J(m) \to \infty \) (resp. \( -\infty \)), \( \phi_{0,1}(O(m)|A = 0) \to 1 \) (resp. \( \phi_{0,1}(O(m)|A = 1) \to 1 \)),

and hence both assumptions A1 and A2.

The multiclass case extends the all versus all technique. This method consists in performing binary comparisons for each couple of classes, and to allocate a vote to the classifier result. The chosen class is the one with the maximal number of votes. The \( \phi \) function design is based on this method: For each couple of classes \( i, j \) (\( j > i \)) two values \( \phi_{i,j}(O(m)|A = i) \) and \( \phi_{i,j}(O(m)|A = j) \) are computed. A belief indicator for class \( \#i \) is the following one:

\[
\tilde{\phi}_i(O(m)|A = i) = \sum_{k<i} \phi_{k,i}(O(m)|A = i) + \sum_{j>i} \phi_{i,j}(O(m)|A = i)
\]

The function \( \phi \) is computed as follow:

\[
\forall i, \phi(O(m)|A = i) = \frac{\tilde{\phi}_i(O(m)|A = i)}{\sum_j \phi_{j,i}(O(m)|A = j)} \tag{2}
\]

\( \phi \) also satisfy assumptions A1. Assumption A2 has not been proved in the general case. However, if the belief grows in one assumption \( \#i \) more than for any other one, i.e. \( \forall j \neq i, \Delta \phi_i(O(m)|A = i) > \Delta \phi_i(O(m)|A = j) \), the assumption is satisfied, i.e. \( \phi(O(m)|A = i) \) increases.

4. APPLICATION TO CLASSIFICATION OF TUG SIGNALS

The presented results are now applied to classification of TUG signals and aims to identify each part of the test. The section is split in three parts: the first one is about the inertial sensors signals preprocessing. The validation process and the performance measurement is presented in the second section. And the classification results are discussed in a last section.

4.1. Inertial sensors signals preprocessing and annotation signal

The TUG database has been recorded using a triaxial accelerometer and magnetometer set on a person’s chest. The sensor works at 200 Hz, and at each time index a \( 6 - D \) vector is collected:

\[
s(m) = [\mathbf{a}(m)^T \mathbf{m}(m)^T]^T
\]

The first component \( \mathbf{a}(m) \) corresponds to the 3D accelerometer signal. In quasi-static phases, it indicates the sensor orientation with respect to the floor. The second term \( \mathbf{m}(m) \) correspond to the 3D magnetometer signal and in environment with no electromagnetic perturbation, it indicates the sensor orientation with respect to the earth electromagnetic field (the north).

This signal is first low pass filtered to reduce signal’s perturbation level. Sensor orientation is then estimated from \( s(m) \) using yaw/pitch/roll angles and, as a reference, sensor’s orientation when the person is sat on its chair [12]. The observation vector used to perform classification is hence a \( 3 - D \) vector of 3 angles.

\( K = 19 \) acquisitions have been recorded during a clinical campaign. For each one of them, an annotation signal - describing the different test phases detailed in the introduction section - has been hand made during the acquisition campaign. This signal takes discrete values from 0 to 5 according to the phases number. The length of k-th signal is \( M_k \) and the taken values are referred as \( A_k(m) \).

4.2. Validation process

The graph-based classifier has been used on this database using the following function \( \phi \):

- Bayesian \( \phi \) for several values of \( T \). The observation vector p.d.f is modelled as a sum of 2 Gaussian laws.
- LDA
- SVM without and with kernel

SVM algorithm has a parameter (generally denoted as \( C \)) which indicates a trade off between a small margin and the error penalty for the training part. Several values of \( C \) have
been tested: \{0.5, 1, 1.5, 2\}. For the kernel case, the radial basis function kernel has been used. It is defined as:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$$

and depends on the parameter \(\gamma\). \(\gamma = 1\) and \(\gamma = 2\) have been tested.

For each function \(\phi\), a cross validation method is used: \(\phi\) parameters are learnt on 18 sequences, and the classifier is tested on the last sequence. This operation is done \(K = 19\) times and the performance is evaluated for sequence \(\# k\) using the criterion:

$$J_k = \frac{1}{M_k} \sum_{m=0}^{M_k-1} \delta(\hat{A}_k(m) - A_k(m))$$

where \(\delta(n)\) equals 1 if \(k = 0\) and 0 otherwise. The classifier global performance is evaluated using the averaged good detection rate on the test sequences: \(J = \frac{1}{K} \sum_{k=0}^{K-1} J_k\).

### 4.3. Classification results

The classification results are shown on figures 1 and 2. In both cases, the different classifiers performance (Bayes, LDA and SVM), with and without graph enforcement, are compared.

On figure 1, section 3.1 results are presented on the left figure: classification performance is given for functions \(\phi^{Bayes}\) for several values of \(T\). On the right figure, the classification results for several set of parameters of the SVM classifier have been represented. In both cases, the graph enforced classifier has an averaged good detection at least 15\% higher than the classic classifier one.

On figure 2, a global snapshot of the different classifiers performance is presented: classical HMM, Bayesian with \(T = 28\) (best case), LDA and SVM with RBF kernel, \(\gamma = 1\) and \(C = 0.5\) (best SVM case). For each classifier, \(J_k\) repartition is shown for the graph enforced and graph free classifiers. For each classifier, it is shown that the graph has a significant impact on the classifier’s performance.

Finally, concerning the TUG analysis and the processed database, the graph enforced Bayesian classifier with \(T = 28\) leads to the best performance, with an averaged good detection rate close to 85\%.

### 5. REFERENCES


