Rupture detection for context aware applications

Gaelle Loosli, Sang-Goog Lee, and Stephane Canu

***Abstract*— Automatic context detection through wearable computers and sensors integrated in clothes is the question we address in this paper. Among the many problems that exists in this area, we are particularly interested in the automatic detection of changes in the state of the user, in order to developpe some *context aware* applications. This paper presents a machine learning method for rupture detection, based on one class support vector machines. It also illustrates our algorithm on two experiments in which data are collected by biological sensors and accelerometers.**

I. Context for context detection

The main objective of our research is to design a learning mechanism integrated in a wearable computer to detect, adapt and learn the context. Our approach is not to deal with the whole problem but to focus on a subtask with many potential applications: the automatic detection and learning of the human context. We propose a system that is able, from different sensors, to identify the human context (what is the user doing and what is its emotional state) through learning.

What "to be able to learn" means for a wearable? It means to be able to interpret data provided by sensors in terms of useful information for the system. This depends on the sensors available and it is also strongly application-driven. This knowledge of such a user context can be used in many different applications such as medical intelligence, affective computing (see Picard's work [1]), human computer interface, athlete monitoring or human activity retrieval (see [2], [3]).

The first generation is that of supervised learning mechanisms. Context learning is seen as a classification task with fixed known static classes (the contexts). This is the case in Healey's work [4] where the discriminant analysis has been used to classify different biological signals. In the same conceptual framework, Kim et al. [5] used support vector machines for emotion recognition. These systems are limited because they require the list of possible contexts to be fixed and known in advance for all possible users. The second generation of context learning architecture can detect on-line new contexts through on-line learning. In these works the context is considered in a dynamical way by taking time into account. This implies the introduction of some unsupervised learning mechanism. To do so Van Laerhoven and Aidoo [6] proposed to used Kohonen maps while others advocated the use of hidden Markov model [7]. These mechanisms are interesting but they are not specifically designed to learn dynamical contexts form biological times series. The third generation of context learning architecture does not exists yet. It is a learning mechanism allowing embedded semi-supervised context learning. It should be both a multi-situation and a multi-user mechanism. In other words it should be adaptive to different situations and users thanks to a novelty detection mechanism. Then these detected new states will be provided with an adapted semantics. Our ultimate objective is to create an algorithm of this third generation to be embedded and concealed in a simple shirt equipped with sensors, computing power and wireless network connection.

In this present work, a more powerful and adapted mechanism than those of the second generation is proposed. It aims at learning from the detection of abrupt changes. Our approach consists in monitoring the incoming signals to automatically detect any change in its probabilistic structure of the data without any assumption about it. To do so, a new unsupervised detection mechanism using kernel one class support vector machine (1class SVM) designed to do novelty detection has been adapted. It has been chosen because of its efficiency and because of the outstanding performances of SVM in previous learning tasks. Note that no assumption is made about the nature of the change. Our system transforms an incoming stream of data into a sequence of unlabeled states. As it is, it can be seen as the first step of a third generation context learning mechanism. The remaining part, the labeling of the detected states will be done in further research.

Definition: Context is a highly general notion [8]. Different contexts (or sub contexts) can be defined depending on the system we are interested in. The system is the couple user-machine and context differs regarding the task it has to perform. From this point of view four types of contexts can be distinguished as follows:

- *Environment* (independently of the system) - light, temperature...
- *Interactions* System/Real-World - location, movements of the system...
- *Inner context* (current internal state of the system) - biological data for human, energy levels for electronic devices...
- *Time context* (history).

We consider the context as a composition of three notions. First we define a *state* a fact that is objectively observable and lasts for some time. An *event* will be a punctually and objectively observable fact. States and events may generally be part of *interactions* or *environment*. For instance, an *emotion* is defined as an *inner context* that may depend on states and events and that may not be an objective observation. Because of that, emotions labels may be treated as non reliable when available.

II. Abrupt changes in signals and 1class SVM

As a starting hypothesis we are assuming that the wearer activity and affective state can be represented by a sequence of states. For a given state, the observed time series are the realization of a stationary (or weakly non stationary) process. We assume also that the sequence of states changes slowly in comparison with the measured data. Two times series at different time scales have to be considered: the context time series $c(t)$ (unobserved) and the measurements $x(t)$. The context C is a discrete random variable while the data is a real multidimensional one $x \in \mathbb{R}^d$, d being the number of sensors. The problem of retrieving C given $x(t), t \in [1, T]$ is a problem of signal segmentation.

Because no prior knowledge is made about the nature of the underlying probability distributions, we are looking for a non parametric signal segmentation technique, *i.e.* a method performing an automatic decomposition of a given multidimensional time series into stationary (or weakly non stationary) segments. A way to perform such a segmentation is to use change detection framework from signal processing [9] together with a kernel learning technique [10] to deal with the non parametric requirement. To perform this segmentation online, our detection system has to rely on local characteristics (in time) of the available signals.

From the mathematical point of view the problem can be defined as follows. Let $x(t), t \in [1, T]$ a multidimensional time series. It is assumed to be a realization of a random process with conditional density $\mathbb{P}_t\big(x(t)\big|x(t-1), ..., x(1)\big)$. Again, following [9], the detection problem can be stated as a statistical hypothesis test:

$$\begin{cases} \mathcal{H}_0 & : \mathbb{P}_t = \mathbb{P}_{t-1}, \quad \forall t \in 2, T \\ \mathcal{H}_1 & : \exists t_0 \in 2, T \text{ such that } \mathbb{P}_{t_0} \neq \mathbb{P}_{t_0 - 1} \end{cases}$$

In a first step we decided to simplify the detection procedure by comparing probability distributions of the fixed length T time series before and after time t. Assuming local stationarity, $\mathbb{P}_\tau = \mathbb{P}_b,\ \forall \tau \in [t-T, t]$ and $\mathbb{P}_\tau = \mathbb{P}_a,\ \forall \tau \in [t+1, t+T+1]$, the test performed by our approach is the following:

$$\begin{cases} \mathcal{H}_0 & : \mathbb{P}_b = \mathbb{P}_a \qquad \text{conditional density is the same} \\ \mathcal{H}_1 & : \mathbb{P}_b \neq \mathbb{P}_a \qquad \text{conditional density has changed} \end{cases}$$

The method described here after implement this hypothesis test using one class support vector machine to model the unknown density.

After a brief review of the existing related work a description of the one class support vector machine algorithm will be

A. Related works

There has been much interest in signal segmentation in literature due to the large number of potential applications in the fields of speech recognition, musical indexation or fault detection to cite just a few. These works produced a wide variety of algorithm to perform change detection (see [9], [11] for a complete overview). Up to now, main streams of the available approaches deal with parametric models of the underlying probability distribution. Regarding the non parametric approaches, many models have been used (see [12] for a detailed review in the framework of novelty detection) such as neural networks [13], hidden Markov and bayesian model [14], [15], SVM [16] (and other kernel methods [17]) and belief measurements [18]. The advantage of using SVM with kernels lies is the efficiency of the method in terms of performance, computing time and memory requirement.

B. One class support vector machines: 1-class SVM

The 1-class SVM is a method that aims at learning a single class, by determining it's contours. We try to maximize the margin between the hyperplane and the origin, while most (possibly all is the problem is separable) of the points are lying after the hyperplane on the sphere. If a point is on the wrong side of the hyperplane, it is considered as an outlier.

Formulation: To explain 1-class SVM, we can begin by giving a kernel. A kernel $k(x, y)$ is a positive and symmetric function of two variables (for more details see [19]) lying in a Reproducing Kernel Hilbert Space with the scalar product: $\langle f, g\rangle_{\mathcal{H}} = \sum_{i=1}^{k} \sum_{j=1}^{l} f_i g_j k(\mathbf{x}_i, \mathbf{x}'_j)$.
In this framework, the 1-class SVM problem with the sample $(\mathbf{x}_i), i = 1, m$ is the solution of the following optimization problem under constraints for $f \in \mathcal{H}$:

$$\begin{cases} \min\limits_{f,\rho,\boldsymbol{\xi}} & \frac{1}{2}\|f\|^2_{\mathcal{H}} + C \sum\limits_{i=1}^{m} \xi_i - \rho & \\ \text{with} & f(\mathbf{x}_i) > \rho - \xi_i & i = 1, m \\ \text{et} & \xi_i \geq 0, & i = 1, m \end{cases} \tag{1}$$

where C is a scalar that adjusts the smoothness of the decision function, ρ is a scalar called bias and ξ_i are slack variables.

The dual formulation is:

$$\begin{cases} \max\limits_{\boldsymbol{\alpha} \in \mathbf{R}^m} & -\frac{1}{2}\boldsymbol{\alpha}^\top K \boldsymbol{\alpha} & \\ \text{with} & \boldsymbol{\alpha}^\top e = 1 & \\ \text{and} & 0 \leq \alpha_i \leq C & i = 1, m \end{cases} \tag{2}$$

where K is the kernel matrix $K_{ij} = k(\mathbf{x}_\mathrm{i}, \mathbf{x}_\mathrm{j})$ and $\mathbf{e} = [1, \ldots, 1]^\top$. The 1-class SVM solution is then given by solving a quadratic optimization problem of dimension m under box constraints. The decision function is $D(x) = sign(f(x) - \rho)$. The input points are considered as part of the current class as long as the decision function is positive. This problem is solved with a method derived from the SimpleSVM method: the 1-class SimpleSVM. The SimpleSVM algorithm [20] is based on the decomposition of the database into three groups (the *working* set (I_s), the *inactive* set (I_0) and the *bounded* optimisation problem on the working set only. Having a solution, it checks whether the group repartition is relevant. If not the groups are updated (by adding a violator point in the working set) and it iterates over these two steps [21], [22].

C. Rupture detection algorithm

We are interested in detecting rupture in signals. Let's first describe the kind of results we expect from such an algorithm. We expect the algorithm to detect the ruptures as quickly as possible, with as few false detection as possible (see figure 1).

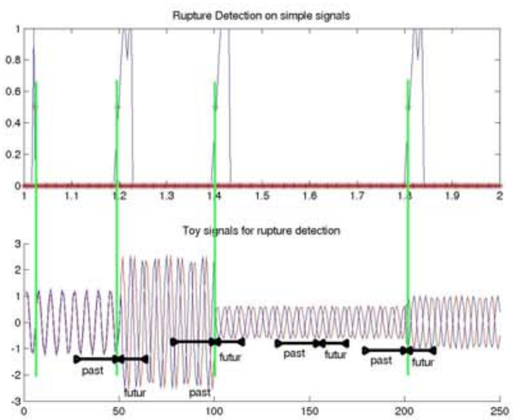

Fig. 1. Rupture illustration.

D. Principle

Based on the idea of novelty detection using 1-class SVM, our method aims at learning the current state and test how well it can recognize the next points (see figure 2).

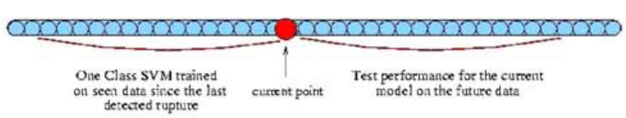

Fig. 2. This figure illustrates how the algorithm works from a general perspective. A rupture is decided when the current trained SVM is enable to well-classify the coming points.

The delay expected in the detection is the length of the signal used to test the current model. In other words, it depends on the number of *future points* we are considering to compute the rupture detection indices (i.e. the misclassification rate).

Figure 3 shows the relation between rupture detection and 1class SVM. On the first part we show the signals, drawn from two different but close gaussian distributions. The second graph shows the ruptures found by our algorithm on those data and the last part is the output of a 1class SVM trained on the

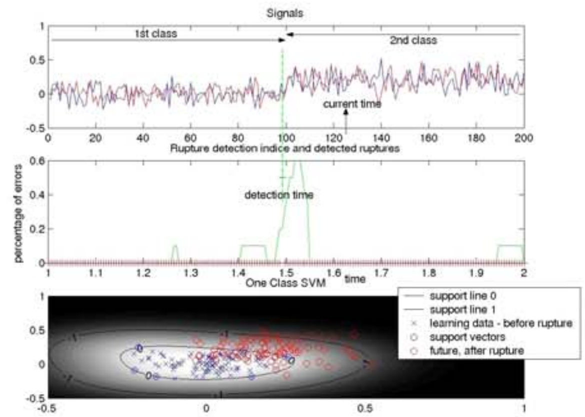

Fig. 3. 1 class illustration.

E. Algorithm

In this section we present the algorithm for rupture detection. This method by itself is simple and the clever part of it is hidden in the the use of the 1-class SVM. The underlying idea is to characterize the current state. As long as the data can be explained by the current model, we expect them to belong to the same state. As soon as part of the incoming point can't be explained any longer, we expect and decide there has been a rupture in the signal. From the 1-class SVM point of view, as long as incoming points lye inside the area determined by the support vectors, they are considered as similar to the previous points.

Algorithm 1 : *Rupture Detection*

1. (data, threshold) ← initialize
2. (features) ← preprocessingData(data)
3. **for** each point
    if 3.1 ClassificationRate(i-1)>threshold
        3.1.1 store rupture time;
        3.1.2 (model)← learn new class (features);
    else 3.2
        3.2.1 (model) ← learn current class (features);
    end
    ClassificationRate(i) = testSVM(model);
end

F. Perspectives

This rupture detection method can be improved, in particular by working on the time representation. The sliding window system is a weak point in our algorithm and we are interested in more powerful method for future work. Moreover we plan to carry out some real on-line detection. Now we have described the main technique used in this part of the project, we will

III. Experiments

In our experiments only the inner context of the user is taken into account. This user's context is composed of the user's activity (sitting, standing, walking, running...) and the affective state. The affective state are of different types [23] including emotions (joy, sadness...), moods (cheerful, gloomy...), interpersonal stances (distant, warm...) attitude (liking, valuing...) and affect disposition (nervous, reckless...). Our goal is not to go through all these affective states but to focus on the the one detectable using noninvasive sensors (simple physiological and physical sensors). Here after for simplicity these kind of affective states will be refer in an abuse of language as emotional state.

We want to acquire signals representing different situations for the user. Those signals should contain information to distinguish states, events and emotions. Moreover we require that the fact that the activity is monitored does not influence the usual behavior of the user. This means that the sensors should not modify or prevent movements or reactions.

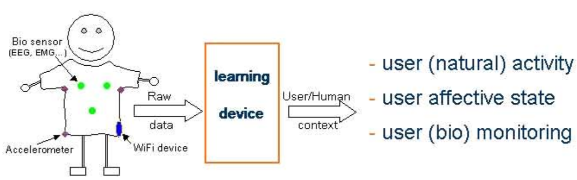

Fig. 4. Our target is to realize the software associated with a shirt equipped with invisible sensors, computer and wireless connection capable of monitoring in real time the wearer activity and affective state.

A. Can you keep control in video games?

In games, states are defined by objective facts (win, lose, particular events...) and there is no particular condition to do the experimentation. The chosen game is XBlast, a game that can be played over the network. The main goal is to kill adversaries with bombs. The target game is a simple game that enable us to define a relatively simple problem.

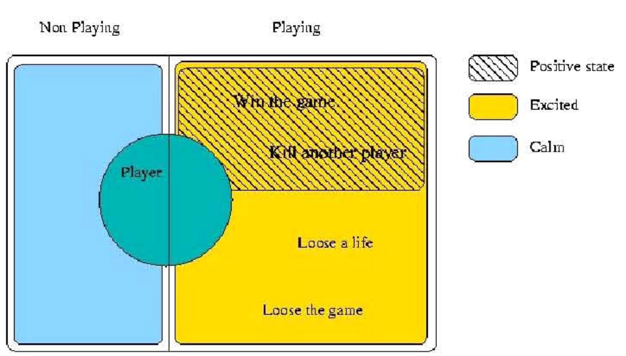

Fig. 5. The player can be either playing or not, which gives two *states*. Moreover during the game, the player can loose a life, kill another player, loose or win the game. This induces four *events* that we can try to detect. Considering emotions, we decided to try to determine positive and negative Emotions could be deduced from signals, states and events. The signals should represent mixtures of these classes. In this experiment, accelerometers are not relevant since the user is continuously sitting. Only the biological sensors are used. One player is equipped with the sensors and filmed (figure 6). He does not know which part of the game will be used for experiments. Figure 6 also shows a sample of signals output.

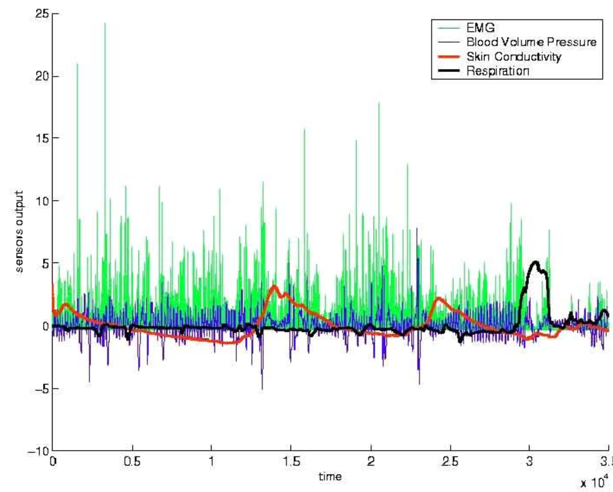

Fig. 6. Example of collected signals. Here are represented the normalized output of the 4 used sensors (EMG, Blood Volume Pressure, Skin Conductivity and Respiration.)

Part of those sets are labeled according to the video. The registered events are:

- player looses a life
- player kills another player
- the game begins
- the player wins
- dangerous events (screen shrinks, many bombs around...)

In this case the labels are not indicating the state of the user but mark where a rupture should be found. However those labels are not completely reliable. First they are not precise (1 second error is quickly done when manually detecting events and this corresponds to several hundreds of input points) and second they are not exhaustive. We did this labeling to have some indicators for the evaluation of our method. However we did expect to obtain some differences between manual and automatic labels.

Results: Biological data are sampled at 256Hz. We work on a sliding window of 64 points (25ms) and take into account the following features: minimum, maximum, mean, variance, Fourier transform on 32 frequencies. We thus work on points in dimension 36. Taking into account the relatively high frequency compared to our application, we consider every eighth point. We apply the 1-class SVM to these feature points (with a gaussian kernel of bandwidth 0.00008). This algorithm is learned on the previous seconds and we test its performance on the next second. Here are the ruptures detected with a threshold of ten percent of misclassified points to decide a

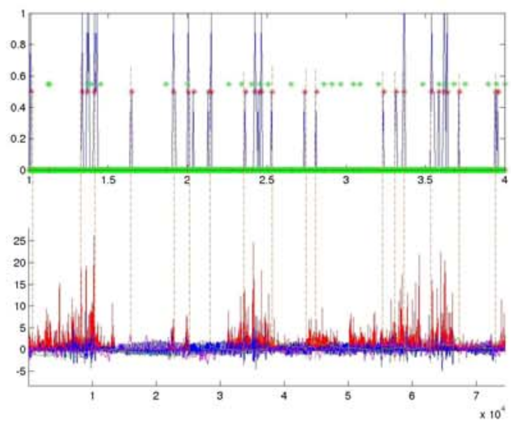

Fig. 7. Results of automatic rupture detection on the first dataset. Red crosses shows the automatically found ruptures and green crosses are the manual labels. On this example it is obvious that automatic labels are more precise. On the other hand some events (mainly killing an other player) do not appear in the signals.

TABLE I

COMPARISON BETWEEN MANUAL AND AUTOMATIC RUPTURE DETECTION.

Number of meaningful ruptures (a posteriori)	31	
Manual Ruptures	26/31	83.9%
Automatic Rupture	21/31	67.7%
In both methods	16/31	51.6 %
Only in manual	10/31	32.3%
Only in automatic	5/31	16.1%
False detection	6/33	18.8%
Non detected	10/31	32.2%

In table I, we sum up the results of this experiment. The two main conclusion we draw from it are first, that manual labeling does not seem to be enough and second that automatic rupture detection is efficient enough (good recognition rate, low false detection rate). We see that automatic rupture detection actually detects ruptures that we can't see on a video.

The problem we can point out is the non detection problem. Indeed several events are not detected with our method. As said previously, events like *killing an other player* do not appear in biological signals. Since our goal was to be able to detect events as well as state and emotions, we are in difficulty. However, it turns out that those events do not really affect the user's state when we can't notice them in the data. Let's illustrate this point : killing a player who has no chance to win is not an important event during the game while taking the last life of the last adversary in game is relevant. This last case will be detected as it will be combined to the emotion of winning the game. From this experiment we can think of a future application in which we would actually both detect rupture and label states. We can as well redefine our goals. We observed that similar events are not meaning the same depending on when they appear. We also observed that meaningful events do appear in the data. Our conclusion from those observations is that it will be hard to really have an objective criteria on

Labeling and Perspectives of the experiment: Now that our system can retrieve changes in the signals, the next step is to give names to each state between every two labels. We can consider the hypothesis that we can *know* some kind of very general states and learn them a priori. But most of classes will be unknown. We will require our system to be able to recognize already seen classes (those classes would be the ones that are relevant to an application for instance) as well as be able to create new states that could be labeled by the user if needed. In this experiment, relevant labels are *playing*, *not playing*, *excited* and *calm*.

B. Natural behavior

In this experiment we try to retrieve information from sensors when the user is doing nothing particularly (with no target application). The idea is then to equip the user with various sensors and let him freely move around with no instruction. This experiment is done with biological and motion sensors and filmed in order to facilitate the interpretation of the signals and labeling.

Since we can't precisely define a priori classes, we will determine them from the video record.This should be easy for motion classes and external events that may occur. As far as emotion are concerned, we will rely on both video and user's feedback to make distinction between calm and strong emotions. This distinction may turn out to be really hard to make since we place the user in a neutral environment. We have to consider the fact that labels will overlap since the user can be walking while feeling very agitated when somebody suddenly calls him. We are then working on mixtures of contexts.

In this experiment we don't know *a priori* which sensors will be relevant so we decided to put on all the sensors we have, namely the five biological sensors and the accelerometers. The user is equipped with the sensors and filmed while wandering around or sitting at his desk. The collected signal are shown on figure 8.

Data Treatment Results: We work on a sliding window of 50 points (50ms) and take into account the following features: minimum, maximum, mean, variance, Fourier transform on 32 frequencies. We thus work on points in dimension 36 per sensor. Similarly to the previous settings, we consider every tenth point. The gaussian kernel's bandwidth is 0.0003.

In table II, we sum up the results of this experiment. As in previous experiment with video games, we observe that the automatic rupture detection from the signals enable us to find some ruptures that are not observable on the video record. For instance the change of breathing that occurs when running or going up the stairs is pretty obvious in the data whereas we can't see it.

Compared to video games, the automatic rupture detection seems to be more noisy. We face here the problem of false detection. We need here to carry out some more experiments

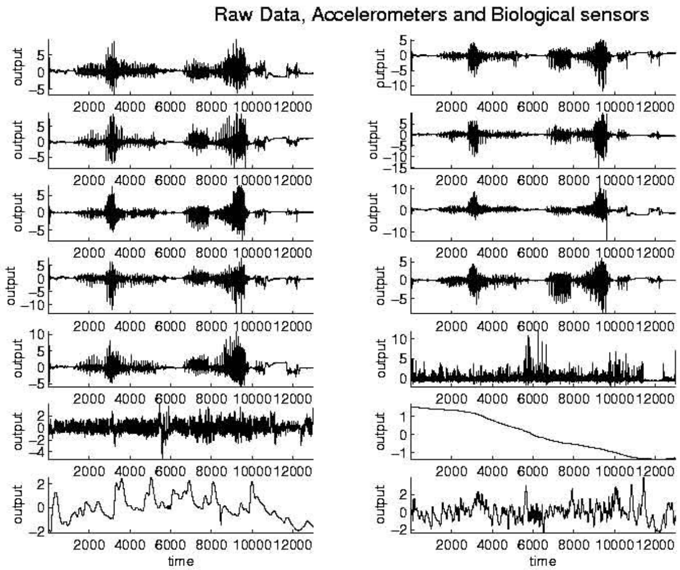

Fig. 8. Example of collected signals. Here are represented the normalised output of the sensors (3 tri-axial accelerometers, EMG, Blood Volume Pressure, Temperature, Skin Conductivity and Respiration).

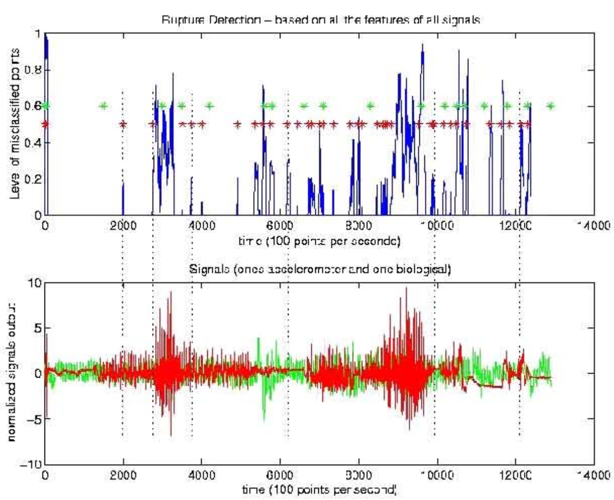

Fig. 9. Here are the ruptures detected with a threshold of ten percent of misclassified points to decide a rupture on the dataset. Red crosses shows the automatically found ruptures and green crosses are the manual labels. For the sake of visibility, the second part of the figure shows one biological signal and one accelerometer signal although the method is applied to all the sensors.

control on the different parameters of the method (mainly the kernel bandwidth and the sliding window on the data).

This experiment, coupled to the previous one, let us think that we can rely on such a rupture detection method to separate different states. The next step will be to be able to add some semantic on these sections of signals or to the ruptures themselves.

Labeling and Perspectives of the experiment: As in previous experiment, the next step is to give names to each state between every two labels. Here the relevant labels could be

TABLE II

COMPARISON BETWEEN MANUAL AND AUTOMATIC RUPTURE DETECTION

Number of meaningful ruptures (a posteriori)	24	
Manual Ruptures	17/24	70.1%
Automatic Rupture	22/24	91.7%
In both methods	15/24	62.5 %
Only in manual	2/24	8.3%
Only in automatic	7/24	29.1%
False detection	12/36	33.3%
Non detected	2/24	8.3%

keep in mind that we can't apply supervised learning in these problems. The best we can do is to give to the machine an idea of what to expect for a specific application. It should be able to extrapolate the correct wanted classes and find out what information is relevant to it.

IV. PERSPECTIVES

Advanced and useful (for the user) wearable software application requires context aware programming. Efficient context aware programming not only requires the ability to sense about the context changes but also to learn on line the context (because users are different and all uses are not predictable). So learning algorithm have to be embedded in autonomous learning systems. How to do it is the problem addressed by our work. In this work an embedded learning software architecture has been proposed as a first step toward the solution of the problem. Based on non parametric unsupervised learning algorithm (the one class SVM), it aims at detecting the abrupt changes in the available signals and by doing so performs at the same time the detection and the learning of the contexts. Two experimentations on real data illustrate the potentiality of our method. The first target application focuses on the detection of the emotion of a video player while the second one deal with the analysis of usual activities such as walking, sitting... The natural next step of our work is the automatic labeling of the detected states. To do so graphical model such as hidden Markov models have to be considered. Then previous information about possible user's state have to be defined in an automatic way. This is a very interesting and promising challenge. The experimentation phase has to be developed taking different users during longer time. Other potential applications have to be considered such as the analysis of the affective state of a sportsman (for instance a tennis player) using our system. This may help to understand how a player manages its games. Audio sensors and speech processing algorithm should be integrated to improve the reliability of the system. This require to do multimodal integration between the voice and the physiological behavior.

REFERENCES

[1] R. W. Picard, S. Papert, W. Bender, B. Blumberg, C. Breazeal, D. Cavallo, T. Machover, M. Resnick, D. Roy, and C. Strohecker, "Affective learning : A manifesto," *BT Technology Journal*, vol. 22, no. 4, pp. 253–269, 2004.

[2] K. V. Laerhoven and H.-W. Gellersen, "Spine versus porcupine: A study in distributed wearable activity recognition," in *ISWC '04: Proceedings of the Eighth International Symposium on Wearable Computers*

[3] S. S. Intille, L. Bao, E. M. Tapia, and J. Rondoni, "Acquiring in situ training data for context-aware ubiquitous computing applications," in *CHI '04: Proceedings of the 2004 conference on Human factors in computing systems*. ACM Press, 2004, pp. 1–8.

[4] J. Healey, "Wearable and automotive systems for detecting affect from physiology," PhD Dissertation, M. I. T, MediaLab, 2000.

[5] K. H. Kim, S. W. Bang, and S. R. Kim, "Emotion recognition system using short-term monitoring of physiological signals," *Medical and biological engineering and computing*, vol. 42, 2004.

[6] K. V. Laerhoven and K. Aidoo, "Teaching context to applications," *Personal Ubiquitous Comput.*, vol. 5, no. 1, pp. 46–49, 2001.

[7] D. Zhang, D. Gatica-Perez, S. Bengio, I. McCowan, and G. Lathoud, "Multimodal group action clustering in meetings," in *VSSN '04: Proceedings of the ACM 2nd international workshop on Video surveillance & sensor networks*. ACM Press, 2004, pp. 54–62.

[8] H. Lieberman and T. Selker, "Out of context: Computer systems that adapt to, and learn from, context," vol. 39, 2000.

[9] M. Basseville and I. V. Nikiforov, *Detection of Abrupt Changes - Theory and Application*. Prentice-Hall, 1993.

[10] B. Schölkopf and A. J. Smola, *Learning with Kernels*. MIT Press, 2002.

[11] F. Gustafsson, *Adaptive filtering and change detection*. John Wiley & Sons, Ltd, 2001.

[12] M. Markou and S. Singh, "Novelty detection: a review, part 2: neural network based approaches," *Signal Process.*, vol. 83, no. 12, pp. 2499–2521, 2003.

[13] C. Fancourt and J. C. Principe, "On the use of neural networks in the generalized likelihood ratio test for detecting abrupt changes in signals," in *Intl. Joint Conf. on Neural Networks, pp. 243-248, at Como, Italy*, 2000.

[14] S. Roberts, E. Roussos, and R. Choudrey, "Hierarchy, priors and wavelets: structure and signal modelling using ica," *Signal Process.*, vol. 84, no. 2, pp. 283–297, 2004.

[15] Y. Qi and T. Minka, "Expectation propagation for signal detection in flat-fading channels," in *Proceedings of the IEEE International Symposium on Information Theory*, 2003.

[16] M. Davy and S. Godsill, "Detection of abrupt spectral changes using support vector machines," in *Proc. IEEE ICASSP-02*, 2002. [Online]. Available: citeseer.ist.psu.edu/davy02detection.html

[17] X. Nguyen, M. J. Wainwright, and M. I. Jordan, "Nonparametric decentralized detection using kernel methods," *IEEE Transactions on Signal Processing (accepted for publication)*, 2005.

[18] S. Lenser and M. Veloso, "Non-parametric time series classification," in *Under review for ICRA'05*, 2005.

[19] M. Atteia and J. Gaches, *Approxiation Hilbertienne*. Presses Universitaires de Grenoble, 1999.

[20] S. V. N. Vishwanathan, A. J. Smola, and M. N. Murty, "SimpleSVM," in *Proceedings of the Twentieth International Conference on Machine Learning*, 2003.

[21] G. Loosli, S. Canu, S. Vishwanathan, A. J. Smola, and M. Chattopadhyay, "Une boîte à outils rapide et simple pour les SVM," pp. 113–128, 2004.

[22] G. Loosli, "Fast svm toolbox in Matlab based on SimpleSVM algorithm," 2004, `http://asi.insa-rouen.fr/~gloosli/simpleSVM.html`.

[23] K. R. Scherer, "Ways to study the nature and frequency of our daily emotions: Reply to the commentaries on "emotions in everyday life"," *Social Science Information*, vol. 43, no. 4, pp. 667–689, 2004.