31st Annual International Conference of the IEEE EMBS
Minneapolis, Minnesota, USA, September 2-6, 2009

Acoustic Fall Detection Using One-Class Classifiers

Mihail Popescu, *Senior Member*, *IEEE*, Abhishek Mahnot

Abstract—**Falling represents a major health concern for the elderly. To address this concern we proposed in a previous paper an acoustic fall detection system, FADE, composed of a microphone array and a motion detector. FADE may help the elderly living alone by alerting a caregiver as soon as a fall is detected. A crucial component of FADE is the classification software that labels an event as a fall or part of the daily routine based on its sound signature. A major challenge in the design of the classifier is that it is almost impossible to obtain realistic fall sound signatures for training purposes. To address this problem we investigate a type of classifier, one-class classifier, that requires only examples from one class (i.e., non-fall sounds) for training. In our experiments we used three one-class (OC) classifiers: nearest neighbor (OCNN), SVM (OCSVM) and Gaussian mixture (OCGM). We compared the results of OC to the regular (two-class) classifiers on two datasets.**

I. Introduction

More than one third of about 38 million adults 65 and older fall each year in the United States [1]. About 30% of people who fall suffer severe injuries such as fractures and head trauma [1] that can render them unable to raise or to ask for help. It is well documented in the medical literature [2-4] that the longer the lie on the floor the poorer is the outcome of the medical intervention. Hence, it is imperative that the falls are detected and the necessary help is provided as soon as possible.

Among the fall detection methods [5], acoustic fall detection is the least investigated. Acoustic sensors have been previously used in habitat monitoring [6-13]. In [6]-[11] a set of acoustic sensors was used to differentiate between several sound classes such as breaking glass, screams, steps, door sound and human sound. A microphone was placed in each room of the apartment to indentify the location of the sound. The acoustic sensor used in the ListenIn system [12] was designed for activity monitoring (baby noise or loud noise). The alarm, together with the encrypted sound, was send to a mobile device held by a caregiver. In [13] two microphones were used to detect kitchen activities. The two microphone setup was used to remove noise using a method based on the eigenvalues of the cross spectral matrix. Human falls were not included in the sound classes detected in any of the above acoustic systems. Only recently, two acoustic fall detection systems have been proposed [14, 15]. In [14] a microphone together with a floor vibration sensor were employed to detect falls based on 110 sound and vibration features that include Mel-frequency cepstral coefficients (MFCC) and shock response spectrum. After a sequential forward floating selection of the features a Gaussian mixture classifier was used to detect falls from non-falls. In fact, all the previously mentioned sound based-activity recognition applications used either Gaussian mixtures or nearest neighbor classifiers together with MFCC features to detect various activities of daily living (ADL).

There are several problems encountered when training classifiers to detect ADL and to predict abnormal behavior. First, one cannot predict all the normal activities, such that a classifier be trained for each type of behavior. Second, following from the first problem, the classifier has to adapt and learn new normal activities in order to reduce the false alarm rate when detecting the abnormal ones. Third, some abnormal behaviors are hard (some plain impossible) to simulate. This is our biggest challenge in designing an acoustic fall detector. That is, the sound of a falling human body cannot be perfectly simulated. Using fall dummies such as Rescue Randy [14] cannot produce realistic fall sounds due to the hard skin (unlike the human one) and lack of bones of the mannequins. Some researchers tried to obtain training data by recording sound of falls performed by stunt actors [15]. While this method may be visually realistic (can be used for video-based fall detection), it is not sound-wise (perfectly) realistic due to the natural instinct of human beings to avoid a painful fall (aside of the fact that we could not convince even the stunt actor to fall on plain concrete).

In this paper we describe the experiments performed with FADE, a dedicated acoustic **fa**ll **de**tection system based on a linear array of acoustic sensors and a motion detector [15]. The system is inexpensive and built from off-the-shelf components. In our first experiments [15] we used nearest neighbor together with MFCC to distinguish between falls simulated by a stunt actor and various man-made noises. To address the problems described in the previous paragraph, we investigated two possible solutions: fuzzy rule systems (do not need training data, just expert knowledge about the activity) and one-class classifiers (need only the non-fall data). To build the fuzzy rule system [16], we observed that the fall have most of the spectral energy in the low frequency range (under 300 Hz). The results [16] were encouraging but we realized that MFCC features do not lend themselves to an expert interpretation. Some new features that we are currently investigating show more promise. The second approach, one-class classifiers is described in this

M. P. is with the Health Management and Informatics Department, University of Missouri, Columbia, MO 65211, USA (corresponding author, phone: 573-882-1266; fax: 573-882-6158; e-mail: popescum@missouri.edu).

A. M. is with the Electrical and Computer Engineering Department, University of Missouri, Columbia, MO 65211, USA.

paper.

The structure of the paper is as follows: in section II we present the architecture of FADE, in section III we briefly describe the one-class classifier algorithms used for fall detection, in section IV we describe the test data and methodology, in section V we show the results and in section VI we give the conclusions.

II. SYSTEM ARCHITECTURE

The architecture of the FADE system is shown in Fig. 1.

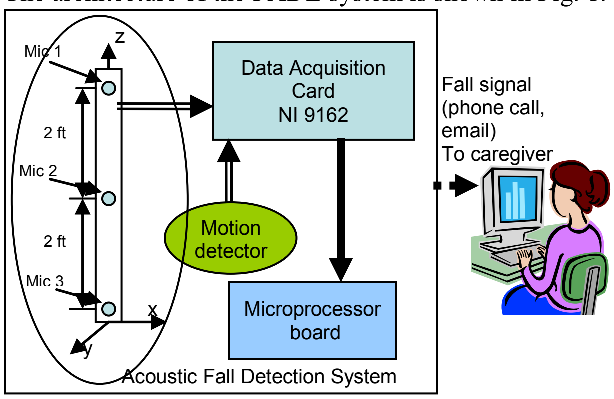

Fig. 1 The proposed fall detector architecture

The fall detector consists of a linear array of electret condenser acoustic sensors (three shown: Mic 1, Mic 2 and Mic 3) mounted on pre-amplifier boards Cana Kit CK495 (about $20 each). More acoustic sensors might be considered in the future to help improve beam forming and source separation. The acoustic sensor array was mounted vertically (along z-axis) on a wall in order to be able to capture sound height information. The FADE incorporates a motion detector for further reduction of the false alarms. The working hypothesis for FADE is that the person is alone in the department. If motion is detected during a given interval (one minute) after a fall event is computed as likely, the caregiver alert is not issued. Instead, the event that provoked the alarm is cataloged as a false alarm and used to retrain the classifier(s). In fact, we intend to integrate FADE in a home monitoring system where information from the entire sensor network can be used to decide if the fall was possible. For example, the person wakes up in the middle of the night (bed sensor alert), goes to the bathroom (bathroom sensor alert), noise detected (is this a fall?) then goes back in bed (bed sensor triggers – no, it was not a fall). In the first stage of our project we intend to detect falls after which the person cannot move. A more ambitious goal would be to try to detect all falls, including the non-injury ones. However, it would be very hard to assess the efficiency of the detection algorithm since these falls are usually not reported. The existence of pets in the apartments is addressed by using a pet-immune motion detector. In order to preserve the privacy of the patient, the sound will be internally processed on a microprocessor board and only an external fall signal (email or pager) will be send to the caregiver.

In this work, since we were mainly interested in comparing the two class classifiers to the one-class ones for acoustic fall detection, we simplified the setup shown in Figure 1. More precisely, we only used one microphone to record data. As a consequence, since we did not use the sound height information, it is expected that the fall detection performance shown in this paper is lower than in [15]. The sound was recorded on a laptop using a National Instruments data acquisition card NI 9162 with 8 differential analog inputs. The recorded sound (as described in Section IV) was later processed using MATLAB (http://www.mathworks.com).

III. ALGORITHM DESCRIPTION

The main steps of the signal processing algorithm (see Figure 2) were: signal preprocessing and fall recognition.

A. Signal preprocessing

The first step in preprocessing was noise removal using a Wiener filter [17].

The next step was to compute the energy E_w of the signal s in a window k using:

$$E_w(k) = \sum_{n=Nk-(N-1)/2}^{n=Nk+(N-1)/2} s^2(n) \qquad (1)$$

where N is the number of samples in the window. The considered window was 1 second of sound signal (N=20000 samples for a sampling frequency f_s=20000 Hz) with a 50% overlap between consecutive windows. If the energy of the signal in a window was smaller than a given threshold, E_{THR}, the signal was labeled "no fall event" and no further processing was performed. An experimental value E_{THR} = 2 was determined by averaging the energy of 60 seconds of silence (no sound events).

B. Fall recognition

The feature extraction follows the energy calculation if the signal from a window has the energy above E_{THR}. We used the mel frequency cepstral coefficients as features (MFCC). Since MFCCs mimic human auditory system, they might be a good choice if FADE will be used in the future to detect distress calls (e.g. “help”). In the same time, since fall signals are mostly bounded to the 0-300 Hz energy band, other sound features, e.g. energy sub-band ratios, might perform better than MFCC. The number of coefficients (features) used was C=7. The features were extracted using the MATLAB function, *mfcc*, from [17]. To make the system less dependent on the distance to the sound source, we did not use the first cepstral coefficient (proportional to the signal average) in the recognition procedure. The recognition was performed using one of the three one-class classifiers described in the next section. A fall has to be detected in two consecutive windows in order to be reported as such.

C. C. One-class classifiers

We can approach our pattern classification problem in two ways: as a class imbalance [18] or as a one-class classifier [19] detection. In the class imbalance approach, various methods are used for the sampling of the two classes (fall and non-fall) together with a traditional two-class classifier. This approach does not work in cases where samples from one class cannot be acquired due to unpredictability (as in fraud detection) or to harm caused to people (falling on a hard surface). In the one-class classifier (OCC) approach, also known as novelty detection [20], abnormality detector, fault detector, outlier detector or concept learning [19], we use only the samples from the most abundant class (i.e. non-falls) to train the classifier. The main advantage of this approach is that the classifier can easily adapt to new incoming data without any worries about data imbalance.

In this paper we examine three OCCs: OC Gaussian mixture, OC nearest neighbor, and OC support vector machine. The OCCs can be classified [19] in density methods (such as Parzen and Gaussian mixture) and boundary methods (such as nearest neighbor and SVM. The unified approach to OCC proposed in [19] does not use a threshold for accepting the target class objects. Instead, it assumes that a certain percentage, t_0, of the training data are outliers. The main effect of this formulation is that it labels as outliers some target feature vectors, that might be due to, for example, poor data quality. Although this formulation translates to an internal threshold (for each t_0), it makes the threshold adaptable when new data is received. This approach makes the detection algorithm more robust to changing environmental conditions (e.g. the pile of a new carpet).

The OC Gaussian mixture (GM) method, OCGM, is mathematically similar to the traditional one, that is:

$$OCGM(\boldsymbol{x}) = \sum_{i=1}^{M} w_i \, N_i(\boldsymbol{x}), \quad (1)$$

where $\boldsymbol{x}$ is the *mffc* feature vector extracted from signal s, w_i are a set of weights and N_i a set of normal distributions of the *mfcc* features. As mentioned above, instead of using a (probability) threshold p_0 to decide if $\boldsymbol{x}$ belongs to the target class (i.e. $OCGM(\boldsymbol{x})>p_0$), the percent of the training class t_0 that represents outliers is used to compute the optimality threshold [19].

The OC nearest neighbor (denoted as NNd in [19]) is defined as:

$$OCNN(\boldsymbol{x}) = \frac{NN(\boldsymbol{x})}{NN(NN(\boldsymbol{x}))}, \quad (2)$$

where $NN(\boldsymbol{x})$ is the nearest neighbor of $\boldsymbol{x}$.

As opposed to the regular SVM (denoted as SVDD in [19]) that separates the two classes in the feature space by a hyper plane, OC SVM surrounds the target class in the feature space by a hyper-sphere. Formally, we need to minimize:

$$R^2 + C\sum_i \xi_i, \quad (3)$$

where ξ_i are slack variables, R is the radius of the hyper-sphere and C is a constant, with the constraints that the objects be in a sphere of radius R:

$$||\boldsymbol{x}_i - \boldsymbol{a}||^2 \leq R^2 + \xi_i, \quad (4)$$

where $\boldsymbol{a}$ is the center of the sphere. In the above formulation, $\boldsymbol{a}$ and R are computed such that t0 of the training set objects will lay outside the sphere. Details about OCSVM can be found [19].

In this paper we used the MATLAB implementation of the above OCC methods called DDTools [21]. We compared the above methods to their two class counter parts implemented in the PRTools (www.prtools.org) MATLAB package.

IV. Study Methodology

We collected two data sets: one for training and another for testing (see Table I).

TABLE I. EXPERIMENTAL DATA

	Falls	Non-falls	Dataset
Training	30	60	90 files/1s
Testing	36	72	1 file/1h

The training data consisted in 90 sound sequences, about 1 s long that consisted of 30 falls and 60 non-falls. The non-falls sounds were produced by dropping objects (keys, books), knocking (wall, door, table), clapping and phone-calling related (talking, ringing, hanging off). The falls were performed by the authors on various surfaces such as carpet, soft-surface mat and hard-surface mat.

The testing data consisted in an hour-long recording performed in our lab. In that time 72 non-fall sound were produced (similar to the ones described in the training data) and 36 falls. We mention that even if the testing data was recorded using a somewhat artificial scenario (almost two events per minute), the resulting ROC curves will provide a reasonable comparison of two classification approaches. However, the resulting ROC curves will be pessimistic (high FAR) due to the high density of the non-fall events.

The OC classifiers were compared to their two-class counterparts using receiver-operator characteristic (ROC) curves and the related area under the curve (AROC). The results on the training data were obtained using leave-on-out cross validation. For the OCCs we used only the non-falls for classifier training. An alternative approach that consisted in using all data (falls and non-falls) but setting the outlier percent t_0 = 0.33 (that is 30/90) did not result in better results. The results on the testing data were obtained by training the classifiers on the training data. Again, OCCs were only trained on the non-fall training data. Some results are presented in the next section (more can be found in [22]).

V. Results

A. Results on the training data

The ROC comparison of one-class Gaussian mixture (M=1) to the two-class one (M=1) is shown in Figure 2. The related AROCs are 0.76 and 0.87 for OCGM (with t_0=0.1) and two-class GM, respectively. We see that the two-class classifier outperforms the one-class one, but they seem to be very close in the medium and high false alarm rate range.

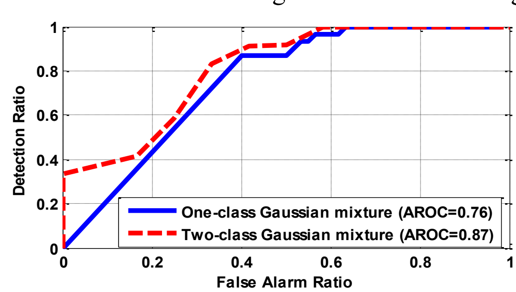

Fig. 2. OCGM compared to the two-class Gaussian mixture

The comparison between OCSVM and two-class SVM is shown in Figure 3. The AROC for the two class SVM is higher than the one for OCSVM, but the difference between the two is only 2%. It seems that the training using only the non-fall class had the least impact on the SVM in comparison to the other algorithms considered here.

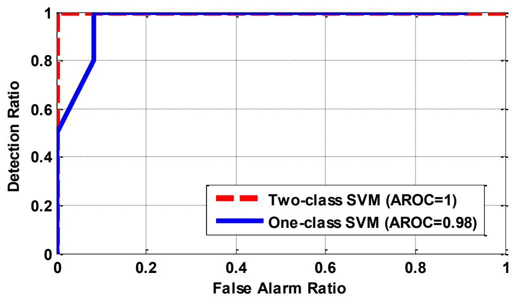

Fig. 3 ROC comparison between OCSVM and two-class SVM

The results for the OCNN are given in figure 4. Here too, the OCNN performed worse than the two-class NN by about 10%. In addition, OCNN has the worst performance among the classifier tried.

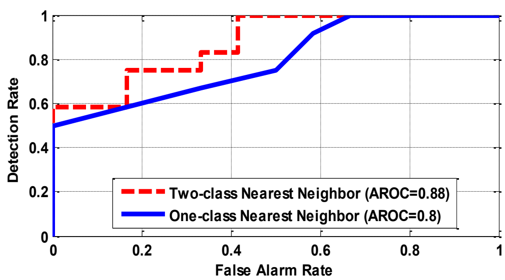

Fig. 4 ROC comparison between OCNN and two-class NN

Considering the results obtained on the training data, we further chose to run only the OCSVM on the test data. The comparison between the OCSVM and regular SVM on the test data are shown in Figure 5.

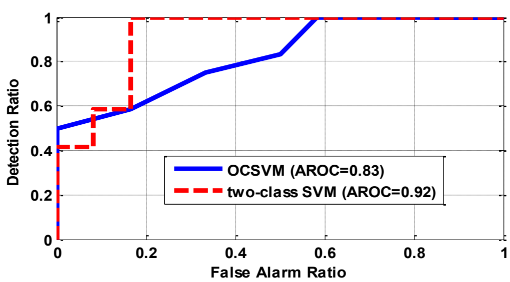

Fig. 5 ROC comparison between OCSVM and two-class SVM

Although the regular SVM has a higher AROC, the OCSVM seems to perform better in the low false alarm rate (FAR). In fact, this is the area of the ROC we are the most interested in, since we would like our final FADE system to have the smaller possible FAR. However, more experiments presented in [22] support the general conclusion that two-class classifiers outperform OCCs for fall detection.

VI. Conclusions

In this paper we investigate a possible solution for the situation when data for one of the two classes is not available in order to train a two-class classifier. The solution investigated here consists in reformulating the two-class classifiers such that they accept only one class for training.

We applied three one-class classifier methods to acoustic fall detection. In our case, we cannot collect realistic fall data, hence only non-fall sound (environmental) are available. The results obtained on two limited datasets suggest that, although the one-class SVM underperforms overall the regular SVM, it achieves an acceptable classification performance mainly in the low FAR region.

More work has to be done in the area of refining the sound fall features and combining the OCSVM with fuzzy rule systems.

References

[1] Center for Disease Control (CDC), http://www.cdc.gov/ncipc/factsheets/adultfalls.htm.

[2] P. J. Ratclife, J.G.G Ledingham, P. Berman, G.K. Wilcock, J. Keenean, "Rhabdomyolysis in elderly people after collapse", *British Med. J.*, vol. 288, pp. 1877-8, 1984.

[3] R. J. Gurley, N. Lum, M. Sande, B. Lo, M. H. Katz "Persons found in their homes helpless or dead". *N Engl J Med*.; 334(26):1710-6, 1996.

[4] C. G. Moran, R.T. Wenn, M. Sikand, A.M. Taylor, Early mortality after hip fracture: is delay before surgery important", *J. of Bone and Joint Surgery*, pp. 483-9, 2005.

[5] N. Noury, A. Fleury, P. Rumeau, et al., "Fall detection-principles and methods", Proc. of the 29th IEEE EMBS, Lyon, France, Aug. 23-26, 2007.

[6] E. Castelli, M. Vacher, D. Istrate, L. Besacier, J.F. Serignat, "Habitat telemonitoring system based on the sound surveillance", ICICTH (International Conference on Information Communication Technologies in Health), 11-13 July 2003, Samos Island, Greece.

[7] M. Vacher, D. Istrate, L. Besacier, E. Castelli, Jean-Francois Serignat, "Smart audio sensor for telemedicine" Smarts Objects Conference (SOC) 2003, 15-17 May, Grenoble, France.

[8] D. Istrate, E. Castelli, M. Vacher, L. Besacier, JF Serignat, "Information extraction from sound for medical telemonitoring", *IEEE Trans. on Information Tech. in Biomedicine*, vol. 10, no.2, April 2006.

[9] Istrate D., Binet, M., Cheng, S., "Real Time Sound Analysis for Medical Remote Monitoring", 30th Int. IEEE EMBS Conf., Vancouver, BC, Aug. 20-24, 2008, pp., 4640-4643

[10] Fleury A, Noury N, Vacher, M., Glasson H., Serignat J.-F., "Sound and Speech Detection and Classification in a Health Smart Home", 30th Int. IEEE EMBS Conf., Vancouver, BC, Aug. 20-24, 2008, pp. 4644-4647.

[11] N. C. Laydrus, E. Ambikairajah, B. Celler, "Automated sound analysis system for home telemonitoring using shifted delta cepstral features", 15th International Conference on Digital Signal Processing, pp. 35-38, 2007.

[12] C. Schmandt, G. Vallejo, "'Listenin' to domestic environments from remote locations", Proc. of the 2003 Int. Conf. on Auditory Display, Boston, MA, 6-9 July 2003.

[13] David Maunder, D., Ambikairajah, E., Epps, J. and Celler, C., "Dual-Microphone Sounds of Daily Life Classification for Telemonitoring in a Noisy Environment", 30th Int. IEEE EMBS Conf., Vancouver, BC, Aug. 20-24, 2008, pp. 4636-4639.

[14] Litvak D, Yaniv Zigel, Y., Gannot, I., "Fall Detection of Elderly through Floor Vibrations and Sound", 30th Int. IEEE EMBS Conf., Vancouver, BC, Aug. 20-24, 2008, pp. 4632-4635.

[15] Popescu, M., Li, Y., Skubic, M., Rantz, M., "An Acoustic Fall Detector System that Uses Sound Height Information to Reduce the False Alarm Rate", 30th Int. IEEE EMBS Conf., Vancouver, BC, Aug. 20-24, 2008, pp. 4628-4631.

[16] M. Popescu, S. Coupland, "A fuzzy logic system for fall detection", AAAI Fall Symposium, Washington DC, Nov 7-9 2008, pp. 78-83.

[17] M. Slaney, "Auditory toolbox 2.0", http://www.slaney.org/malcolm.

[18] Japkowicz, N., Stephen, S., The Class Imbalance Problem: A Systematic Study , Intelligent Data Analysis J., Vol. 6, No. 5, Nov. 2002.

[19] Tax DMJ, One-class classification, PhD Thesis, TU Delft, NL, 2001.

[20] Markou M., Singh, S., Novelty detection: A review - part 1: Statistical approaches, Signal Processing J., vol 83, 2003.

[21] Tax D.M.J., DDtools, the Data Description Toolbox for Matlab, version 1.5.7, 2007.

[22] Mahnot, A, "Using one class classifiers for fall detection", MS Thesis, University of Missouri, Columbia, MO, 2009.