On-Line Fall Detection via Mobile Accelerometer Data

S. V. Georgakopoulos, S. K. Tasoulis, I. Maglogiannis, V. P. Plagianakos

On-line Fall Detection via Mobile Accelerometer Data

S.V. Georgakopoulos[1], S.K. Tasoulis[3], I. Maglogiannis[2], and V.P. Plagianakos[1]

[1] Department of Computer Science and Biomedical Informatics,
University of Thessaly, Papassiopoulou 2–4, Lamia, 35100, Greece
`{spirosgeorg,vpp}@dib.uth.gr`

[2] Department of Digital Systems,
University of Piraeus,
Grigoriou Lampraki 126, Piraeus, 18532, Greece
`imaglo@unipi.gr`

[3] Helsinki Institute for Information Technology HIIT,
Department of Computer Science,
University of Helsinki, Finland
`tasoulis@cs.helsinki.fi`

Abstract. *Mobile devices have entered our daily life in several forms, such as tablets, smartphones, smartwatches and wearable devices, in general. The majority of those devices have built-in several motion sensors, such as accelerometers, gyroscopes, orientation and rotation sensors. The activity recognition or emergency event detection in cases of falls or abnormal activity conduce a challenging task, especially for elder people living independently in their homes. In this work, we present a methodology capable of performing real time fall detect, using data from a mobile accelerometer sensor. To this end, data taken from the 3-axis accelerometer is transformed using the Incremental Principal Components Analysis methodology. Next, we utilize the cumulative sum algorithm, which is capable of detecting changes using devices having limited CPU power and memory resources. Our experimental results are promising and indicate that using the proposed methodology, real time fall detection is feasible.*

Keywords: Principal Component Analysis, Incremental Principal Component Analysis, Cumulative Sum, Smartphones, Accelerometer Sensor

1   Introduction

The life quality improvement of social communities and individuals constitutes fundamental priority on a wide range of research fields. The opportunity that every human living independent of the continuous concern of the fellowmen, is integral right. At the same time, the population of elderly is increased, causing new requirement for social inclusion and access to healthcare.

In recent literature [15, 9, 5] is reported that approximately 33% of elders over the age of 65 and 50% over the age of 85 experience a fall each year.

Such incidents may cause light or serious injuries. The studies indicate that hip fractures of 50% of older people make them unable to live independently, while 25% and 33% of them will die within six months and a year, respectively. Falling injuries are frequently identified both in home and in hospital environments and make many elderly persons live under continue observation, causing discomfort. Discreet monitoring of the individuals in order to identify such major injuries, is a challenging task.

Several techniques have been produced to detect emergency events, having the common aspect of collecting sensor data in an attempt to interpret activity. More specifically, in the field of activity recognition with focus to fall detection, many recent studies can be found in the literature [1, 2, 6, 3]. Some studies use microphones to detect falls, while others use information obtained by cameras that track the movement and the trajectory of humans in assisted environments, where long and/or unusual inactivity can be interpreted as a fall. Another approach is to use signals acquired by accelerometers and gyroscopes [7]. To analyse these data, machine learning [10] and statistical methods [4] have been proposed.

In this work, we attend to discreetly monitor an individual using data from a smartphone (or a wearable sensor) that is equipped with a 3-axis accelerometer, to on-line detect falls. To interpret the accelerometer data, we project the three dimensional data to the principal component direction. Instead of computing the space of eigenvectors by the traditional Principal Component Analysis (PCA), which requires a batch computation step, we employ the Incremental Principal Component Analysis (IPCA). Finally, we incorporate a parametric model, which is able to detect the change at unknown time instants. The technique we used here is the Cumulative Sum (CuSum) algorithm [11].

The rest of the paper is structured as follows: In Section 2, background material for dimensionality reduction and classification methods are provided. In Section 3, we present the proposed methodology and the experimental results. Finally, Section 4 contains concluding remarks and pointers for future work.

2   Background Methods

In this section, we review the basic tools used in the proposed methodology. In particular, we present the Incremental Principal Component Analysis (IPCA) for the feature extraction task and the Cumulative Sum (CuSum) algorithm for the classification of the transformed data.

2.1   Incremental Principal Components Analysis

The typical computational approach to PCA requires all the data input to be available in order to compute the eigenvalues and eigenvectors of the sample covariance matrix, and thus it belongs to the category of batch methods. This approach is not feasible when the data are incrementally derived from an on-line stream. Thus, an incremental method is required to estimate the principal components for observations arriving sequentially.

This can be achieved by updating the principal components for by each arriving observation vector, while avoiding to estimate the covariance matrix as an intermediate result. Here, for that purpose we employ the Candid Covariance-free IPCA (CCIPCA) method [14], which is based on the works of Oja and Karhunen [8] and Sanger [12]. A short description of the method follows.

Let $d_1, d_2, \ldots$ be the sample vectors that are acquired sequentially at each time point and let u_1 be the first principal component. Each $d_n, n = 1, 2, \ldots$, is a a-dimensional vector, where each dimension corresponds to an accelerometer signal for the case at hand. Without loss of generality, we can assume that d_n has a zero mean, since the mean may be incrementally estimated and subtracted out. Then, the n-th step estimate u_1^n of u_1 is given by

$$u_1^n = \frac{n-1-l}{n} u_1^{n-1} + \frac{1+l}{n} d_n d_n^T \frac{u_1^{n-1}}{\|u_1^{n-1}\|},$$

where $(n-1)/n$ is the weight for the last estimate and $1/n$ is the weight of the new data, while the one dimensional projection y_n onto u_1^n is given by

$$y_n = u_1^n d_n.$$

The positive parameter l is called the amnesic parameter. With the presence of l, larger weight is given to new samples and the effect of old samples will gradually fade out. Finally, to begin the iteration, we set $u_1^0 = d_1$, the first direction of data spread. A mathematical proof of the convergence of CCIPCA can be found in [16].

2.2 Cumulative Sum

The Cumulative Sum (CuSum) is a change detection algorithm that can be used for off-line or on-line change detection. The CuSum has been firstly proposed in [11]. In this work, we utilize the on-line version of the algorithm.

To describe the functionality of the algorithm, we consider a sequence of independent random variables y_n, which correspond to a signal for various discrete time instances n, with a probability density $p_\theta(y)$, which depends only to the parameter θ. To capture the change of the signal at an unknown time instance t_0, the parameter θ have to change from the initial value θ_0 (θ is equal to θ_0) to θ_1 where $\theta_0 \neq \theta_1$. The approximation of these parameters that can be addressed using a training set of sample data.

More specifically, we assume the following hypotheses concerning the parameter θ:

$$H_0 : \theta = \theta_0$$
$$H_1 : \theta = \theta_1 \tag{1}$$

In the fall detection problem considered here, the signal samples are captured by the accelerometer in real time and the proper hypotheses must be computed.

When the decision is continuously in favor of hypothesis H_0, there is absence of signal change, while a decision in favor of H_1 corresponds to signal samples that indicate a change, i.e. a fall.

In this paper, the following notation will be used. Let

$$S_n = \sum_{i=1}^{n} s_i, \text{ where } s_i = ln\frac{p_{\theta_1}(y_i)}{p_{\theta_0}(y_i)} \tag{2}$$

is the log-likelihood ratio for the observations from y_i to y_n, where n is the current time instant and refer to s_i as sufficient statistic. Considering the particular case where the distribution is Gaussian, with μ the mean value and σ the constance variance, when the changing parameter θ corresponds to μ. The probability density is denoted as

$$p_\theta(y) = \frac{1}{\sigma\sqrt{(2\pi)}} \exp^{-(y-\theta)^2/2\sigma^2} \tag{3}$$

and the sufficient statistic s_i as

$$s_i = \frac{\theta_1 - \theta_0}{\sigma^2}(y_i - \frac{\theta_0 + \theta_1}{2}). \tag{4}$$

To detect the fall, at each instant time, the following decision rule is utilized:

$$g_n = S_n - \mu_n \geq h, \text{ where } \mu_n = \min_{1 \geq j \geq n} S_j. \tag{5}$$

The stopping time, denoted as t_a, is

$$t_a = \min\{n : g_n \geq h\} \tag{6}$$

and can be rewritten as

$$t_a = \min\{n : S_n \geq \mu_n + h\}. \tag{7}$$

The above decision rule compares the cumulative sum S_n and the adaptive thresholding $\mu_n + h$, since the μ_n keeps memory of the past observations whereas, the parameter h must be specified. The parameter h is critical, affects the decision rule threshold and therefore the signal change correctness.

3   Proposed Method and Experimental Results

In this section, we describe the technical details of the proposed methodology and we examine its efficiency on processing mobile accelerometer data for on-line fall detection. To construct the training set, accelerometer data were recorded using a smartphone that was in the pocket or purse of walking and falling volunteers. The sampling time during the motion is divided to timesteps, each timestep corresponds to the acquired accelerometer sensor data (approximately
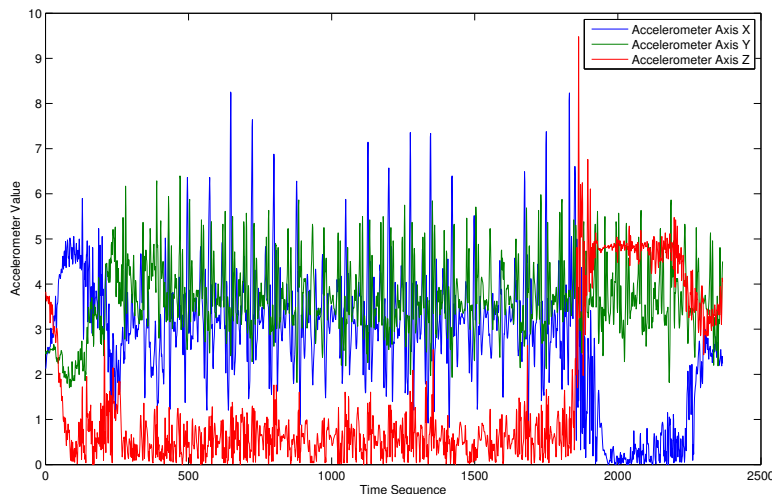
Fig. 1. The time sequence of the three accelerometer axes.

12ms). Consequently, using the same mobile device and settings, the test set was constructed.

To detect the change in the data, the CuSum algorithm needs the mean values of the walking and falling states. Using the training set experiments, we can easily compute these mean values. Additionally, we have supplied the CuSum algorithm with a value for the parameter h. Since different values may lead to different results, a wide range of possible values of h parameter was tested. Below, we report the most promising region for h values and the corresponding results (Table 1).

Figure 1 illustrates an example of raw data recorded by the 3-axis accelerometer. In this experiment, an actual fall occurred approximately at timestep $t = 1800$. Note that the three axes are not equally sensitive regarding the fall; in this experiment y-axis seems to be less informative. Additionally, because of the possible rotation of the device during the fall, there also exists the problem of accelerometer values to shift from one axis to another.

As a first experiment, since the CuSum algorithm can be applied only on one dimensional input data, we apply it to each of the three axis independently. Figure 4 exhibits the data recorded along the x-axis (left) and the output of the CuSum algorithm (right). Note that the actual as well as the detected fall is indicated by a red dot. Figures 5 and 6 show similar plots regarding the y-axis and z-axis, respectively. In this experiment, as it was expected, if the CuSum algorithm is executed using as input the x-axis (or z-axis) data, the fall is reliably detected. On the other hand, y-axis data do not provide enough information and the CuSum fails to detect the fall incidence.
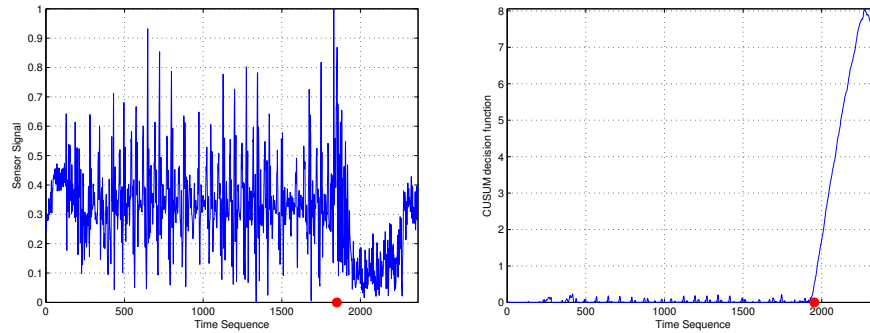
Fig. 2. Data series of the accelerometer data projected on the first PCA eigenvector (left) and the corresponding CuSum decision function (right). The red dot denotes the actual and the detected fall. In this instance, the CuSum algorithm successfully detected the fall.

Fig. 3. Data series of the accelerometer data projected on the first IPCA eigenvector (left) and the corresponding CuSum decision function (right). The red dot denotes the actual and the detected fall. In this instance, the CuSum algorithm successfully detected the fall.
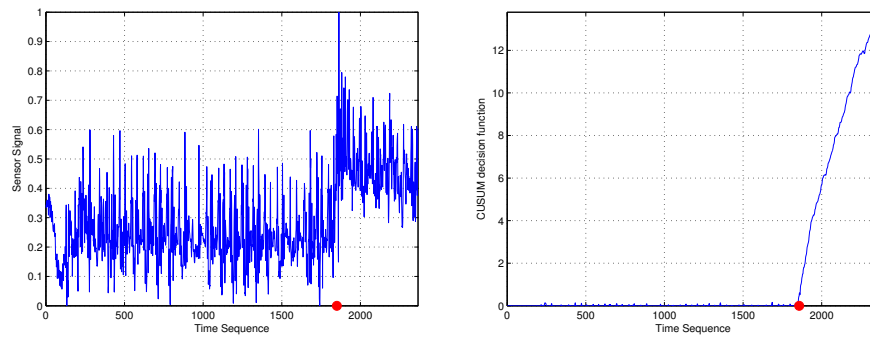
Fig. 4. Data series of the accelerometer x-axis (left) and the corresponding CuSum decision function (right). The red dot denotes the actual and the detected fall. In this instance, the CuSum algorithm successfully detected the fall.
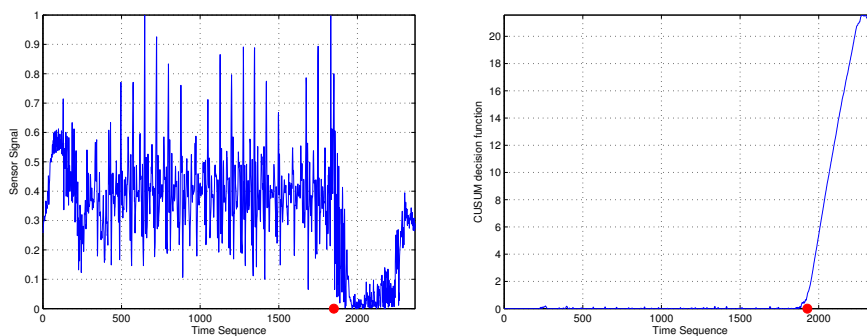

Fig. 5. Data series of the accelerometer y-axis (left) and the corresponding CuSum decision function (right). The red dot denotes the actual fall. In this instance, the CuSum algorithm failed to detect the fall.

Fig. 6. Data series of the accelerometer z-axis (left) and the corresponding CuSum decision function (right). The red dot denotes the actual and the detected fall. In this instance, the CuSum algorithm successfully detected the fall.
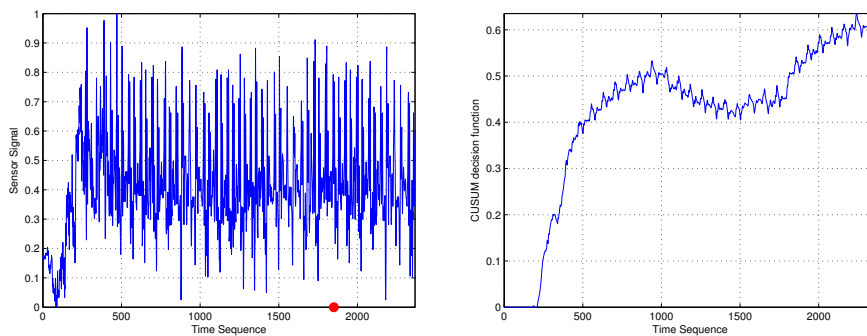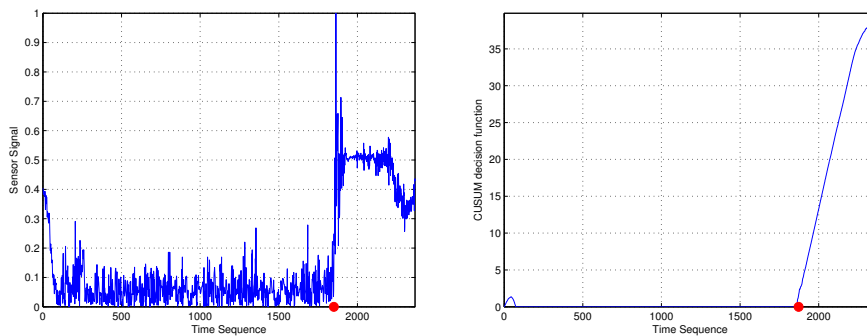
Table 1. Experimental Results

h	0.010	0.050	0.100	0.150	0.200	0.500	1.000	1.500	2.000
Accelerometer Axis X									
False Alarm	112	27	9	5	1	1	0	0	0
Delay (timesteps)	6	7	28	39	39	54	78	91	101
Accelerometer Axis Y									
False Alarm	-	-	-	-	-	-	-	-	-
Delay (timesteps)	-	-	-	-	-	-	-	-	-
Accelerometer Axis Z									
False Alarm	21	13	9	6	5	2	1	0	0
Delay (timesteps)	7	8	8	9	9	15	17	22	26
PCA Projection									
False Alarm	130	47	18	10	3	0	0	0	0
Delay (timesteps)	7	40	44	88	92	103	122	141	162
IPCA Projection									
False Alarm	88	29	11	2	0	0	0	0	0
Delay (timesteps)	1	1	3	4	5	7	18	31	45

To alleviate the problem of having to choose (for each experiment) the correct axis that captures adequate information regarding the fall, we have employed the Principal Component Analysis (PCA) method. Using the PCA method, the 3-dimensional signal of various walking and falling cases is utilized to compute the corresponding space of eigenvectors. Next, the new input data are projected on the direction of the eigenvector that corresponds to the largest eigenvalue. To this end, the transformed signal becomes one dimensional, capturing the most significant information of the original signal. Finally, the CuSum algorithm is applied to detect the change. Figure 2 exhibits a successfully detected fall.

Although this approach gives relatively good results, it requires the PCA method, which demands a batch computation step for the calculation of the eigenspace. Since in this study our aim is the **on-line** fall detection, we employ the Incremental Principal Component Analysis (IPCA), in an attempt to achieve higher classification results using sensor data streams. As a first step, the classic PCA method is applied on the training set (comprised of many walking and falling samples) to compute the initial eigenspace. Subsequently, each incoming input signal updates the eigenspace and is projected on the direction of the eigenvector corresponding to the largest eigenvalue. The transformed one-dimensional data series is then fed to the CuSum algorithm to detect changes, i.e. falls. Figure 3 illustrates the transformed signal and the output of the CuSum algorithm. In this instance, the fall was successfully detected.

Finally, to further evaluate the efficiency of the proposed methodology we use two metrics; the number of "false alarms", which is the number of times that the algorithm mistakenly reports a fall and the "delay", i.e. the time elapsed between the actual and the detected fall. In Table 1 the results of CuSum algorithm for the different scenarios discussed above are presented. When the x-axis or the z-axis were used, correct fall detection was possible for specific values of

parameter h. On the other hand, y-axis provided no information regarding the fall, so the CuSum algorithm failed [13]. The PCA method, utilizing information from the three accelerometer axis, gave better results (less false alarms, lower delay). However, the h parameter still needs fine tunning. Finally, the proposed methodology seems more robust to different parameter h values. The range of acceptable h values can lead to real time fall detection without false alarms.

As a general remark it must be noted that larger values for the parameter h may lead the CuSum algorithm to be more accurate (less false alarms), but increase the delay, as well. On the other hand, lower h values reduce the delay, but introduce more false alarms. Thus, the proposed algorithm having a wider region of acceptable h values can be successfully utilized to on-line real time fall detection.

4   Conclusion

In this work, we present an on-line real time fall detection methodology that detects falls utilizing the 3-axis accelerometer data obtained by the sensor of a smartphone. Human falls were detected using signal change detection by the Cumulative Sum (CuSum) statistical data mining algorithm.

Original 3-axis data were projected on to a suitable direction computed by the Incremental Principal Component Analysis. The experimental results indicate that the proposed methodology is capable to detect falls in real-time. In a future work we intend to incorporate additional sensors (e.g. microphone) to improve the accuracy of our system and enhance the smartphone application. In addition, we will apply our methodology to public fall datasets and make a comparative analysis with other methods.

References

1. Doukas, C., Athanasiou, L., Fakos, K., Maglogiannis, I.: Advanced sound and distress speech expression classification for human status awareness in assistive environments. The Journal on Information Technology in Healthcare 7(2), 111–117 (2009)
2. Doukas, C., Maglogiannis, I.: Advanced classification and rules-based evaluation of motion, visual and biosignal data for patient fall incident detection. International Journal on Artificial Intelligence Tools 19(2), 175–191 (2010)
3. Khawandi, S., Daya, B., Chauvet, P.: Real time recognition of elderly daily activity using fuzzy logic through fusion of motion and location data. International Journal of Computer Applications 54(3), 55–60 (September 2012)

4. Lee, R.Y.W., Carlisle, A.J.: Detection of falls using accelerometers and mobile phone technology. Age and Ageing 40(6), 690–696 (2011)
5. Melzer, I., Benjuya, N., Kaplanski, J.: Postural stability in the elderly: a comparison between fallers and non-fallers. Age Ageing 33, 602–607 (2004)
6. Miaou, S., Sung, P.H., Huang, C.Y.: A customized human fall detection system using omni-camera images and personal information. In: Distributed Diagnosis and Home Healthcare. pp. 39–42. IEEE Computer Society (2006)
7. Noury, N., Herve, T., Rialle, V., Virone, G., Mercier, E., Morey, G., Moro, A., Porcheron, T.: Monitoring behavior in home using a smart fall sensor and position sensors. In: Microtechnologies in Medicine and Biology, 1st Annual International, Conference On. 2000. pp. 607–610 (2000)
8. Oja, E., Karhunen, J.: On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix. Journal of Mathematical Analysis and Applications 106, 69–84 (1985)
9. Ong, T., Tan, W., Marshall, L., Sahota, O.: The relationship between socioeconomic status and fracture in a fracture clinic setting: Data from the nottingham fracture liaison service. Injury (in press)
10. Ozdemir, A.T., Barshan, B.: Detecting falls with wearable sensors using machine learning techniques. Sensors 14(6), 10691–10708 (2014)
11. Page, E.S.: Continuous Inspection Schemes. Biometrika 41(1/2), 100–115 (1954)
12. Sanger, T.D.: Optimal unsupervised learning in a single-layer linear feedforward neural network. Neural Networks 2(6), 459–473 (1989)
13. Tasoulis, S.K., Doukas, C.N., Plagianakos, V.P., Maglogiannis, I.: Statistical data mining of streaming motion data for activity and fall recognition in assistive environments. Neurocomputing 107, 87–96 (May 2013)
14. Weng, J., Zhang, Y., shiuan Hwang, W.: Candid covariance-free incremental principal component analysis. IEEE Trans. Pattern Analysis and Machine Intelligence 25, 1034–1040 (2003)
15. West, J., Hippisley-Cox, J., Coupland, C.A., Price, G.M., Groom, L.M., Kendrick, D., Webber, E.: Do rates of hospital admission for falls and hip fracture in elderly people vary by socio-economic status? Public Health 118, 576–581 (Dec 2004)
16. Zhang, Y., Weng, J.: Convergence analysis of complementary candid incremental principal component analysis. Tech. rep., Comput. Sci. Eng., Michigan State Univ., East (2001)