Triaxial Accelerometer Located on the Wrist for Elder People's Fall Detection

Armando Collado Villaverde, María D. R-Moreno, and David F. Barrero

Universidad de Alcalá, Departamento de Automática
Crta. Madrid-Barcelona, Alcalá de Henares, Madrid, Spain

Abstract. The loss of motor function in the elderly makes this population prone to falls, and their fragility make the consequences of falls very serious. Actually, falls are one of the most notable concerns in elder care. Not surprising, there are several technical solutions to detect falls, however none of them have achieved great acceptance.
The popularization of smart watches provides a promising tool to address this problem. We present a solution that applies Machine Learning techniques to process the output of a smart watch accelerometer, being able to detect a fall event with high accuracy. To this end, we simulated the two most common types of falls in elders, gathering acceleration data from the wrist, then apply that data to train two classifiers. The results show high accuracy classifiers able to detect falls.

Keywords: Fall detection, accelerometer, Machine Learning, Classification, Supervised Learning, care for the elderly

1 Introduction

Falls in the elderly are a public health problem [1]. They are a significant source of problems associated to elderly for their direct consequences (such as traumas), but also fo the symptom of infirmity (such as hearth attack). Therefore, it is not surprising that falls are one of the most relevant concerns for elders care professionals and families.

The importance of this topic has motivated the rise of a notable number of solution proposals. Most of them have in common the usage of accelerometers [2]. The small size, availability in cell phones and respect to privacy explain why they are becoming so popular in falls detection. Many approaches use dedicated devices, usually placed on the trunk [3, 4], while others exploit the capabilities and popularity of smart phones [5–7].

Image processing is another popular approach to fall detection [8–10], however it poses some practical problems which in this context are determinant. People use to dislike having cameras in their private rooms, even when they do not record or transmit images. We should also mention the need to install cameras and their limitations to the screened area. We can distinguish a third groups of fall detection systems based on sound or vibrations [11].

Perhaps a notable motivation for elders to reject fall detection devices is their size, which leads to inadequate ergonomics [2]. Fall detection based on cell phones do not present that problem, but raises new ones when they are used by elders. Perhaps the most notable one is that they do not keep the cell phone on them when being at home, where most of the falls happen. There are other additional problems, for instance, women use to keep their cell phones in a handbag, where fall detection algorithms will likely fail because they are trained to detect falls through acceleration sensors close to the body trunk.

In order to overcome the usage disadvantages of previous devices, we propose to exploit the popularity of smart watches. Most of them are programmable devices, and include rich sensing such as accerometers. Some advanced models even include heart monitors and communication capabilities. Those features all together with the price reduction give a good opportunity of improving elders caring. Some of the problems with cell phones do not happen when using smart watches: they are located always in the same place, regardless of gender or age, and perhaps more important in elders, they are every day objects and thus they are not perceived as something invasive. Then, we believe these features will reduce the adoption resistances. In this paper we present a Machine Learning (ML)-based fall-detection algorithm designed to be implemented in a smart watch. Results show the accuracy of the classifiers used over the datasets generated under the supervision of professionals in the field.

The paper begins with a description of the types of falls in elders, then it describes the data acquisition procedure. Section 4 describes the data preprocessing. The main contribution is located in section 5, which describes the training and evaluation of the falls detection algorithm. The robustness of the proposed algorithm is evaluated in section 5.4. The paper finishes with conclusions and future work.

2 Types of falls in elders

Our goal is to implement a falls detection algorithm in a smart watch to monitor the elderly. In order to detect the falls, we pose the problem in terms of classification: given the acceleration values in a time window, classify them as corresponding to a fall or not. Since the final goal is to implement the classifier in a capabilities limited device (a smart watch), the classifier resources consumption is an issue to take into account.

As most ML applications, the big issue is how to gather high quality data to train the classifier. In this particular application data gathering involves people falling, which implies obvious health risks. Other approaches have used volunteers to simulate the fall, who used to be healthy young people, sometimes skill in some martial art or using protections to minimize the risks. This clearly bias the result, however, data gathering of real falls with elder people is a costly, risky and time consuming task [12], just to mention the most obvious difficulties.

The target of our work is elder care, whose falls follow specific dynamics. Elders suffer loose of mobility, which translates to slower motion and increased

reaction time. In case of a fall, a young healthy person would react moving his/her arms to cushion the hit; on the contrary, elders do not tend to react in time, resulting in more violent hits. Another relevant issue for our work is the shift of the center of gravity in elders. With age, people tend to separate their legs, and curve the trunk forward, this implies that the center of gravity in elders tend to be lower and shifted forward. The consequence is that falls in elders rarely happen laterally or backwards.

There are usually two types of falls in elders, which we name syncope and forward falls. We refer to *syncope falls* to those falls consequence of a loss of conscience or hearth condition that prevent using the muscles to control the fall. It results in a vertical motion and a two stages fall: first the knees impact on the ground, and then the trunk moves forward until it hits the ground. The second type of fall usually found in elders is what we name a *forward falls*. They are originated by the collision of a foot with an object while the person is walking, loosing the equilibrium and falling down. The trunk in this type of falls moves forward, and given the increased reaction time of elders, the trunk hits the ground without the hands cushioning.

Given the different dynamics of the falls, it seems reasonable to address them as two different, yet related, problems. To this end we will train two different classifiers, each one specialized in detecting one type of fall. In section 5.4 we study the ability of the classifiers to detect falls of a different type they were trained to.

3 Data acquisition

Acquisition of high quality datasets is a key process in ML. We used a smart watch with a triaxial accelerometer, the hardware imposed a sampling period of 20ms, yielding three measures of acceleration (X, Y and Z) each 20 ms. Y stands for the longitudinal axis, X for the sideways axis and Z for the axis perpendicular to the watch display.

A problem to consider is how to build the base class labeled as 'no fall'. We used data captured along a basket match, removing those samples containing little accelerations. This is clearly an unrealistic activity for an elder, but basket contains numerous vertical and horizontal motion, making it similar to a fall. In this way we avoid the naïve problem of just classing motion and lack of motion. If a classifier is correctly trained to distinguish between basket and falls, it seems fair to assume it will be able to distinguish between a fall and normal activities in the life of an elder.

Using real falls was discarded given the risk of injuring the subject, specially when our target is a fragile population, the elders. Therefore we tried to capture data as much realistically as possible, taking measures to avoid risks. We defined to data acquisition procedures for syncope and forward falls.

3.1 Syncope falls data capture procedure

Syncope falls are characterized by the lack of control, with gravity as the only acting force. Other ML approaches to fall detection used volunteers to simulate this type of fall. In our opinion, this scenario is better simulated by using a nursing mannequin, which is a mannequin with the same joints mobility, size and weight than an adult human. Of course, the results will also be biased, since there is only one mannequin, but the fall is more realistic than a conscious person.

The center of gravity of the mannequin does not reflects well the one found in an elder. For this reason, just releasing the mannequin does not generate a realistic syncope fall; the mannequin tends to stop once the knees hit the ground. In order to generate realistic falls, the mannequin was smoothly pushed forward when it was released. The whole process was supervised by two Geriatrics experts. They helped to optimize the procedure and judged which simulated falls were realistic, and which one should be discarded.

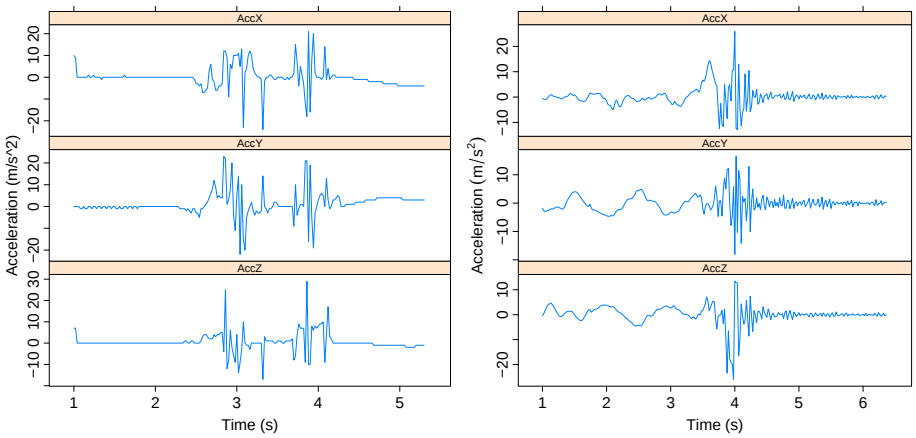
We simulated 42 syncope falls, but the experts only validated 30 falls. 12 simulated falls were discarded for different reasons: in some cases the mannequin did not fall on the knees, or it hit the ground with the knees, but the trunk did not move forward, or the trunk moved laterally. Falls lasted around 500ms, but measurements begun shortly before and finished shortly after the fall. Given that each sample contains three acceleration values, and samples are measured each 20ms, each fall generated around 150 - 300 measures.

3.2 Forward fall data capture procedure

Forward falls begins with the subject walking, when he hits an obstacle over-balancing and falling. This scenario is poorly approximated with a mannequin. In a forward example the subject does have some control, and actually the natural reaction is to raise the hand to cushion the hit. In elders this reaction is slow, and usually they do not have time enough to raise their hand, resulting in more dangerous hits. In our opinion this scenario is better approximated using a subject trained to move slowly. This is not easy, because falling in that way seems unnatural for the subject, but the result is more accurate in opinion of the geriatric experts.

We selected a young volunteer, and placed him on a tatami for safety. The obstacle was a thick pad, which also served to safety stop the falling. The experts in Geriatrics trained the volunteer not to use his hands and move slowly. Once the training was finished, the data capture begun. To simulate the fall, the volunteer begun to walk and after 4-5 steps hit the pad with a foot, falling forward. Given that data might be affected by which foot hit the pad, we repeated the process the same number of times with each foot.

The volunteer simulated 47 falls, but the experts validated 40, 20 for each feet. In order to assess the robustness of the classifier, we gathered data from 20 valid falls of two other volunteers. The duration of the fall is around one second.



(a) Syncope fall acceleration (b) Forward fall accelerations

Fig. 1: Example of accelerations measured on the wrist.

Figure 1 visualizes the acceleration in a typical syncope (a) and forward fall (b). In the syncope fall the two hits (knees and trunk) are clearly visible, while the forward fall shows a first a smooth accelerations due to walking, the fall down and finally the subject laying.

4 Data preprocessing

Data needs some preprocessing in order to feed the classifier. We grouped data in time windows, which contains the samples that serve as input in the classifier. This is also an indirect way to consider history, since the time window contains historical values of acceleration. All windows containing a fall were manually labeled as 'fall', while windows coming from a basket match and those ones that did not contain a fall were labeled as 'not-fall'.

The time window width is a key parameter, we set the width to contain the fall values. We analyzed several syncope falls, observing that the average duration is 500ms, so we set the time window for syncope fall detection to 500ms. Similarly, the window length for forward falls was set to 1200ms.

In addition to raw data coming from the accelerometer, we introduced new attributed summarizing those values. In particular, for each window, we computed the mean and standard deviation of acceleration in each one of the three axis. Those attributes, along with raw data were used to train the classifiers. Table 1 summarizes the attributes considered, however their predictive power greatly varies, as will be analyzed later.

An important issue about data is that it is unbalanced. Since falls are hard to simulate, there were much more data coming from basket than from simulated

Table 1: Attributes under consideration to feed the classifiers: Acceleration in X, Y and Z axis along with mean and standard deviation for each axis. N is the number of samples in the window, which depends on the dataset.

Attribute	Label	Num. attributes
AccelX	AccelX[X_1, ...,x_N]	N
Acceleration Y	AccelY[y_1, ...,y_N]	N
Acceleration Z	AccelZ[Z_1, ...,z_N]	N
Mean X, Y and Z	MeanX, MeanY, MeanZ	3
Std. deviation X, Y and Z	DevX, DevY, DevZ	3

falls. To face this issue, we undersampled the 'not-fall' class, getting the same number of instances for each class. The evaluation of all the classifiers used 10-fold cross-validation.

5 Detection of fall events

Fall detection is addressed as a binary classification problem: classify acceleration measures contained in a time window as 'fall' or 'not-fall'. The dataset was build as described in section 3 and then preprocessed to generate time windows, derived attributes and labels as described in section 4.

The goal is to implement the classifier in a smart watch, which means that the classifier should be as lightweight as possible. We considered some classical classifiers implemented in Weka: C4.5, 1-NN, Logistic regression, Naïve Bayes and PART. Some of these classifiers such as 1-NN are not lightweight, but its good performance motivated us to include them for comparison purposes.

5.1 Determination of sampling rate

An important parameter to be set is the sampling rate. Measures were taken with a hardware that imposed a maximum sampling rate of 20ms, however, it is possible that a good classification could be made using a lower rate. We should consider that there is a direct relationship between the sampling rate and the number of attributes and therefore trying to reduce the sampling rate might pay off.

We briefly study the influence of the sampling rate with the accuracy of the classifiers. To this end we evaluated the accuracy of several classifiers using several sample periods. No feature selection was used. The result is shown in Table 2. We can observe a pattern regardless of the classification algorithm, high sampling rates boost performance, at least in the range of values that we have under consideration. The Nyquist Theorem states that there must be a lower bound to the sampling period from which we should not expect performance improvements. Clearly this bound has not been achieved as the best performance is achieved taking samples each 20ms. Then, this is the value we choose.

Table 2: Evaluation of the influence of the sampling rate on the classifiers accuracy. The table shows the sampling rate, type of fall -syncope (S) or forward (F)-, number of attributes (as indicated in Table 1) and accuracy of the classifiers.

Sample period	Fall type	# attrib	C4.5	1-NN	Reg. Log.	Naïve Bayes	PART
20ms	S	97	95.38%	97.80%	90.26%	85.71%	95.82%
40ms	S	52	92.10%	97.32%	87.48%	84.35%	91.65%
60ms	S	37	88.38%	92.03%	87.24%	83.60%	87.70%
20ms	F	82	95.66%	98.43%	88.76%	86.74%	94.09%
40ms	F	46	91.97%	97.21%	89.34%	84.43%	92.62%
60ms	F	34	88.45%	89.50%	84.25%	83.46%	85.04%

Table 3: Twelve best predictive attributes ranked by its correlation to the class for syncope and forward falls. Correlation among the attributes are not considered.

Forward fall				Syncope fall			
% Inf.	Attrib.	% Inf.	Attrib.	% Inf.	Attrib.	% Inf.	Attrib.
0.515	DevZ	0.173	AccelX1	0.541	DevY	0.233	AccelX11
0.429	DevY	0.163	MeanZ	0.523	MeanZ	0.232	AccelX8
0.399	MeanX	0.159	AccelZ24	0.238	AccelY12	0.229	AccelY11
0.316	DevX	0.154	AccelX2	0.235	AccelX7	0.229	AccelX9
0.209	MeanY	0.144	AccelZ23	0.235	AccelX12	0.227	AccelX10
0.195	AccelX0	0.135	AccelX3	0.233	AccelX13	0.226	AccelX14

5.2 Overview of attributes classification power

The number of attributes used so far is relatively high. To illustrate this point, consider a syncope fall window that last 500ms, containing 25 samples, each sample with three values of acceleration (X, Y, Z), which yields 75 attributes. In addition, there are six derived attributes (mean and standard deviations), resulting in 81 attributes. This high number of attributes may result in higher computational costs and eventually also may degenerate the classifier performance.

In order to reduce the number of attributes we estimated the predictive power of the attributes. To this end we applied Correlation Feature Selection Subset Evaluation [13], as implemented in Weka, This method evaluates the correlation of each attribute with the class, and the correlation among the attributes as well, giving better ranks to those attributes highly correlated with the class and low correlation among other attributes.

Table 3 ranks attributes by its predictive power for both types of falls, syncope and forward. The table suggests that derived attributes have higher predictive power in comparison to raw acceleration values. In particular the standard deviation on Y has the best score for syncope falls, and gets the second position for forward falls. Mean acceleration on Z has the second highest score in syncope

Table 4: Performance of syncope (S) and forward falls detection with the number of selected attributes. Attribute selection was performed using a wrapper approach.

		C4.5	1-NN	Log. Reg.	Naïve Bayes	PART
Accuracy	S	0.98	1	0.92	0.93	0.96
	F	0.98	1	0.89	0.91	0.95
Recall	S	0.98	1	0.94	0.90	0.97
	F	0.98	1	0.91	0.92	0.97
Attributes	S	12	7	16	9	7
	F	9	13	11	12	7

falls, while appears as a good attribute for forward falls. Raw data seems to have less predictive power, in particular on Z. Interestingly, the mean Z acceleration get a pretty high score for syncope falls, while raw acceleration values on Z does not appear in the table. Syncope fall includes many raw acceleration values on X, with similar scores, this suggests that those attributes may be highly correlated.

The high score that derived attributes achieves and the hint of highly correlated raw acceleration values suggest that the number of attributes may be reduced significantly. For this reason in the following we will evaluate the classifiers integrating the feature selection.

5.3 Evaluation of classifiers

Given the importance of the feature selection, we have performed the classifier evaluation along with the feature selection using a wrapper [14] approach. This method exploits the iteraction between the classifier and the attributes, yielding, in theory, to better results, specially where there are a high number of redundant attributes, as this is the case. We have used the Weka `WrapperSubsetEval` implementation of the method. The attribute search algorithm was Hill Climbing.

Table 4 summarizes the performance of the classifiers. For instance, C4.5 with 97 attributes (Table 2) scores 95.38% accuracy, while using wrapper feature selection it increases to 98%, with only 12 or 9 attributes, depending on the type of fall. The other classifiers behave similarly. The 1-NN classifier has an outstanding performance, with a perfect accuracy and recall. The similarity of the instances in the training set may explain this; the robustness analysis done in the next section supports this hypothesis. Despite the magnificent performance of 1-NN, the need to store all the training set dissuades us to implement it in a smart watch. However, it suggests that perhaps using a Nearest Centroid Classifier may conduct to a good classifier while keeping low computational needs.

5.4 Robustness analysis of forward falls

A clear weakness of the previous approach is the lack of diversity in the training set. Syncope falls used just a single mannequin, while forward falls included falls

Table 5: Performance of forward falls detection classifiers shown in Table 4 evaluated with a testing set composed by unseen people simulated falls.

		C4.5	1-NN	Log. Reg.	Naïve Bayes	PART
Accuracy	F	0.91	0.66	0.97	0.98	0.98
Recall	F	0.90	0.98	0.95	0.96	0.98
Attributes	F	9	13	11	12	7

from one personal. This obviously reduces the complexity of the problem, and a natural question that rises is how much the classifier degrades its performance when exposed to different people. In order to provide an insight to this question, we performed a robustness experiment.

Given the lack of alternative mannequin, we focused on forward falls. As described in section 3, we captured data from three people, one of them repeated 40 times the fall simulation, while the others repeated the simulation 20. We exposed the classifiers trained with data coming from the first volunteer (whose performance is shown in Table 4), to the simulated falls of the other two volunteers. The resulting performance is shown in Table 5. As we expected, the performance drops, but in most cases remain above 0.9. The most dramatic case is 1-NN, whose accuracy dramatically reduces to 0.66. Logistic regression, Bayes and PART seems quite robust and actually they increase the performance.

6 Conclusions and future work

In this paper we have described an ML application to detect falls sensing the acceleration on the wrist. The goal is to implement a fall detection system in a smart watch oriented to elders care. This population is prone to suffer two types of falls: syncope and forward falls. We simulated those type of falls and measured acceleration on the wrist. These data, along with measures coming from a basket match were used to train and evaluate a classifier with a wrapper-based feature selection.

The accuracy of the classifiers were quite high, above 0.9 in all the algorithms under consideration. The number of selected attributes remained low, between 7 and 17, depending on the algorithms and type of fall. Interestingly the classifiers showed high robustness, they were trained with data from one person, but their performance remained pretty high when detecting falls from unseen people, in some cases it even increased.

In a near future we expect to expand the detection with new sensors. In particular we plan to ensemble fall detectors based on different sensing, in particular imaging and sound.

7 Acknowledgements

The authors thank the contribution of Isabel Pascual Benito, Francisco López Martínez and Helena Hernández Martínez, from Department of Nursing and

Physiotherapy of the University of Alcalá, for their help designing and supervising the simulated falls procedure. This work is supported by UAH (2015/00297/001), JCLM (PEII-2014-015-A) and MINECO (TIN2014-56494-C4-4-P).

References

1. Sadigh, S., Reimers, A., Andersson, R., Laflamme, L.: Falls and fall-related injuries among the elderly: a survey of residential-care facilities in a swedish municipality. Journal of community health **29** (2004) 129–140
2. Noury, N., Fleury, a., Rumeau, P., a.K. Bourke, Laighin, G.O., Rialle, V., Lundy, J.E.: Fall detection - Principles and Methods. 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (2007) 1663–1666
3. Ravi, N., Dandekar, N., Mysore, P., Littman, M.L.: Activity recognition from accelerometer data. In: AAAI. Volume 5. (2005) 1541–1546
4. Gibson, R.M., Amira, A., Ramzan, N., Casaseca-de-la higuera, P., Pervez, Z.: Multiple comparator classifier framework for accelerometer-based fall detection and diagnostic. Applied Soft Computing Journal **39** (2016) 94–103
5. Luštrek, M., Kaluža, B.: Fall Detection and Activity Recognition with Machine Learning. Informatica **33** (2008) 205–212
6. Albert, M.V., Kording, K., Herrmann, M., Jayaraman, A.: Fall classification by machine learning using mobile phones. PLoS ONE **7** (2012) 3–8
7. Zhang, T., Wang, J., Liu, P., Hou, J.: Fall Detection by Embedding an Accelerometer in Cellphone and Using. Journal of Computer Science **6** (2006) 277–284
8. Miaou, S.G., Sung, P.H., Huang, C.Y.: A customized human fall detection system using omni-camera images and personal information. Conference Proceedings - 1st Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare, D2H2 2006 **2006** (2006) 39–42
9. Cucchiara, R., Prati, A., Vezzani, R., Emilia, R.: and Alarm Generation. Expert Systems **24** (2007) 334–345
10. Auvinet, E., Multon, F., Saint-Arnaud, A., Rousseau, J., Meunier, J.: Fall detection with multiple cameras: an occlusion-resistant method based on 3-D silhouette vertical distribution. IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society **15** (2011) 290–300
11. Zigel, Y., Litvak, D., Gannot, I.: A method for automatic fall detection of elderly people using floor vibrations and soundProof of concept on human mimicking doll falls. IEEE Transactions on Biomedical Engineering **56** (2009) 2858–2867
12. Bagalà, F., Becker, C., Cappello, A., Chiari, L., Aminian, K., Hausdorff, J.M., Zijlstra, W., Klenk, J.: Evaluation of accelerometer-based algorithms on real-world falls. PloS one **7** (2012) e37062
13. Hall, M.A.: Correlation-based Feature Subset Selection for Machine Learning. PhD thesis, University of Waikato, Hamilton, New Zealand (1998)
14. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial intelligence **97** (1997) 273–324