A WEARABLE REAL-TIME FALL DETECTOR BASED ON NAIVE BAYES CLASSIFIER

Xiuxin Yang, Anh Dinh, Li Chen

Department of Electrical and Computer Engineering
University of Saskatchewan, Saskatoon, Canada

ABSTRACT

In this paper, we implement a wearable real-time system using the Sun SPOT wireless sensors embedded with Naive Bayes algorithm to detect fall. Naive Bayes algorithm is demonstrated to be better than other algorithms both in accuracy performance and model building time in this particular application. At 20Hz sampling rate, two Sun SPOT sensors attached to the chest and the thigh provide acceleration information to detect forward, backward, leftward and rightward falls with 100% accuracy as well as overall 87.5% sensitivity.

Index Terms— fall detection, Naive Bayes classifier, Sun SPOT, accelerometer, machine learning

1. INTRODUCTION

Fall detection has become a hot research topic due to the increasing population over 65 throughout the world, as well as the serious effects and problems caused by fall. In 2007, 21.5% Japanese people aged over 65, one-third to one-half of those had experienced falls [1]. In terms of elderly people themselves, hip fractures and head injuries are serious effects which may cause accidental death. For government and medical organizations, an effective approach to avoid and prevent fall among the elderly is able to reduce expenditure and decrease the pressure of shortage of healthcare personnel. Thus a system designed for fall detection is essential.

Although three fall detection approaches are proposed in previous research, including wearable-based, camera-based and ambience-based [2], the wearable-based one is widely used due to the flexibility of accelerometers or gyroscopes in measuring people's movement. Tamura et al. [1] employs one tri-axis accelerometer and one gyroscope to detect movement so that the wearable airbag is inflated once a fall occurs.

The final objective of fall detection is to generate fall alarm and trigger protection system (i.e., an airbag or other means) so that people is properly protected. Some previous researches detect fall based on confirming people's lying position on the ground. While such a detecting strategy maybe have high accuracy, it could do nothing in preventing falls. Thus the practical system should accurately detect potential falls and trigger protection system before they cause serious injuries to people.

In terms of algorithm, most previous researches employ threshold approach. Tamura et al. [1] assume that a fall occurs when the acceleration is less than ±3 m/s^2 and the angular velocity exceeds 0.52 rad/s (approximately 30° /s). The obvious demerit is that calibration is needed in threshold-based method, and fixed threshold can not easily adapt to the variance of different people, which decreases system accuracy. In this work, we use artificial intelligence method to perform fall detection.

2. METHODOLOGY

2.1. System requirements

The basic requirements of designing a wearable real-time system for fall detection are as follows: (i) It must be small, light weight and easy to wear, (ii) It must be able to detect forward, backward, leftward and rightward falls while the wearer is standing or walking, (iii) It must generate fall alarm before a fall really occurs, so that the fall alarm could trigger protection system to prevent fall injuries. Based on the above considerations, the hardware of the system should consist of acceleration sensors which collect people's movement information, and a fall detector which is used to identify potential falls.

2.2. Characteristics of fall

Before we design an actual system to achieve the mentioned goal, the characteristics of fall are discussed for better understanding. Most common falls are divided into three types based on the falling height: fall from lying; fall from sitting; and fall from standing or walking [2]. Since fall from lower height causes less severe effect to the elderly, we only discuss fall from standing or walking, which is the leading cause of bone fracture and related injuries.

Table 1 shows the time of different falls, where all the falls are simulated from a standing position down to 40 cm height mattress. Fall time is defined as the time interval from standing position to people's body completely contacting the mattress. The average falling time is less than 2000 ms, and before people really fall down, the detector should generate an alarm signal which could be used to trigger a protection equipment, such as a wearable airbag, so

978-1-4244-5377-1/10/$26.00 ©2010 IEEE

that the fall and related injures to elderly people are prevented.

Table 1: Time of falls.

	Forward fall (ms)	Backward fall (ms)	Leftward fall (ms)	Rightward fall (ms)
1	1640	1830	1700	1750
2	1510	2060	1560	1700
3	1260	1690	1440	1450
4	1370	1760	1560	1560
5	1190	1630	1620	1500
Average	**1394**	**1794**	**1576**	**1592**

2.3. Training and testing process

This fall detection system consists of two parts: training and testing. Figure 1 shows block diagram of the training and testing process. Acceleration data of simulated falls, standing and walking are used to train the Naive Bayes classifier, where the raw data are firstly cut into small windows with feature selection, then a Gaussian model is built for each attribute. After building up the classifier, unknown acceleration data is processed by the Naive Bayes classifier to acquire detection result.

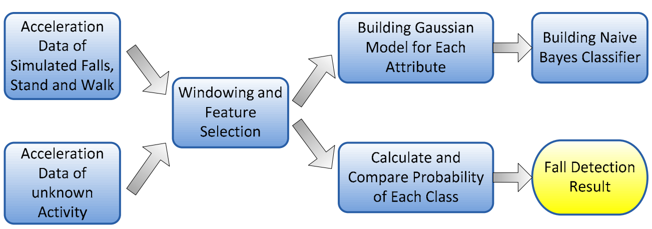

Figure 1: Block diagram of the training and testing process.

2.4. Data pre-processing

Before going through the Naive Bayes classifier, raw data must be pre-processed using a windowing technique in order to increase classification accuracy. Windowing technique is used to divide the sensor signal into smaller time segments (i.e., windows) and classification algorithm is applied on each window, which means each window generates a detection result.

The window size for pre-processing is very important in detection accuracy. Small window will misclassify pattern, while big window has a potential of mis-identifying a fall (if falling time is much less than the window time). After numerous experiments, we find that 0.2 second is a good window time for detecting fall. At 20Hz sampling rate of the acceleration sensors, the window size is 4 samples.

In Figure 2, every 4 samples (i.e., S1 to S4) make up of one window. The mean value is calculated to produce mean_1, mean_2, etc in each window. As time rolls on, the window jumps forward 0.2 second (i.e., 4 samples time). Since this approach is very easy to implement, it is ideally appropriate for real-time application in embedded system.

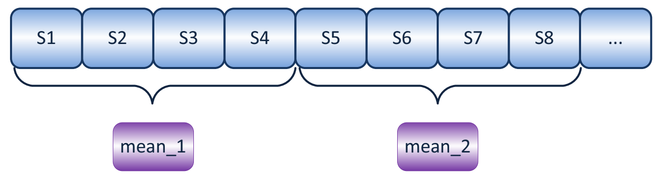

Figure 2: Windowing technique: fixed window size.

2.5. Feature selection

Due to the characteristics of falls, such as maximum or minimum peak in time domain, falls can be differentiated from standing or walking.

Figure 3 indicates the training of falls, where each type of fall is trained 3 times, totally 12 times. The horizontal axis is time, and the vertical axis is the acceleration value before normalization. The peaks in Figure 3 display forward, backward, leftward and rightward fall very well. Forward and backward fall have obvious peaks in acceleration Z axis (yellow color), negative and positive peak respectively; leftward and rightward fall have obvious peaks in acceleration X axis (blue color), positive and negative peak respectively. In addition, acceleration Y axis (red color) can not be used to classify different falls, because each kind of fall has a positive peak in Y axis direction. However, this attribute is able to detect falls from normal physical activity efficiently.

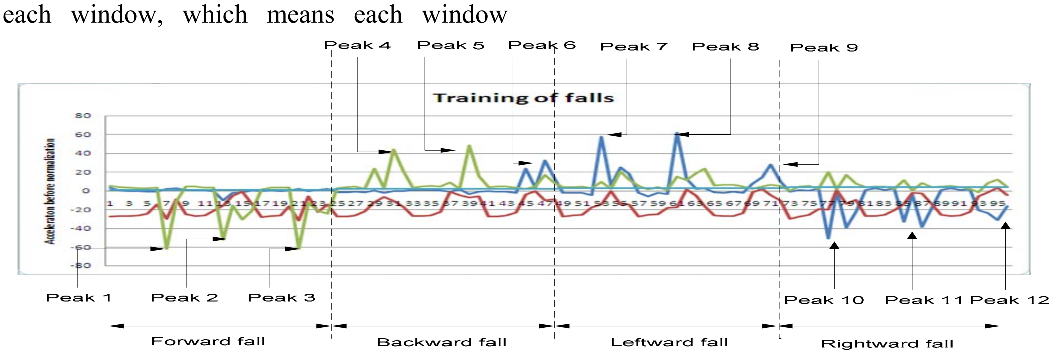

Figure 3: Training of falls. Each peak represents a fall.

2.6. Algorithm comparison using WEKA

It is crucial to select an algorithm to detect fall in real time. The selected algorithm should not only detect falls as accurate as possible, but also generate fall alarm as fast as possible. Low accuracy of fall detection is not able to prevent falls; slow algorithm needs more time for calculation so that protection system becomes ineffective. In order to select an optimal algorithm, we utilize WEKA (Waikato Environment for Knowledge Analysis) [3], developed at the University of Waikato. This workbench contains a collection of visualization tools and algorithms for data analysis and predictive modeling. Machine learning algorithms such as Artificial Neural Network, Decision Tree and Naive Bayes can be easily realized in WEKA.

In Table 2, five popular artificial intelligence algorithms are compared and evaluated in WEKA, and all the 5 algorithms perform with the default setting. Both the training and testing are acceleration data of 5 physical activities (i.e. sit, stand, lie, walk, walk up the stairs) collected from two 3-axis accelerometers attached to the chest and to the right thigh of a subject, and off-line analysis is realized in WEKA. The laptop running WEKA is a 1.6GHz processor with 512 MB memory.

Table 2 shows the comparison result. In terms of accuracy, Naive Bayes algorithm overtakes the others with 88.06% accuracy, while Support Vector Machine (SVM) is only 64.40%. Time taken to build model is a parameter to evaluate the complexity of an algorithm, which is also important for real-time applications. SVM algorithm consumes 2190ms to build the model, while Naive Bayes algorithm consumes very little time. Therefore, Naive Bayes is chosen as the classifier in fall detection, with a relatively high accuracy as well as less processing time. For details of Naive Bayes algorithm please refer to [4].

Table 2: Algorithms Evaluation Result.

Algorithm	Correctly Classified Instances (%)	Incorrectly Classified Instances (out of 486)	Time to build model (millisecond)
SVM	64.40	173	2190
OneR	74.69	123	0
C4.5(J48)	81.07	92	550
Neural Network	84.57	75	920
Naive Bayes	**88.06**	**58**	**0**

3. IMPLEMENTATION ON THE SUN SPOT

The Sun SPOT is a Java programmable embedded device designed for flexibility [5]. The device includes a tri-axis accelerometer with maximum -6G to +6G acceleration range embedded in the sensor board and a 180MHz 32-bit ARM920T microprocessor. In addition, the device is small and has a maximum of 7 hours operation without recharging the battery. The 8 multi-color LEDs on board can be used to display the detection result; no other display equipment is needed. The sensors can communicate among themselves wirelessly to build a wireless body sensor network.

Figure 4 shows the Sun SPOT sensor and how the sensors are attached to a testing subject.

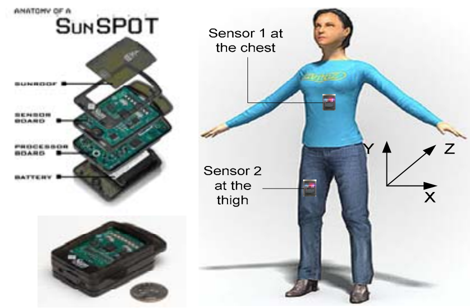

Figure 4: Sun SPOT sensor (left), sensors are attached to a testing subject.

The master sensor (sensor 1) provides 3 attributes (X-axis, Y-axis and Z-axis accelerations), and the slave sensor (sensor 2) provides 1 attribute ($|a| = \sqrt{a_x^2 + a_y^2 + a_z^2}$). In the Gaussian model of each attribute, the mean value is replaced by half the average peak value of the falls of training, as shown in Figure 3. After experiments, we found this modification increased the accuracy of fall detection.

Figure 5 shows the flow chart of the master sensor, which receives the forth attribute from the slave sensor then performs calculation and sends detection results wirelessly to a receiver connected to a computer to record and display results in real time. Switch 1 controls the training process; switch 2 controls the testing process. In the computer terminal, all the training and testing data are received and recorded in a .csv file for further processing. Even without the computer, the user can also acquire the detection results and monitor activity patterns using the on-board LEDs on the master sensor.

4. RESULTS AND DISCUSSION

The fall detection testing is performed by a 23 years old female subject, starting from a still standing position and ending at a mattress of 40 cm height.

The strategy is that each fall is trained three times successively; also stand and walk are trained once to build up the Naive Bayes model. After training, each type of fall is tested 20 times. Once a fall occurs in the testing process, a particular LED on the Sun SPOT board will lit up. At the same time, the acceleration information is transmitted to the laptop terminal wirelessly for real-time monitoring and further processing.

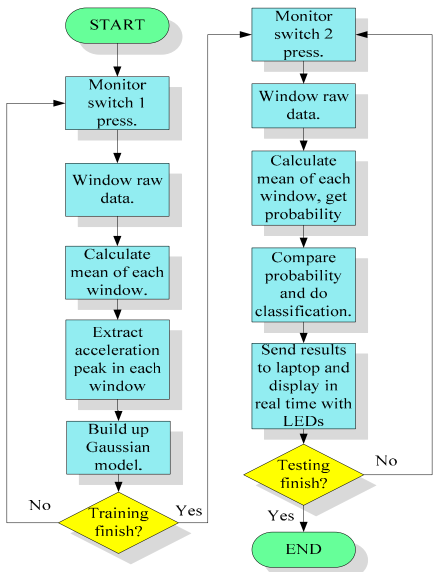

Figure 5: Flow chart of the master sensor.

Table 3 shows the confusion matrix of fall detection results. Eighty-eight falls are successfully detected as falls with 100% accuracy; although one specific fall is possible to be classified as another fall (e.g., forward fall is classified as leftward fall once and backward fall once in Table 3). Normal activities (i.e., stand or walk) are impossible to be recognized as falls, indicating there is no false alarm while user is wearing this system in their daily activities. When evaluating fall detection system, the criteria sensitivity is proposed in [6].

$$Sensitivity = \frac{TP}{TP + FN}$$

where TP (true positive) represents that a fall occurs, and the device detects it; FN (false negative) represents that a fall occurs, but the device does not detect a fall. High sensitivity indicates the fall detector has the ability to detect more actual falls with fewer false alarms.

Table 3: Confusion matrix of fall detection results.

Activity		Classified As					
		A	B	C	D	E	F
A: Forward fall B: Backward fall C: Leftward fall D: Rightward fall E: Stand F: Walk	A	22	1	1	0	0	0
	B	1	18	1	2	0	0
	C	1	0	19	0	0	0
	D	2	1	1	18	0	0
	E	0	0	0	0	100	0
	F	0	0	0	0	2	98

Table 4 shows the statistic of fall detection results, in which 88 falls are successfully detected as falls, and 77 falls are exactly detected as particular falls, with an overall sensitivity of 87.5%.

Table 4: Statistic of fall detection results.

Fall	For-ward	Back-ward	Left-ward	Right-ward	Overall
Exactly detect/Fall	22/24	18/22	19/20	18/22	77/88
Sensitivity %	92	82	95	82	87.5

5. CONCLUSION

The characteristics of fall are discussed for better understanding fall and related fall detection algorithm. After comparing 5 popular artificial intelligence classifiers, Naive Bayes algorithm is proved to work well in both accuracy performance and model building time in this particular application. A wearable real-time fall detector is successfully implemented using the Sun SPOT with Naive Bayes algorithm. The combination of acceleration data collected from two Sun SPOT sensors attached to the chest and thigh is not only able to detect falls, but also able to recognize normal physical activity. The overall fall detection sensitivity is 87.5% with less false negative detections. In additional to fall detection, the acceleration data collected from the sensors can be used to estimate people's daily energy expenditure, also the data themselves as well as the fall detection results can be sent to a remote server through internet to realize telehealth applications.

REFERENCES

[1] T. Tamura, et al, "A Wearable Airbag to Prevent Fall Injuries," *IEEE Transactions on Information Technology in Biomedicine, Volume 13, Issue 6*, pp. 910 – 914, Nov. 2009.
[2] X. Yu, "Approaches and Principles of Fall Detection for Elderly and Patient," *e-health Networking, Applications and Services, 2008. HealthCom 2008. 10th International Conference on*, Singapore, pp. 42 – 47, 7-9 July 2008.
[3] G. Holmes, A. Donkin, and I.H. Witten, "WEKA: A Machine Learning Workbench," *1994 Second Australian and New Zealand Conference on Intelligent Information Systems,* Brisbane, Australia, pp. 357-361, 29 Nov-2 Dec 1994.
[4] X. Yang, A. Dinh, and L. Chen, "Implementation of a Wearable Real-Time System for Physical Activity Recognition Based on Naive Bayes Classifier," *International Conference on Bioinformatics and Biomedical Technology (ICBBT 2010),* China, April 2010.
[5] http://www.sunspotworld.com/Tutorial/index.html,2009.
[6] N. Noury, et al. "Fall Detection – Principle and Methods," *29th Annual International Conference of the IEEE on Engineering in Medicine and Biology Society (EMBS2007),* France, pp.1663 – 1666, 22-26 Aug 2007.