Unsupervised Segmentation and Labeling for Smartphone Acquired Gait Data

Matthew Martinez[1,2], Phillip L. De Leon[1]

[1]**New Mexico State University**
Klipsch School of Elec. & Comp. Eng.
Las Cruces, NM 88003-8001

[2]**Sandia National Laboratories**
Albuquerque, NM 87185

ABSTRACT

As the population ages, prediction of falls risk is becoming an increasingly important research area. Due to built-in inertial sensors and ubiquity, smartphones provide an attractive data collection and computing platform for falls risk prediction and continuous gait monitoring. One challenge in continuous gait monitoring is that significant signal variability exists between individuals with a high falls risk and those with low risk. This variability increases the difficulty in building a universal system which segments and labels changes in signal state. This paper presents a method which uses unsupervised learning techniques to automatically segment a gait signal by computing the dissimilarity between two consecutive windows of data, applying an adaptive threshold algorithm to detect changes in signal state, and using a rule-based gait recognition algorithm to label the data. Using inertial data, the segmentation algorithm is compared against manually segmented data and is capable of achieving recognition rates greater than 71.8%.

Keywords: falls risk, gait, accelerometer, machine learning, smartphone app.

1 INTRODUCTION

As the elderly population sharply increases, the prediction of falls risk has become an important research area since falling is one of the leading causes of injury and death among people over the age of 65 [1]. An individual's risk factor is determined by external and age-related factors, including but not limited to home safety, medications, muscle weakness, and gait deficits. External risk factors can be measured using a variety of assessments [1] that account for fall and medical histories, prescription/non-prescription medications, and home safety. Physiological risk factors, such as gait deficits, can be measured using foot pressure sensors, motion capture systems, and inertial sensors. Measurements from these sensors can be incorporated into biomechanical models, and can be used in machine learning systems to predict an individual's risk of falling [2]. Micro Electro-Mechanical Systems (MEMS) based Inertial Measurement Units (IMUs) are an attractive choice for measuring gait because of their low cost and demonstrated effectiveness in falls prediction [2]. Such IMUs can be found in most smartphones, making them compelling devices for falls research.

An important aspect of smartphone-based falls risk prediction is the ability to automatically isolate gait segments. Automatic and correct gait segmentation aids in both building and testing predictive falls risk models. Furthermore, automated segmentation algorithms enable the ability to continuously monitor an individual's gait for a change in falls risk. One of the challenges in correctly identifying gait segments for falls risk prediction is that gait mechanics change as an individual ages, which include a decrease in step length and step width, which are correlated with the risk of falling [3–5]. Part of the challenge of being able to correctly identify gait segments is that gait degradation results in less smooth transitions between active states. Thus, it can be difficult even with manual (human-based) signal segmentation, to accurately identify the transitions between walking and other types of activities, i.e. turning.

Activity recognition using smartphones and inertial sensor-based systems is an active research area where many solutions for identifying walking, running, bicycling, stair climbing, etc. have been proposed [6]. One of the aspects of activity recognition is the ability to identify a state transition or *change point*, t_{CP}. Several change point detection algorithms for time-series data exist including Bayesian analysis [7–9], Hidden Markov Models (HMMs) [10], direct density ratio estimation [11], and kernel-based models [12].

This paper proposes a new, unsupervised segmentation and labeling algorithm for inertial measurements of gait. Our algorithm is based on the algorithm first presented in [11]. Our contributions include a new approach to change point detection which uses a Direct Density Ratio Estimation (DDRE) technique previously used in other fields such as social networking and activity recognition, [11]. In addition, we have included an adaptive threshold algorithm described in [13] to detect the most relevant change points within the time series to improve robustness in the labeling of gait segments. The proposed algorithm is applied to a data set of inertial measurements acquired using smartphones from individuals with both high- and low-falls risk where the assessment is based on [3–5]. The results of the automatic segmentation are compared and evaluated to manual segmentation. This paper is organized as follows. In section 2, the data collection procedure and segmentation problem are described and in section 3, the DDRE technique, adaptive thresholding and gait recognition algorithms are described. In section 4, we evaluate the gait signal segmentation algorithm. Finally, we provide conclusions.

2 DATA COLLECTION

Gait data was collected by The Electronic Caregiver Co. (ECG), Mobile Fall-Risk Assessment Unit which includes a pressure sensitive walkway and two Apple® iPhone® 6. Each iPhone was running the ECG GaitLogger app [14] which logs inertial measurements of each participant's gait. Raw sensor data is processed on the iPhone using the InvenSense's Digital Motion Processor, which performs 6-axis (accelerometer and gyroscope) sensor fusion [15]. The app logs the iPhones' attitude, unbiased rotation rate, acceleration due to gravity and the device, and device acceleration where acceleration due to gravity is filtered out.

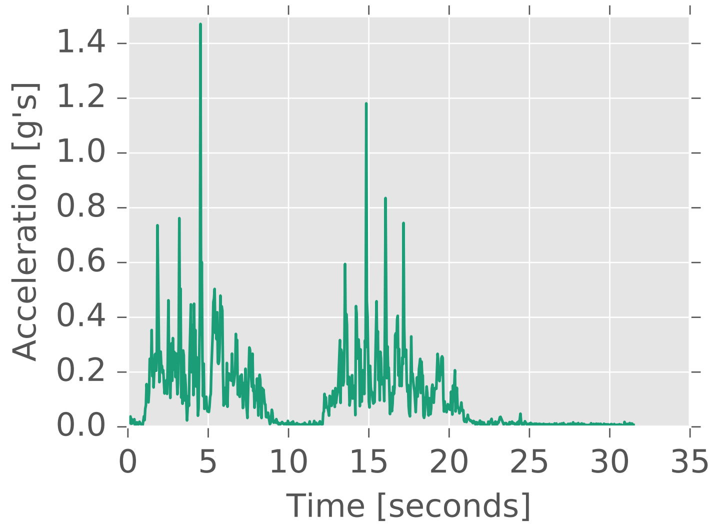


Figure 1: Vector magnitude of acceleration signal for a walking individual showing both the outbound and return segments.

For each data collection session, the participant first completed a comprehensive falls risk screening [1]. The participant then walked down the walkway (outbound), performed a turn, and walked back to the initial starting point (inbound). The logging app ran for approximately 30 seconds, providing the participant ample time to complete the session. In an ideal setting, four change points (start, turn, return, stop) would be present in the data. Figure 1 shows the vector magnitude of the acceleration signal for an individual with a high-falls risk[1]. In both figures, the outbound signal starts at approximately $t = 1$ s and ends at approximately $t = 9$ s, and the inbound signal beings at approximately $t = 12$ s and ends at $t = 21$ s.

3 PROPOSED SEGMENTATION ALGORITHM

The proposed segmentation algorithm consists of three stages illustrated in Figure 2. The input to the algorithm is a multivariate time-series, $\mathbf{y}(t) \in \Re^d$. The first stage of the algorithm measures the *dissimilarity* between two windows of data using the DDRE technique [11, 16]. The assumption is for each activity, i.e. walking, standing and turning, the underlying distribution of the signal data will be dissimilar enough to indicate a possible change point. These change points represent different transition events, i.e. standing to walking, walking to turning, turning to walking, and walking to standing. In the second stage, an adaptive threshold algorithm is applied to the dissimilarity measure, which detects potential change points. The adaptive threshold algorithm is inspired by the constant false alarm rate (CFAR) detector used for radar target detection [13]. For each potential change point, a threshold is calculated from a window of leading and lagging data points. Once a detection has occurred,

[1]Individuals with a high- or low-falls risk are simply referred to as high- or low-risk individuals.

Figure 2: Block diagram of proposed gait signal segmentation algorithm consisting of three stages. The D-dimensional gait signal, $\mathbf{y}(t)$ is fed into the algorithm and a set of times at which change points occur, $\{t_{CP}\}$ are returned.

a peak detection algorithm is used to find the local maximum, which represents the peak dissimilarity between consecutive windows. The signal data between consecutive change points is passed to the final stage of the algorithm, which performs gait segment recognition in a similar manor to [17]. The recognition algorithm, uses a set of rules to determine if a gait signal is present in a given segment. Additionally, the algorithm joins segments based on given set of rules.

3.1 Stage 1: Dissimilarity Measure

The dissimilarity between two time windows of gait data is measured with the Pearson (PE) divergence [11, 16]

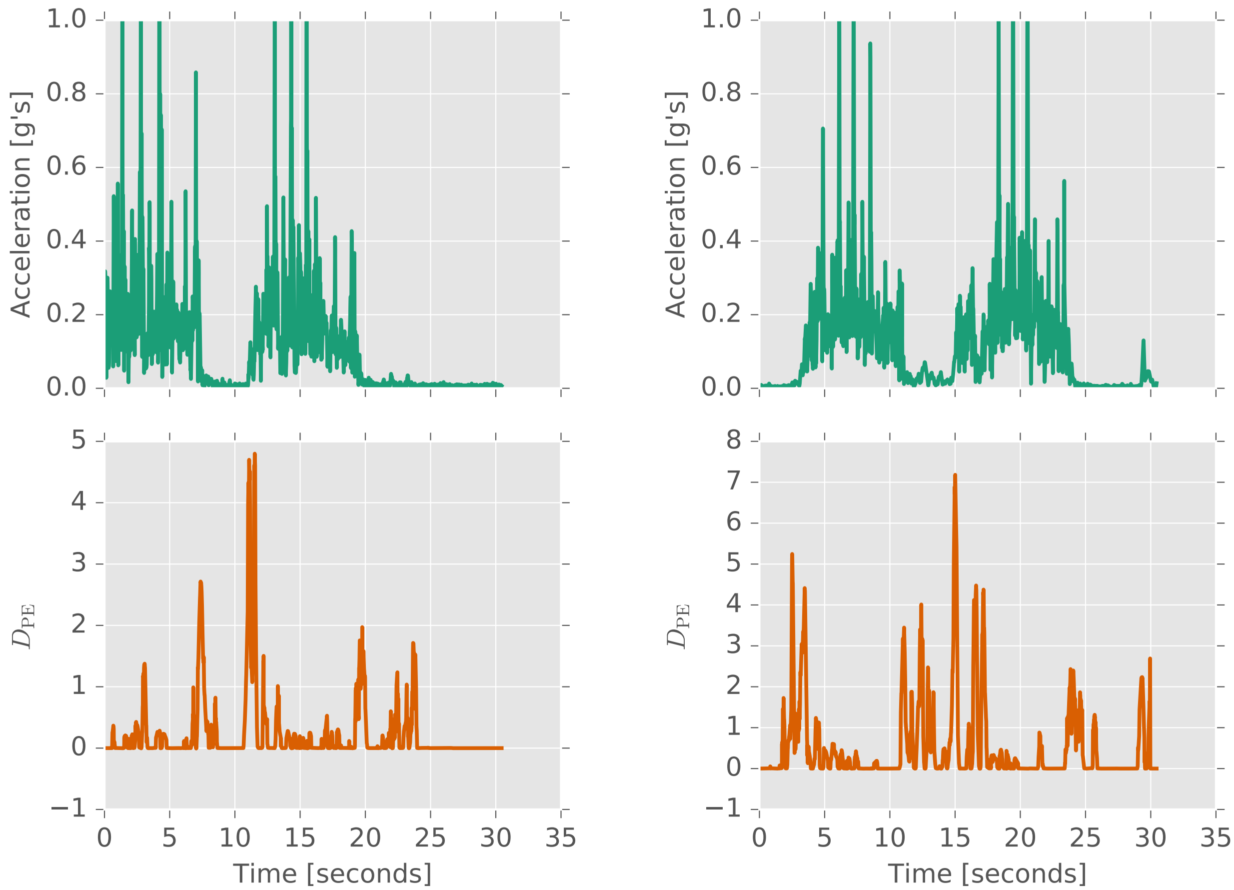
$$D_{\mathrm{PE}}\left(P||Q\right) = \frac{1}{2}\int q\left(\mathbf{X}\right)\left[\frac{p\left(\mathbf{X}\right)}{q\left(\mathbf{X}\right)} - 1\right]^2 d\mathbf{X} \tag{1}$$

where $p\left(\mathbf{X}\right)$ and $q\left(\mathbf{X}\right)$ are the probability density functions (pdfs) of two consecutive data windows. Calculation of the D_{PE} can be challenging since parametric methods require *a priori* knowledge of the underlying pdf and multivariate, non-parametric methods suffer from a curse of dimensionality [18]. Because of the ratio, $p(\mathbf{X})/q(\mathbf{X})$ in (1), DDRE techniques [11, 16], are attractive since estimation of individual pdfs is avoided all together. The authors [11, 16] have provided a MATLAB implementation for DDRE. For this paper the algorithm was ported to the python 2.7 numerical environment.

Figure 3 shows example dissimilarity measures for a high- and low-risk individual. For Figure 3 (a), the algorithm finds a change point at approximately $t = 2.5$ s, and then just after $t = 7.5$ s where the outbound signal end. An additional change point is found at approximately $t = 11$ s where turn ends and the inbound signal begins. Finally a change point is found at approximately $t = 20$ s where the return signal ends. Other irrelevant change points are found within the gait cycle. For Figure 3 (b), the algorithm finds a change point at $t = 2.5$ s where the outbound signal begins, $t = 11$ s where the outbound signal ends, $t = 15$ s where the inbound signal begins, and $t = 24$ s where the inbound signal ends.

3.2 Stage 2: Adaptive Threshold Detection

The most relevant $D_{\mathrm{PE}}\left(P||Q\right)$ values, x are found using an adaptive threshold algorithm based on the CFAR detector [13], as in Figure 2. The concept for the threshold calculation is shown in Figure 4 where x is the value under test, the cross-hatch boxes are guard value, and the gray boxes make up the leading and lagging windows used for computing the threshold. The use of guard values prevents closely spaced divergence values from biasing the threshold calculation since the divergence is calculated over a sliding window. The threshold



(a) (b)

Figure 3: Dissimilarity measure or Pearson divergence D_{PE} for a high risk (a) and general risk faller(b). The top plots are the magnitude of the acceleration measurements, and the bottom plots are the D_{PE}.

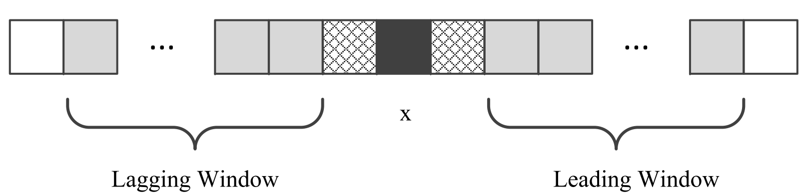
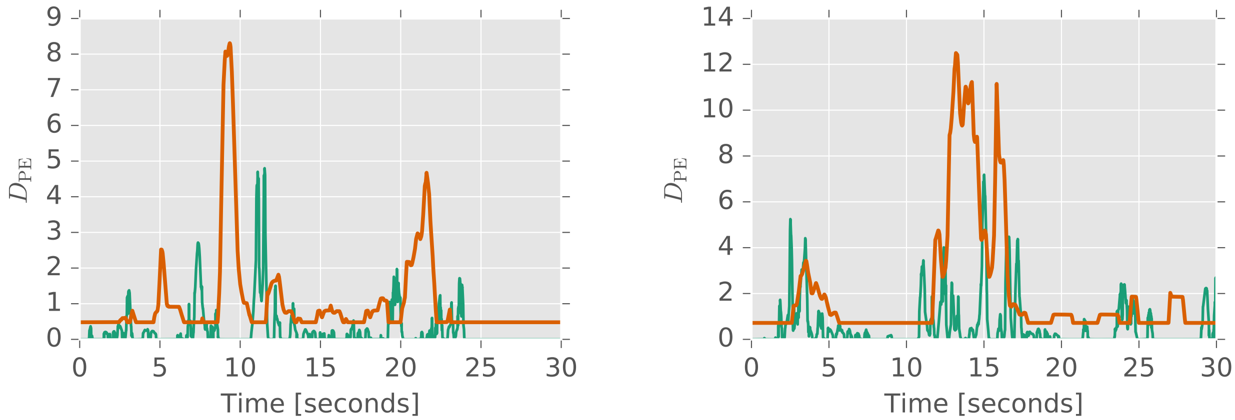


Figure 4: Adaptive threshold concept. x is the test sample, the hatched boxes are the guard samples, and the gray boxes are the sample used for the threshold calculation.



(a)

(b)

Figure 5: D_{PE} values (green curve) and threshold (orange curve) for a high- (a), low-risk individual (b). Candidate change points, $\{t_{\mathrm{CP}}\}$ occur when the green curve is greater than the orange curve.

is calculated as

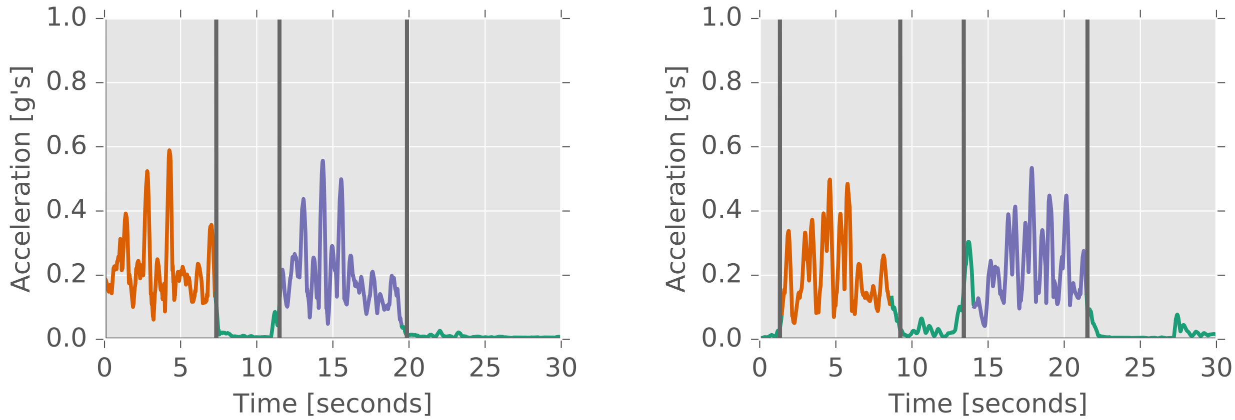
$$T = \frac{\alpha}{N} \sum_{i=0}^{N-1} x_i \tag{2}$$

where N is total number of samples in both the leading and lagging windows and α is a constant which is determined by the desired probability of false alarm [13]. The detection of irrelevant change points is mitigated by setting a minimum threshold level and minimum time between candidate change points. Once a change point is detected, the algorithm searches for the peak divergence. Figure 5 shows an example of the output of the threshold detection algorithm. The green curve shows the D_{PE} values and the orange curve is the detection threshold, T. Candidate change points, $\{t_{\mathrm{CP}}\}$ occur when the green curve is greater than the orange curve. In Figure 5 (a), the algorithm finds key change points occurring at approximately $t = 3$ s, $t = 7$ s, $t = 11$ s, $t = 13$ s, $t = 20$ s, $t = 23$ s, and $t = 24$ s. In Figure 5 (b), the algorithm finds key change points occurring at approximately $t = 2$ s, $t = 2.5$ s, $t = 11$ s, $t = 12$ s, $t = 15$ s, $t = 16$ s, $t = 17$ s, and $t = 25$ s.

3.3 Stage 3: Gait Recognition and Labeling

The proposed gait recognition algorithm utilizes two tests, similar to [17], for gait pattern segmentation and labeling as in Figure 2. These tests check for the number of peaks within a segment, i.e. gait data between candidate change points and the duration of the segments. Valid segments are then passed on to a labeling process.

The first test is to apply a peak detection algorithm to each individual segment. Peaks are retained if they exceed a minimum amplitude and time span between peaks. The next step checks the number of accepted peaks. If the minimum peak threshold is exceeded than the segment is passed on to the duration test, else the segment is discarded. The minimum peak test is designed to remove segments that do not contain a valid gait pattern. A threshold of



(a) (b)

Figure 6: Segmented gait signal for a (a) high-risk and (b) low-risk individuals. The orange portion of the signal is the outbound segment, the purple portion is the return segment, and the green portions are the discarded segments. Manually identified change points are represented by the vertical lines. In this example this close agreement between the algorithm-selected and manually selected change points.

two peaks is used since a change point could have occurred within the gait cycle isolating a series of two steps.

The second test determines if the segment exceeds a minimum duration. If the duration threshold is exceeded then it is passed on to be labeled, else the segment is held and the next segment is processed through the minimum peaks test. If the new segment passes the minimum peaks test then the two segments are merged and passed on for labeling. Otherwise, the new segment is discarded and the previous segment is passed on for labeling.

The segment labeling portion of the gait recognition algorithm checks if a segment is unique or if it is part of a previous segment. If the difference between the time of occurrence for the last peak in the previous segment and the first peak in the current segment does not exceed a given threshold then the segments are assigned the same label. Each new gait segment is assigned a unique label. All discarded segments are assigned a label of zero. Figure 6 shows an example of the automatically-segmented signal where the outbound signal is orange, the return signal is purple, and the discarded portions are green. Manually selected change points are denoted with the vertical lines. In this example this close agreement between the algorithm-selected and manually selected change points.

4 RESULTS

The data set used for this paper consisted of a total of 108 files, which were processed using both the proposed segmentation algorithm and manual segmentation. Each file contained either an outbound segment or an outbound and a return segment. A total of 156 segments were manually identified, where 60 files only had an outbound segment and 48 files had an outbound and a return segment. Manual segmentation was performed in MATLAB

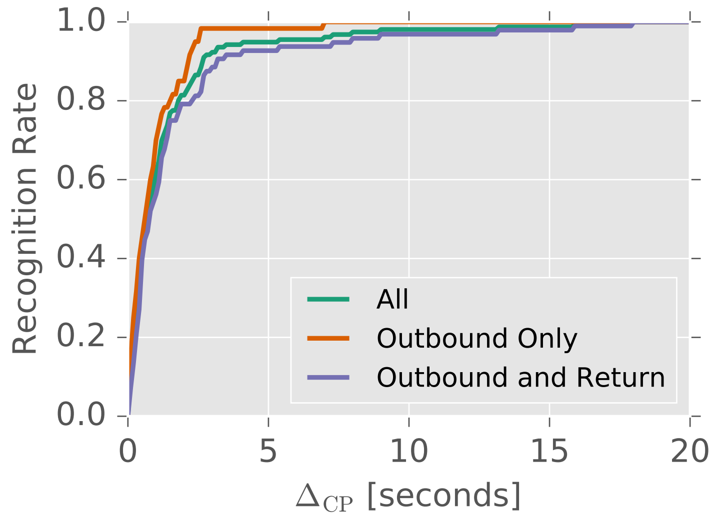


Figure 7: Detection rate versus difference between manually and algorithm identified change points Δ_{cp}. For all gait segments, a recognition rate of 75% is achieved when $\Delta_{\mathrm{CP}} = 1.4$ seconds

uses a custom graphical user interface to prompted the user to select the times of the four change points. These change points were used as ground truth for evaluating the proposed segmentation algorithm, although we recognize that manually identified change points may have some minor error.

The vector magnitude of the acceleration measurements was used as the input to the proposed segmentation algorithm, since the reference planes for the inertial sensors will differ for the left and right smartphones [14]. The DDRE algorithm used the default parameters suggested in [11] and the adaptive threshold algorithm used leading and lagging windows that were each 1.5 s long, and the guard samples spanned 0.75 s.

A segment was identified correctly if the algorithm-selected change points were within $\pm\Delta_{\mathrm{CP}}$ s from the manually-identified change points. Figure 7 shows recognition rates for increasing Δ_{CP}. The green curve shows the recognition rate for all segments, the orange curve shows the recognition rate for files containing only an outbound segment, and the purple curve shows the recognition rates for the files only containing both outbound and return segments.

A recognition rate of 71.8% is achieved when $\Delta_{\mathrm{CP}} = 1.3$ s (average stride time based on the walkway measurements) for all files. For files containing only an outbound signal the recognition rate is 78.3%, and files containing both an outbound and return segment the recognition rate is 67.7%. If $\Delta_{\mathrm{CP}} = 2.6$ s(two strides), the recognition rates are 88.5%, 98.3%, and 82.3% for all gait patterns, outbound only files, and both outbound and return files.

5 CONCLUSION

This paper has proposed an unsupervised segmentation and labeling algorithm for inertial gait measurements. The algorithm utilized a direct density ratio estimation technique to estimate the Pearson divergence of consecutive windows of data. The divergence is then is used to find candidate change points and an adaptive threshold algorithm was used to find the most relevant change points. Segments containing valid gait patterns are then found using a rule based gait recognition algorithm, which allows for the joining and splitting of signal segments. Finally the algorithm was evaluated against manually-identified change points using a data set containing 156 gait patterns. For all gait patterns, the algorithm achieves a 71.8% recognition rate when the algorithm-identified change points are within one average stride of the manually-identified change points and 88.5% when the algorithm-identified change points are within two average strides.

References

[1] J. M. Fabre, R. Ellis, M. Kosma, and R. H. Wood, "Falls risk factors and a compendium of falls risk screening instruments," *J. Geriatric Physical Therapy*, vol. 33, no. 4, pp. 184–197, Oct-Dec 2010.

[2] J. Howcroft, J. Kofman, and E. D. Lemaire, "Review of fall risk assessment in geriatric populations using inertial sensors," *J. Neuroengineering Rehabil.*, vol. 10, no. 91, pp. 1–12, Oct 2013.

[3] J. Verghese, C. Wang, R. B. Lipton, R. Holtzer, and X. Xue, "Quantitative gait dysfunction and risk of cognitive decline and dementia," *J. Neurol., Neurosurg., Psychiatry*, vol. 78, no. 9, pp. 929–935, Sep 2007.

[4] J. Verghese, R. Holtzer, R. B. Lipton, and C. Wang, "Quantitative gait markers and incident fall risk in older adults," *The Journals of Gerontology Series A: Biological Sciences and Medical Sciences*, vol. 64A, no. 8, pp. 896–901, Aug 2009.

[5] S. Lord, B. Galna, J. Verghese, S. Coleman, D. Burn, and L. Rochester, "Independent domains of gait in older adults and associated motor and nonmotor attributes: Validation of a factor analysis approach," *The Journals of Gerontology Series A: Biological Sciences and Medical Sciences*, vol. 68, no. 7, pp. 820–827, Jul 2013.

[6] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *SIGKDD Explor. Newsl.*, vol. 12, no. 2, pp. 74–82, Mar. 2011.

[7] R. Prescott Adams and D. J. C. MacKay, "Bayesian Online Changepoint Detection," *arXiv*, Oct. 2007, stat.ML/0710.3742.

[8] D. Barry and J. A. Hartigan, "Product partition models for change point problems," *Ann. Statist.*, vol. 20, no. 1, pp. 260–279, 03 1992.

[9] J. A. H. Daniel Barry, "A bayesian analysis for change point problems," *Journal of the American Statistical Association*, vol. 88, no. 421, pp. 309–319, 1993.

[10] J. Kohlmorgen and S. Lemm, "A dynamic hmm for on-line segmentation of sequential data," in *Advances in Neural Information Processing Systems 14 (NIPS 2001)*. MIT Press, 2002, pp. 793–800.

[11] S. Liu, M. Yamada, N. Collier, and M. Sugiyama, "Change-point detection in time-series data by relative density-ratio estimation," *Neural Networks*, vol. 43, pp. 72 – 83, 2013.

[12] F. Desobry, M. Davy, and C. Doncarli, "An online kernel change detection algorithm," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2961–2974, Aug 2005.

[13] M. Richards, *Fundamentals of Radar Signal Processing*. McGraw-Hill, 2005.

[14] M. Martinez and P. L. De Leon, "A smartphone-based gait data collection system for the predicion of falls in elderly adults," in *Proc. Int. Telemetering Conf.*, oct 2015.

[15] "Mpu-6000 and mpu-6050 product specification," InvenSense, San Jose, California.

[16] M. Yamada, T. Suzuki, T. Kanamori, H. Hachiya, and M. Sugiyama, "Relative density-ratio estimation for robust distribution comparison," *Neural Computation*, vol. 25, no. 5, pp. 1324–1370, May 2013.

[17] G. Cola, A. Vecchio, and M. Avvenuti, "Improving the performance of fall detection systems through walk recognition," *Journal of Ambient Intelligence and Humanized Computing*, vol. 5, no. 6, pp. 843–855, 2014.

[18] D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley, 2008.