sensors

Article

Characterizing Dynamic Walking Patterns and Detecting Falls with Wearable Sensors Using Gaussian Process Methods

Taehwan Kim [1], Jeongho Park [1], Seongman Heo [1], Keehoon Sung [2] and Jooyoung Park [1,*]

[1] Department of Control and Instrumentation Engineering, Korea University, Sejong-Ro 2511, Sejong-City 30016, Korea; kteaw0110@korea.ac.kr (T.K.); seanpark0107@korea.ac.kr (J.P.); smheo@vadas.co.kr (S.H.)

[2] Research Institute of Korean Air Lines Co., Ltd., Daejeon-City 34054, Korea; skhz89@naver.com

* Correspondence: parkj@korea.ac.kr; Tel.: +82-10-9003-1810

Academic Editors: Edward Sazonov and Subhas Chandra Mukhopadhyay
Received: 26 March 2017; Accepted: 17 May 2017; Published: 20 May 2017

Abstract: By incorporating a growing number of sensors and adopting machine learning technologies, wearable devices have recently become a prominent health care application domain. Among the related research topics in this field, one of the most important issues is detecting falls while walking. Since such falls may lead to serious injuries, automatically and promptly detecting them during daily use of smartphones and/or smart watches is a particular need. In this paper, we investigate the use of Gaussian process (GP) methods for characterizing dynamic walking patterns and detecting falls while walking with built-in wearable sensors in smartphones and/or smartwatches. For the task of characterizing dynamic walking patterns in a low-dimensional latent feature space, we propose a novel approach called auto-encoded Gaussian process dynamical model, in which we combine a GP-based state space modeling method with a nonlinear dimensionality reduction method in a unique manner. The Gaussian process methods are fit for this task because one of the most import strengths of the Gaussian process methods is its capability of handling uncertainty in the model parameters. Also for detecting falls while walking, we propose to recycle the latent samples generated in training the auto-encoded Gaussian process dynamical model for GP-based novelty detection, which can lead to an efficient and seamless solution to the detection task. Experimental results show that the combined use of these GP-based methods can yield promising results for characterizing dynamic walking patterns and detecting falls while walking with the wearable sensors.

Keywords: walking; fall detection; wearable sensors; Gaussian process; dynamic model; dimensionality reduction; novelty detection; latent feature space

1. Introduction

By incorporating a growing set of sensors and adopting machine learning technologies, wearable devices have recently become a prominent health care application domain for a variety of multi-disciplinary research areas. As a useful health care application domain along these lines, remote health monitoring systems (RHMS) (see e.g., [1,2]) have been developed to perform health care tasks such as continuous recording of health data and identifying health status. As the number of smartphones equipped with various multipurpose sensors including acceleration sensors and gyro sensors have rapidly increased, continuously recording and identifying health status from daily smartphone use is now possible, and smartphones are rapidly becoming a key intelligent device that can extract useful heath information during people's daily activities. Many smartphone users now use smart watches, with a variety of modern functions such as sensing acceleration and rotation rate along with wireless data transmission capability. Since wearable sensors are becoming increasingly prevalent with smartphones and/or smart watches, it is possible to detect various health-related human activities in mobile device environments. Human behavior patterns that can be recognized by such means include walking and falling down. Among the relevant health care research topics, one of the most important issues is detecting falls while walking. Since falls may lead to serious injuries, automatically and promptly detecting them is particularly important.

In this paper, we are concerned with the problem of characterizing dynamic walking patterns and detecting falls while walking with wearable sensors by means of machine learning methods. Since many health issues are related to walking and/or falling, many research efforts are currently focused on the problems and questions of human walking patterns and falling. When dealing with these problems, decision making by raw sensor output data only is generally insufficient. For example, simply utilizing thresholds for sensory values from accelerators and gyros to detect falls would easily suffer from detection errors and false alarms. Recently, various machine learning methods such as artificial neural networks (ANNs), support vector machines (SVMs), and k nearest neighbors (kNNs) have been successfully applied to distinguishing activities such as walking and falling (see e.g., [3–6]) based on data from wearable sensors. For more related works on the subject, the reader is referred to papers such as [7–13]. Jeon et al. [7] proposed a method for detecting falls using two accelerometer sensor units. In their proposed method, the falls were detected based on the z-axial acceleration differences between two sensors which are located at subject's chest and abdomen. Ellis et al. [8] addressed the problem of modeling pedestrian trajectory patterns, and proposed the use of Gaussian process regression that can model trajectory data by regressing relative motion against current position. As an application domain, the reference paper [8] considered databases collected from static surveillance cameras, and illustrated the benefit of the proposed approach for long term motion prediction. Ojetola et al. [9] proposed the use of decision trees for detecting four types of falls (forward, backward, right, and left) with wearable sensors, and showed that when applied to experimental data from eight male subjects, the proposed algorithm discriminated between activities of daily living and falls with a precision of 81% and recall of 92%. Choi et al. [10] studied on the use of machine learning algorithms for fall detection and movement classification. In their works, different machine learning algorithms were considered to identify and detect three type of normal activities (standing, walking, and lying down), four types of fall trajectories (forward, backward, left, and right), and near fall situation. Abbate et al. [11] proposed a method for the elderly people's fall detection using smartphone and external sensor, where the falls were detected by threshold. In their method, the system produced an alarm for the acceleration higher than $3g$. Also, they used the fall-like daily activities, which reduced the false alarms. Wang et al. [12] proposed an algorithm which detected five different human walking patterns from data acquired using a triaxial accelerometer. In their works, the sensor's signal was decomposed into frequency scale by discrete Fourier transform, and then the features were classified by multi-layer perceptron neural networks. Sekine et al. [13] distinguished walking on level ground from walking on a stairway using waist acceleration signals. Their method was based on the wavelet coefficients, and the experiment was conducted with elderly people walking corridor and stairs. While machine learning methods directly handling data sets can provide reasonably good performance many cases, understanding intrinsic characteristics of human activity is often more important for attaining enhanced robustness and reliability. In this paper, we examine the problem of understanding and characterizing the intrinsic dynamics of walking patterns and detecting falls while walking with wearable sensors. For solving this problem, we utilize several Gaussian process methods for dynamic modeling, dimensionality reduction, and novelty detection. In particular, for the step of characterizing dynamic walking patterns in a latent feature space, we propose a novel approach called auto-encoded Gaussian process dynamical model (GPDM).

Gaussian processes are a branch of probabilistic kernel method closely related to non-probabilistic kernel methods such as support vector machines [14]. Recently, Gaussian processes have been successfully applied to a variety of data analysis tasks including regression, classification, and non-linear feature extraction (see e.g., [15–17]). One of the main advantages of the GP methods compared to non-probabilistic kernel methods is that, in their solutions, probabilistic information on uncertainty can be additionally provided. Since probabilistic uncertainty information is important in dealing with dynamic modeling, dimensionality reduction, and novelty detection problems, the GP methods are particularly useful for our purposes in this paper.

With this background in mind, we examine the problem of characterizing dynamic walking patterns and detecting falls with wearable sensors based on Gaussian process (GP) methods. In the modeling steps, we propose the combined use of the Gaussian process dynamical model (GPDM) [18–20] with the concept of an auto-encoder (e.g., [21,22]). The proposed combination results in the auto-encoded GPDM, a novel nonlinear dynamic structure for dimensionality reduction. Dimensionality reduction is one of the most important concepts in the field of machine learning. Well-known conventional linear dimension reduction methods include principal component analysis (PCA) and linear discriminant analysis (LDA) [23]. When the underlying structure of the problem is nonlinear and high-dimensional, nonlinear dimension reduction is more useful. Among conventional nonlinear dimension reduction methods are locally linear embedding (LLE) [24], isometric mapping (ISOMAP) [25], and kernel principal component analysis (KPCA) [26]. One of the most promising modern approaches to nonlinear dimensionality reduction is the use of auto-encoders (e.g., [21,22]), which are currently hot topics in various deep learning applications. Our auto-encoded GPDM is an extension of the GPDM [18–20]. In contrast to GPDM, which relies on time-consuming optimization in order to find latent features for given observation data, the GP encoder, which is derived from training the auto-encoded GPDM, can provide an intrinsic representation in low-dimensional latent space for the given observation data without resorting to optimization. For the task of detecting falls while walking, we use the GP-based novelty detection [27–29] for the latent samples resulting from the auto-encoded GPDM. Note that, since the latent samples generated by the training of the GPDM are recycled in our novelty detection step, the combination of these GP-based modules together are seamlessly natural.

The remainder of this paper is organized as follows. In Section 2, after providing preliminary background on Gaussian processes, we present the GP-based solutions for the problem of characterizing dynamic walking patterns and detecting falls while walking. In Section 3, the effectiveness of the GP-based solutions is illustrated by experiments, and in Section 4, the usefulness of the GP-based solutions is discussed. Finally, in Section 5, concluding remarks are provided along with topics for future studies.

2. Methods

2.1. Preliminaries

In this paper, we examine the problem of characterizing dynamic walking patterns and detecting falls while walking with built-in wearable sensors in smartphones and/or smart watches. We deal with the problem by utilizing several Gaussian process (GP) methods. Since the used GP methods are all built on Gaussian process regression (GPR) [15], we begin by introducing the concept of Gaussian process regression (GPR). For more details on Gaussian processes, readers are referred to references such as [15,16].

Gaussian processes are an example of non-parametric probabilistic models and have been successfully applied to many practical problems [15]. The Gaussian process, $\{f(x)\}$, is defined as an indexed family of random variables with index $x \in R^d$ such that for any finite indices, $x_1, \cdots, x_N$, the corresponding random variables, $f(x_1), \cdots, f(x_N)$ are jointly Gaussian. In a certain stage of our application, the index space, R^d, where the index x_i belong, will play the important role of the latent feature space. A Gaussian process $\{f(x)\}$ is characterized by its mean function, $m(x) = E[f(x)]$, and the covariance function (which is also known as the kernel function), $\kappa(x, x') = E[(f(x) - m(x))(f(x') - m(x'))]$. The mean function $m(x)$ is generally assumed to be the zero function, and the covariance function $\kappa(x, x')$ is chosen as a class of functions parametrized by a hyper-parameter vector. The Gaussian process with the mean function $m(x)$ and the kernel function $\kappa(x, x')$ is denoted by $f(x) \sim GP(m(x), \kappa(x, x'))$. The concept of the Gaussian process can also be explained using weight space. In the weight space framework, $f(x)$ is described by a linear combination of basis functions, $\sum_{j=1}^{J} w_j \psi_j(z) = \psi(x)^T w$, where $\psi(x)$ is the feature vector, and the prior distribution of the random coefficient vector w is the Gaussian with a zero mean and a certain covariance matrix Σ_w (i.e., $w \sim N(0, \Sigma_w)$). Note that, in the weight space framework, the mean function of $f(x)$ satisfies $E[f(x)] = E[\psi(x)^T w] = \psi(x)^T E[w] = 0$, which shows that the mean zero assumption of the Gaussian process is natural. Given the training data set $D = \{(x_n, y_n)\}_{n=1}^{N}$, we define the design matrix $Z = [x_1, \cdots, x_N]^T$ as the matrix of the input points of the training data, and $T = [y_1, \cdots, y_N]^T$ as the vector of the corresponding target values. Gaussian process regression often views the output y as generated by a zero mean Gaussian process $f(x)$ along with an additive zero mean white Gaussian noise (i.e., $y = f(x) + \epsilon$, where $f(x) \sim GP(0, \kappa(x, x'))$ and $\epsilon \sim N(0, \sigma_n^2)$). Then, the joint distribution of the random vector $F = [f(x_1), \cdots, f(x_N)]^T$, given Z, can be written as

$$p(F|Z) = N(F|0, K(Z, Z)), \tag{1}$$

where $K(Z, Z)$ is an $N \times N$ matrix, whose (i, j)-th element is $\kappa(x_i, x_j)$. For notational convenience, we usually use K instead of $K(Z, Z)$. Note that, by combining $p(F|Z)$ and $p(T|F)$ together, one can write the resultant marginal likelihood as $p(T|Z) = N(T|0, K + \sigma_n^2 I)$. Hence, the log marginal likelihood for the whole training data D can be written as follows:

$$\log p(T|Z) = -\frac{1}{2} T^T (K + \sigma_n^2 I)^{-1} T - \frac{1}{2} \log |K + \sigma_n^2 I| - \frac{N}{2} \log(2\pi). \tag{2}$$

Finding the optimal hyper-parameters can be achieved by maximizing the above log marginal likelihood function with respect to the hyper-parameters. In addition, the predictive distribution of the output y_* for the test input point x_* can be obtained by applying the conditional density formula for the multi-variate Gaussian distributions [15], i.e.,

$$p(y_*|x_*, D) = N(y_*|k_*^T (K + \sigma_n^2 I)^{-1} T, k_{**} - k_*^T (K + \sigma_n^2 I)^{-1} k_* + \sigma_n^2), \tag{3}$$

where $k_* = [\kappa(x_1, x_*), \cdots, \kappa(x_n, x_*)]^T$, and $k_{**} = \kappa(x_*, x_*)$. With $\alpha = [\alpha_1, \cdots, \alpha_N]^T = (K + \sigma_n^2 I)^{-1} T$, the point estimate of the target value for the given test input point x_* can be written as

$$\hat{y}_* = k_*^T (K + \sigma_n^2 I)^{-1} T = \sum_{i=1}^{N} \alpha_i \kappa(x_i, x_*). \tag{4}$$

Finally, note that Equation (4) is the canonical form of the decision function used in the kernel methods [14].

Table 1. Each unit's data set.

Notation	Meaning
A_x	Acceleration along the x-direction
A_y	Acceleration along the y-direction
A_z	Acceleration along the z-direction
A_T	Total magnitude of acceleration, i.e., $\sqrt{A_x^2 + A_y^2 + A_z^2}$
ω_x	Angular velocity around the x-direction
ω_y	Angular velocity around the y-direction
ω_z	Angular velocity around the z-direction
ω_T	Total magnitude of angular velocity, i.e., $\sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}$

2.2. GP-Based Solutions for Characterizing Dynamic Walking Patterns

In this subsection, we first examine the problem of characterizing dynamic walking patterns with wearable sensors based on dynamic modeling and dimensionality reduction. There has been a wide variety of efforts for understanding dynamic walking (e.g., [30,31]). The two main approaches in these efforts are observation-based strategies and model-based strategies. In this paper, we propose a mixed strategy that utilizes the data obtained from wearable sensors together with the GP-based dynamic model for the problem. Figure 1 is the schematic diagram of our strategy for the problem, and the figure shows the configuration of the wearable sensor units in the experiments for verifying the strategy. As shown in Figure 1, we use two sets of wearable sensor units, one of which is carried in the right pocket while the remaining set is attached to the left wrist with straps. Each unit is comprised of three tri-axial devices measuring accelerations and angular velocities along three perpendicular axes. Also for robustness enhancement, we additionally consider total magnitudes [32,33] for acceleration and angular velocity. Note that these total magnitudes can provide additional measures of the degree of movement intensity. An explanation of each unit's data set is given in Table 1. Note that the dimension of the observation space for the configuration is sixteen, i.e., $y(t) \in R^{16}$. The problem of identifying dynamic systems often relies on the auto-regressive state space model [34–36] described by

$$y_{t+1} = h(y_t, \cdots, y_{t-L_y}, u_{t-1}, \cdots, u_{t-L_u}) + \epsilon_t, \tag{5}$$

where $y_t \in R^D$ is the output vector, $u_t \in R^d$ is the input vector, ϵ_t is the noise, and h is a nonlinear function with some universal approximation capability. However, regarding the problem of characterizing dynamic walking patterns with wearable sensors, modeling the dynamic directly with the observation vector via above auto-regressive model would be impractical because the dimension of the observed output vector, D, is too large. A reasonable framework for modeling the dynamics for the noise-prone high-dimensional data is to use the framework consisting of the GP state equation for low-dimensional latent space [18–20] and the GP output equation. This framework is often called the Gaussian process dynamical model or GPDM [18–20], and has the following state and output equations:

$$x_{t+1} = f_X(x_t), \tag{6}$$

$$y_t = g_Y(x_t), \tag{7}$$

where $f_X(x)$ and $g_Y(x)$ are Gaussian processes with zero means and kernel functions $\kappa_X(x, x')$ and $\kappa_Y(x, x')$, respectively (i.e., $f(x) \sim GP(0, \kappa_X(x, x'))$, $g(x) \sim GP(0, \kappa_Y(x, x'))$). Originally, GPDM [18] was applied to the problem of 3D people tracking based on human motion capture data [18]. In this paper, we use an extended version of the GPDM for the purpose of characterizing dynamic walking patterns with wearable sensor data. For the kernel functions of the GP dynamical model, we use the following:

$$\kappa_X(x, x') = \alpha_1 \exp(-\frac{\alpha_2}{2}\|x - x'\|^2) + \alpha_3 + \alpha_4 x^T x' + \frac{\delta_{x,x'}}{\alpha_5}, \tag{8}$$

$$\kappa_Y(x, x') = \beta_1 \exp(-\frac{\beta_2}{2}\|x - x'\|^2) + \frac{\delta_{x,x'}}{\beta_3}. \tag{9}$$

For the kernel functions of the decoder and transition operator, we used the similar ones with the GPDM paper [18] (i.e., we used the squared exponential (SE), constant (C), linear (LIN), and the white noise (WN) kernels for transition operator, and the squared exponential (SE) and the white noise (WN) kernels for decoder). Also for the encoder, we used the squared exponential (SE) and the white noise (WN) kernels. Note that including the linear term (LIN) is necessary when the dynamics under consideration contains a linear dynamical mode such as periodic oscillation, and the SE kernel are the most important when dealing with smooth interpolation. In addition, note that the constant term (C) is for correlating all values to some degree, and the white noise kernel (WN) is for permitting variation between variables. All of the kernel functions of Equations (8) and (9) are widely used kinds in the studies of kernel methods, and they all satisfy the Mercer law (see, e.g., [14]).

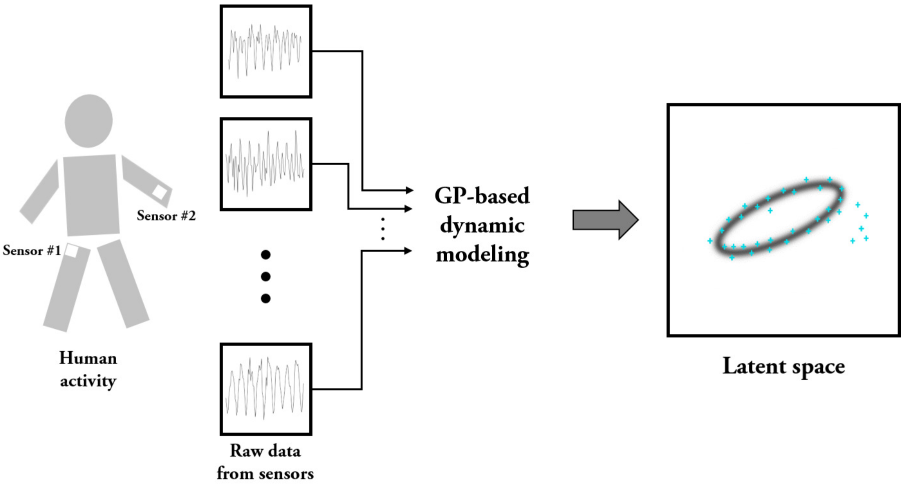


Figure 1. Schematic diagram for representing dynamic walking patterns in a latent space.

Given the matrix of observed data $Y = [y_1, \cdots, y_N]^T \in R^{N \times M}$ and the matrix of the corresponding latent positions $X = [x_1, \cdots, x_N]^T \in R^{N \times m}$, one can obtain the following likelihood and density by defining the kernel matrices K_Y and K_X with $(K_Y)_{ij} = \kappa_Y(x_i, x_j)$ and $(K_X)_{ij} = \kappa_X(x_i, x_j)$, respectively:

$$p(Y|X, \overline{\beta}, W) = \frac{|W|^N}{\sqrt{(2\pi)^{ND}|K_Y|^D}} \exp(-\frac{1}{2}\mathrm{tr}(K_Y^{-1} Y W^2 Y^T)), \tag{10}$$

$$p(X|\overline{\alpha}) = \frac{p(x_1)}{\sqrt{(2\pi)^{(N-1)d}|K_X|^d}} \exp(-\frac{1}{2}\mathrm{tr}(K_X^{-1} X_{out} X_{out}^T)), \tag{11}$$

where $X_{out} = [x_2, \cdots, x_N]^T$, $X_{in} = [x_1, \cdots, x_{N-1}]$. Note that in Equations (10) and (11), $\overline{\alpha}$ and $\overline{\beta}$ are the sets of kernel hyper-parameters for κ_X and κ_Y, respectively, and $W = \mathrm{diag}(w_1, \cdots, w_D)$ is the diagonal matrix of scale parameters. Note that with appropriate prior distributions over $\overline{\alpha}$, $\overline{\beta}$, and W, the GPDM posterior becomes

$$p(X, \overline{\alpha}, \overline{\beta}, W|Y) \propto p(Y|X, \overline{\beta}, W)p(X|\overline{\alpha})p(\overline{\alpha})p(\overline{\beta})p(W). \tag{12}$$

In the original GPDM method, the low-dimensional latent trajectories are obtained through the optimization step, which minimizes the negative log posterior of Equation (12) with respect to the latent variables. However, solving the optimization problem may be computationally expensive and limit its usefulness. In order to overcome this limitation, we propose a novel extension of the GPDM called the auto-encoded GPDM, which is essentially a combination of the GPDM with a variational auto-encoder [22]. The proposed auto-encoded GPDM consists of the GPDM decoder and the GP encoder, where the GPDM decoder is essentially the same as the original GPDM, and the aim of the newly added GP encoder is to discover the intrinsic latent representation, given the observation. More precisely, given the observation data pair $(Y_{1:N-1}, Y_{2:N})$, the GPDM decoder provides the generative model for state transition and output generation (i.e., $p_{trans}(X_{1:N}, \alpha_{trans})$ and $p_{dec}(Y_{1:N}, \beta_{dec}|X_{1:N})$), while the encoding distribution $q_{enc}(X_{1:N}, \gamma_{enc}|Y_{1:N})$ from the GP encoder approximates the true posterior distribution $p(X_{1:N}|Y_{1:N})$. For the GP encoder, we use the so-called automatic relevance determination (ARD) kernel [23] along with the inducing point method [37]. A schematic diagram for the auto-encoded GPDM is given in Figure 2. Note that the hyper-parameters of the GPDM decoder and the GP encoder are denoted by $\{\alpha_{trans}, \beta_{dec}\}$ and γ_{enc}, respectively.

Utilizing the strategy of variational auto-encoders (e.g., [22,38]), we can obtain the following variational objective function, $\mathcal{L}(Y_{1:N}, \alpha_{trans}, \beta_{dec}, \gamma_{enc})$, based on Jensen's inequality [39] as follows:

$$\begin{aligned}
& \log(p(Y_{1:N}, \alpha_{trans}, \beta_{dec})) && (13)\\
= \; & \log(\textstyle\int p_{trans}(X_{1:N}, \alpha_{trans}) p_{dec}(Y_{1:N}, \beta_{dec}|X_{1:N}) dX_{1:N}) && (14)\\
= \; & \log(\textstyle\int q_{enc}(X_{1:N}, \gamma_{enc}|Y_{1:N}) \frac{p_{trans}(X_{1:N}, \alpha_{trans}) p_{dec}(Y_{1:N}, \beta_{dec}|X_{1:N})}{q_{enc}(X_{1:N}, \gamma_{enc}|Y_{1:N})} dX_{1:N}) && (15)\\
\geq \; & \textstyle\int q_{enc}(X_{1:N}, \gamma_{enc}|Y_{1:N}) \log(\frac{p_{trans}(X_{1:N}, \alpha_{trans}) p_{dec}(Y_{1:N}, \beta_{dec}|X_{1:N})}{q_{enc}(X_{1:N}, \gamma_{enc}|Y_{1:N})}) dX_{1:N} && (16)\\
= \; & E[\log p_{dec}(Y_{1:N}, \beta_{dec}|X_{1:N})] + \textstyle\int q_{enc}(X_{1:N}, \gamma_{enc}|Y_{1:N}) \log(\frac{p_{trans}(X_{1:N}, \alpha_{trans})}{q_{enc}(X_{1:N}, \gamma_{enc}|Y_{1:N})}) dX_{1:N} && (17)\\
\triangleq \; & \mathcal{L}(Y_{1:N}, \alpha_{trans}, \beta_{dec}, \gamma_{enc}), && (18)
\end{aligned}$$

where p_{trans}, p_{dec}, and p_{enc} stand for transition distribution, decoder distribution, and encoder distribution, respectively; α_{trans}, β_{dec}, and γ_{enc} are the corresponding parameters for transition distribution, decoder distribution, and encoder distribution, respectively. Here, $X_{1:N}$ are called the latent trajectory (i.e., the state trajectory in the latent space) because they are not measured directly, but can be estimated via the proposed framework of the auto-encoded GPDM. Note that training the proposed auto-encoded GPDM for a single trajectory involves minimizing the reconstruction error for $Y_{1:N-1}$ and $Y_{2:N}$. Here, we denote $Y_{1:N-1}$ and $Y_{2:N}$ as Y and $\tilde{Y}$ in Figure 3 for convenience of notation. In addition, note that training based on $\mathcal{L}(Y_{1:N}, \alpha_{trans}, \beta_{dec}, \gamma_{enc})$ can be handled by achieving the following subgoals (see, e.g., [38]):

$$Y_{recon} \approx Y, \; \tilde{Y}_{recon} \approx \tilde{Y}, \; \tilde{X}_{trans} \approx \tilde{X}, \text{ and } q_{enc}(X) \approx N(0, K), \tag{19}$$

where $Y \triangleq Y_{1:N}$, $\tilde{Y} \triangleq Y_{2:N}$, $X \triangleq X_{1:N-1}$, $\tilde{X} \triangleq X_{2:N}$, and $\tilde{X}_{trans}$ is the output of the transition operation trans : $X_t \mapsto X_{t+1}$ via the state equation used in the auto-encoded GPDM framework. The first expectation on the right side of (17) is called the reconstruction loss term, which is approximated by sampling in the process of training. We call these samples in the latent space the latent samples, and utilize them efficiently in the next step for novelty detection.

Given any observation data, the auto-encoded GPDM is capable of providing an approximate probability density for their corresponding latent objects. Hence, in principle, one can obtain the support of the latent objects by thresholding the resultant probability density function. However, since what matters in the task of detecting abnormal objects is to gain the support of the latent objects, finding their density is stronger than necessary. In this paper, we rely on a practical alternative solution to the task. Since the training phase of the auto-encoded GPDM utilizes the latent samples generated for evaluating the reconstruction loss, we already have latent samples as a by-product of the GPDM training. Based on the latent samples for the normal class, we are able to decide whether test objects belong to normal class. This is the so-called novelty detection problem, which is often called the one-class classification problem or outlier detection problem [40,41]. Recently, GP-based novelty detection methods [27–29] have become available. For the task of detecting falls while walking, we utilize the predictive mean score of the GP-based novelty detection method [27] for the latent samples obtained in the training of the auto-encoded GPDM. Note that, since the latent samples generated for the training of the auto-encoded GPDM are recycled in the novelty detection step, the combination of these GP-based modules is seamlessly natural. Also, note that our method is capable of pre-impact fall detection in the sense that it can detect the falls before the body hits the ground. The schematic diagram for our combination of auto-encoded GPDM and GP based novelty detection for detecting falls is shown in Figure 3. In addition, illustrated falls are shown in Figure 4.

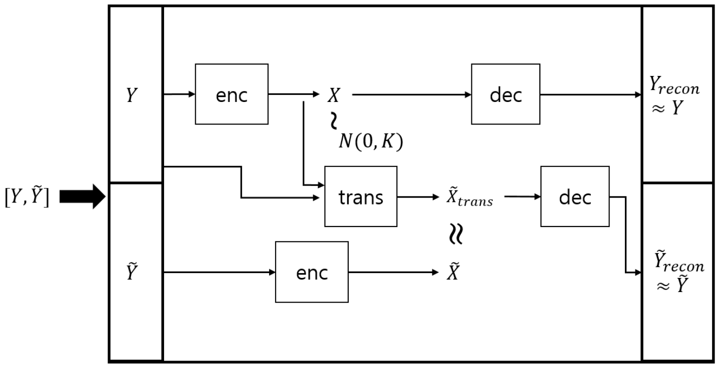


Figure 2. Schematic diagram of auto-encoded Gaussian process dynamical model.

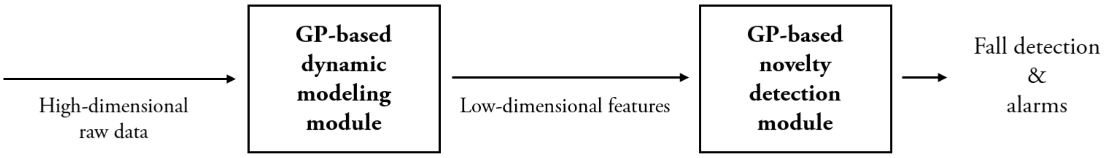

High-dimensional raw data
GP-based dynamic modeling module
Low-dimensional features
GP-based novelty detection module
Fall detection & alarms

Figure 3. Seamless combination of GP-based walking representation and fall detection.

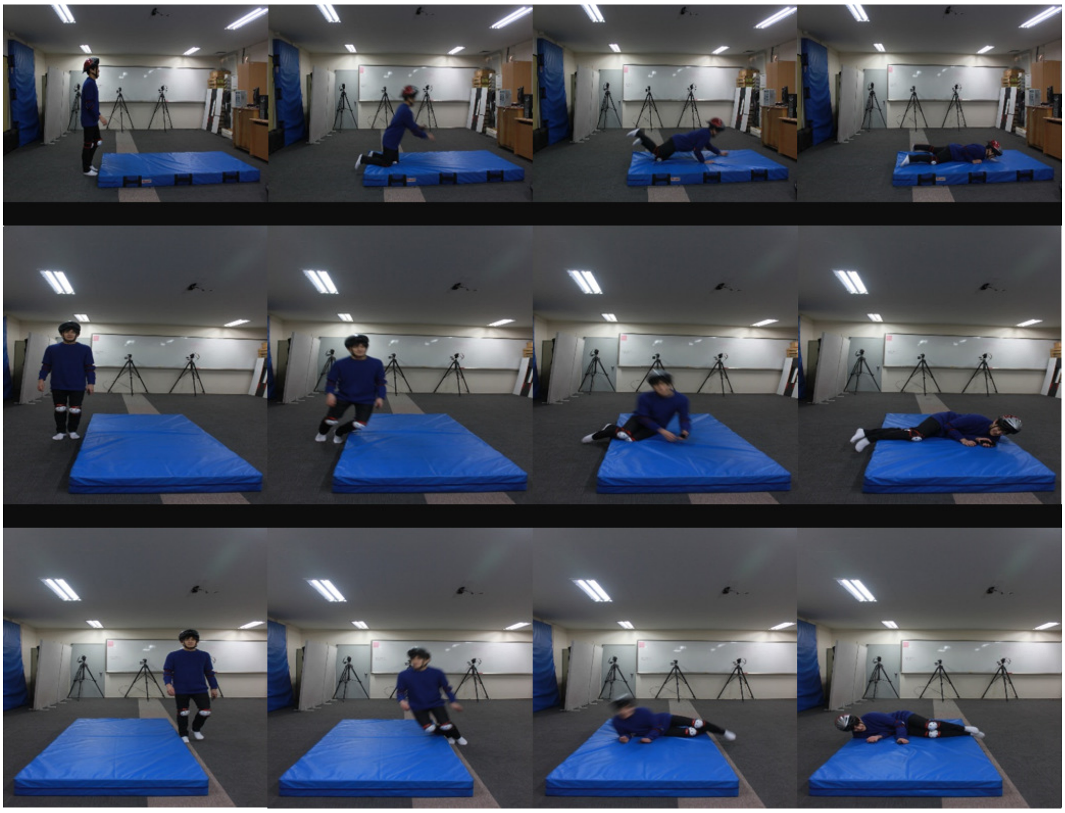


Figure 4. Illustrative falls.

3. Experimental Results

In this section, we first describe the experimental environment and data information to illustrate the proposed GP-based solutions for characterizing dynamic walking patterns and detecting falls. Following the procedure of Figure 5, we performed the experiments at the R&D Center, Korea

University, with its available WiFi networks. The experiment was conducted for the five male volunteers who are 27 years old, 75 kg, 188 cm; 28 years old, 59 kg, 166 cm; 25 years old, 60 kg, 176 cm; 27 years old, 63 kg, 169 cm; 24 years old, 112 kg, 168 cm. During the entire experimental procedure, we used two smartphones: the iPhone 7 (by Apple Inc., Cupertino, CA, USA), one laptop computer: a MacBook Pro (by Apple Inc.), and one application: Matlab (mobile and PC version). The walking data were transmitted from the smartphones to the laptop computer via the WiFi networks. In our experiments, we set the sampling rate, which is the rate of transmitting data from smartphone to PC at 10 Hz. As shown in Figure 1, each test subject performed experiments with the mobile phones in the right pants pocket and the left hand. The smartphone on the left wrist was positioned facing the subject's body, and the other one in the right pocket was positioned facing front. Based on these settings, the test subject walked the predefined courses and fell down at some random moment. Through this process, we obtained the data needed for simulating the proposed method.

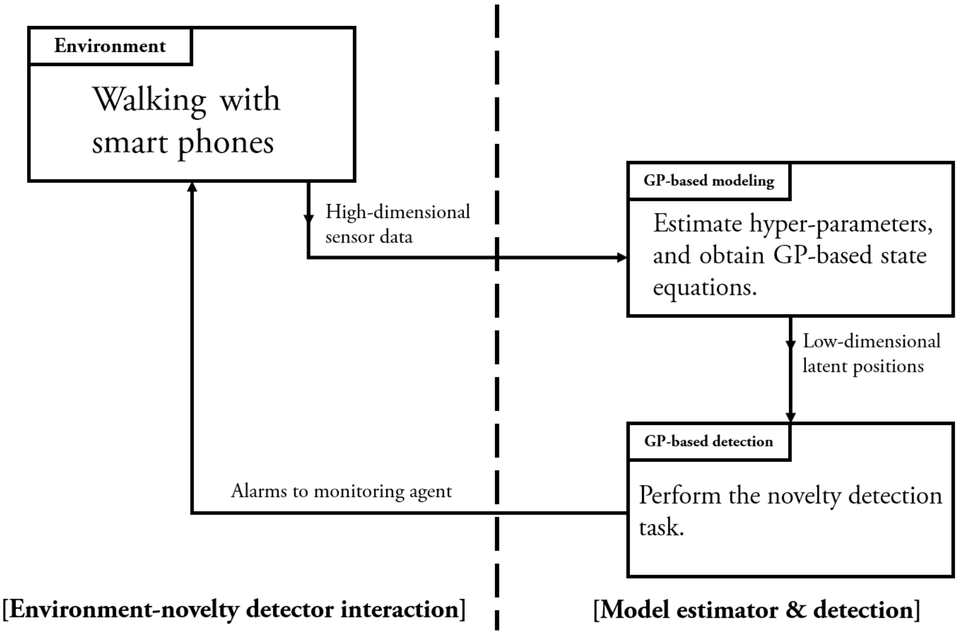

Environment
Walking with smart phones
High-dimensional sensor data
GP-based modeling
Estimate hyper-parameters, and obtain GP-based state equations.
Low-dimensional latent positions
GP-based detection
Perform the novelty detection task.
Alarms to monitoring agent
[Environment-novelty detector interaction]
[Model estimator & detection]

Figure 5. Experimental procedure for GP-based modeling and detection.

In the experiment, we used the acceleration sensors and angular velocity sensors which are built into the iPhone 7. Each smartphone contains two sensors (accelerometer and gyro sensor) that can measure motion data around three orthogonal axes (x, y, z). We thus obtained motion data comprised of twelve features. We additionally considered the feature data with a total magnitude of 3-axes data from each sensor. As a result of acquiring the motion data and pre-processing the data, we obtained a 16-dimensional feature data (Table 1) used for our methodological model. We used the data after applying the commonly used Z-Score normalization technique.

In order to illustrate the GP-based solutions, we considered five sets of training data. In all of these training data, the observation sequence were obtained from five subjects each for 10 sec with the frequency of 10 Hz. In addition, the batch size for training was 50, and initial values of the adjustable parameters were set as: $\beta_{1:3} = [1, 1, 1]$, $\alpha_{1:5} = [1, 1, 1, 1, 1]$, and number of inducing points = 20.

Figure 6 presents the simulation results for the five-fold cross-validation and describes the resultant trajectories in the two-dimensional latent space. The exact meaning of the pictures in the figure is as follows: in the j-th row, which is for the j-th subject, the i-th picture shows the latent trajectories obtained by the proposed auto-encoded GPDM procedure for the i-th experiment, in which the i-th walking data set was used as the test set, and the other four walking data sets were used as the training set for estimating the parameters of the auto-encoded GPDM. In the pictures, the solid line represents some portion of the latent trajectory provided by the GP encoder after training is completed, while the dashed lines indicate some portion of the latent trajectories of the training data sets. Figure 6 shows that the proposed auto-encoded GPDM method worked reasonably well in characterizing dynamic walking patterns in the latent space. From the cross-validation results, one can see obvious similarities between the latent trajectory of the test data and that of the training data. This indicates that the proposed auto-encoded GPDM successfully transformed a high-dimensional time series of noisy observation data signals into a time series of low-dimensional codes, and the training and test data with common characteristics indeed shared similar codes. We believe that this capability of characterizing dynamic walking patterns is of significant practical value, and can be applied to many important real world problems. In addition, we believe that the proposed methods can be deployed into current smartphone and smartwatch systems.

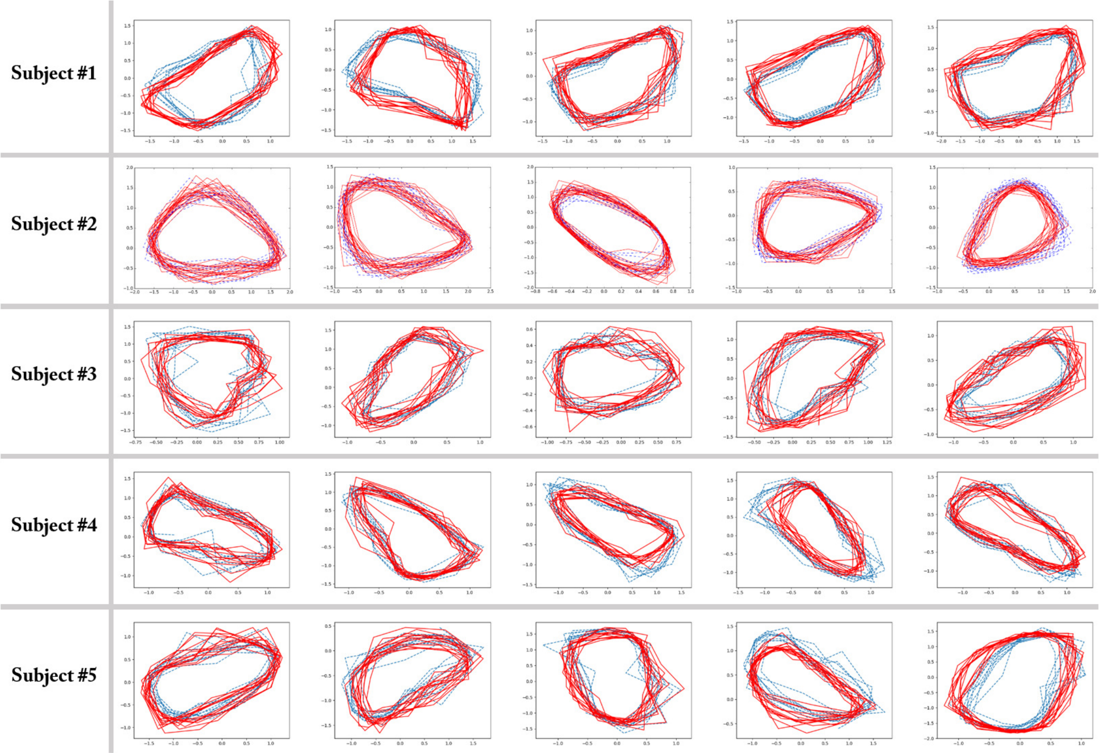
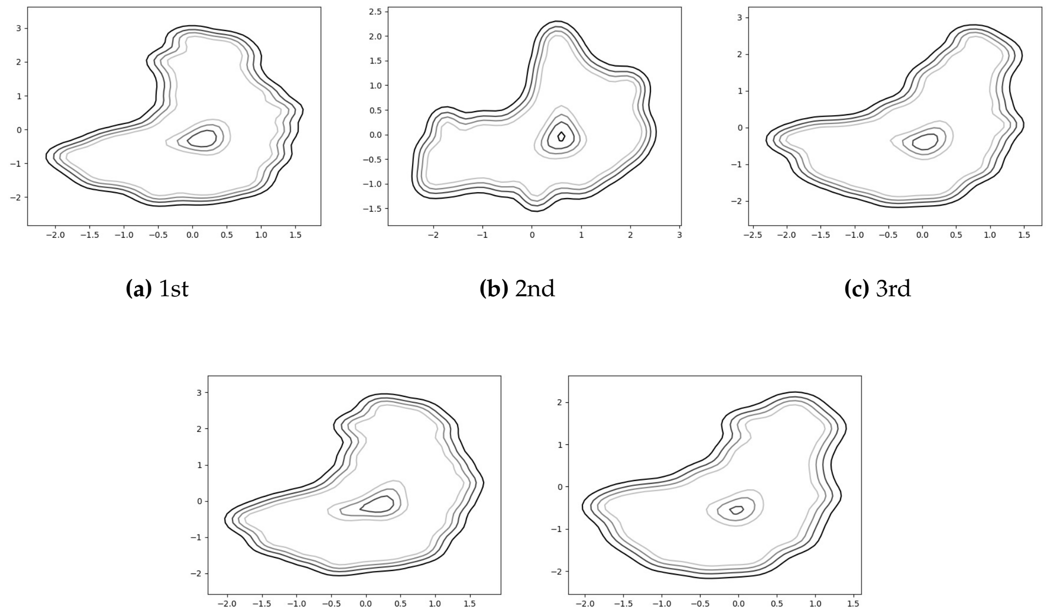


Figure 6. Latent trajectories learned from walking sequences using five-fold cross validation for five subjects (solid lines for test data, and dashed lines for training data). The subjects are the five male volunteers who are 27 years old, 75 kg, 188 cm; 28 years old, 59 kg, 166 cm; 25 years old, 60 kg, 176 cm; 27 years old, 63 kg, 169 cm; 24 years old, 112 kg, 168 cm. Note that there are no units for axes in latent space.

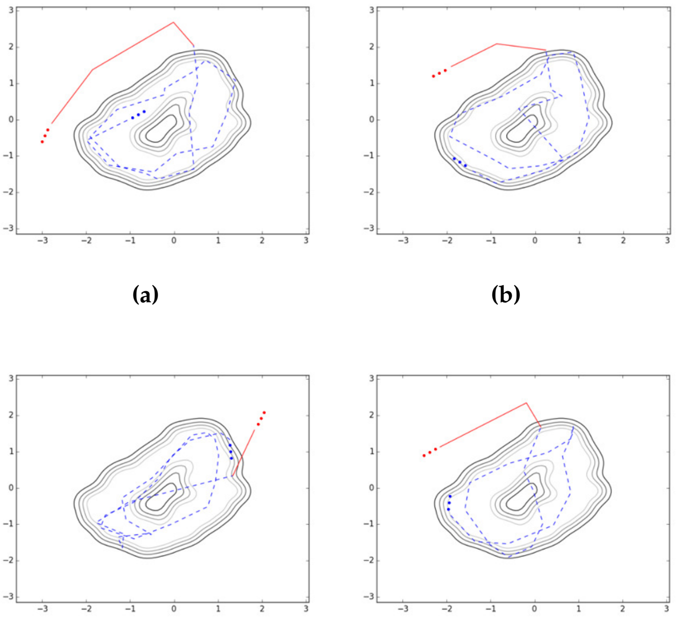
For the task of detecting falls while walking, we utilize the predictive mean score of the GP-based novelty detection method [27–29] for the latent samples obtained in the training of the auto-encoded GPDM. Figure 7 shows how some contours for the normal class look like for an experiment. For the contours, we utilize matplotlib.pyplot.contour [42] to show automatically-chosen-level contours for the predictive mean score. In Figure 8, some typical latent trajectories resulting from falls are shown. As indicated in the figure, one can quickly and reliably provide alarms for falls by noting the abnormal trajectory deviation from the normal region. In addition, the readers can watch the accompanying video [43], which contains an example of not only auto-encoded GPDM results for walking patterns but also detecting falls. Finally, it is one of our future works to deploy the proposed methods in current smartphone and smartwatch systems.



(a) 1st **(b)** 2nd **(c)** 3rd

(d) 4th **(e)** 5th

Figure 7. GP-based novelty detection results for a subject with the five-fold cross validation.



(a) **(b)**

(c) **(d)**

Figure 8. Detecting falls by GP (solid lines for trajectory after fall, and dashed lines for trajectory before fall): left, right, forward, and backward falls. Considered fall trajectories: (**a**) left, (**b**) right, (**c**) forward, (**d**) backward.

4. Discussion

In this paper, we investigated the use of Gaussian process (GP) methods for characterizing dynamic walking patterns and detecting falls while walking with built-in wearable sensors in smartphones. For the task of characterizing dynamic walking patterns in a low-dimensional latent feature space, we presented a novel approach called auto-encoded Gaussian process dynamical model, in which we combine a GP-based state space modeling method with a nonlinear dimensionality reduction method in a unique manner. Our approach is inspired by the GPDM model for human motion capture data [18]. The GPDM model is a GP-based non-parametric model for high-dimensional dynamical systems, and one of the most important strengths of the GPDM model is its capability of handling uncertainty in the model parameters [18]. Compared to the original GPDM works in [18], our works have the following two differences:

- In [18], the GPDM model was applied to the problem of 3D people tracking based on human motion capture data, whereas in this paper, we used an extended version of the GPDM for the purpose of characterizing dynamic walking patterns with wearable sensor data.
- In the original GPDM method, the low-dimensional latent trajectories were obtained through optimization, which minimizes an objective function related with the negative log posterior. The proposed auto-encoded GPDM is equipped with the GP encoder, which is capable of yielding latent representations for given observations. This capability played important roles in our finding latent trajectories for test data (e.g., solid lines in Figure 6).

The capability of the auto-encoded GPDM can be utilized in a variety of ways. As an example along the line, we performed an experiment for the purpose of illustrating more explicitly about intrinsic characteristic of walking. For this illustration, we collected data of standing and lying down together with walking, and obtained their latent trajectories. Here for standing and lying down, we considered the postures of standing upright at a fixed position and lying down flat on the back, respectively. When collecting data for these postures, slow movements such as leaning a little forward or backward, and turning slowly a little to left or right were also allowed to imitate normal daily activities. Table 2 and Figure 9 report the training procedure and contours of the illustrative experiment conducted for the first subject of our volunteers, respectively. Contours of Figure 9 show good contrast between the considered classes of daily activities, and are capable of yielding an important intrinsic characteristic of walking. Finally, note that the above procedure can be similarly applied to the task of detecting fall while walking. In order to explore this applicability, we already performed the corresponding experiment for the first subject of our volunteers, and reported the resultant latent trajectories in Figure 8 for four cases of falls with different directions (left, right, forward, and backward). Note that the solid lines of Figure 8 showed clearly that the latent trajectories associated with the falling phase deviated significantly from those of the walking phase. It is one of our future works to perform more extensive experiments and report statistical results for the purpose of verifying the applicability.

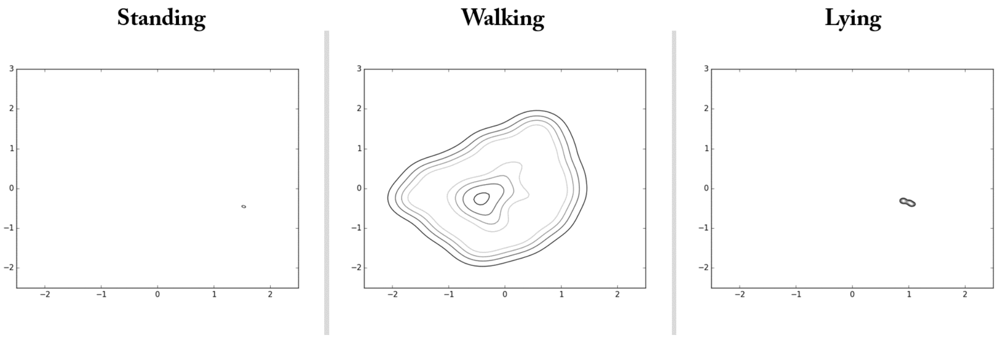


Figure 9. Results for intrinsic characteristics of walking, standing, and lying.

Table 2. Steps for conducting and analyzing the experiment.

1: Obtain five sets of training data for each class of walking, standing, and lying.
2: Obtain five sets of test data for each class of walking, standing, and lying.
3: Train the auto-encoded Gaussian process dynamical model with the training data for walking, and fix the model.
4: Based on the fixed model, compute latent trajectory samples with the training data for each class of walking, standing, and lying.
5: Plot contours of the predictive mean score (see e.g., [27]) based on the latent trajectory samples for each class of walking, standing, and lying.
6: (optional) Perform classification for the test data.

5. Conclusions

In this paper, we examined the problem of characterizing dynamic walking patterns and detecting falls while walking with wearable sensors based on Gaussian process methods for dynamic modeling and novelty detection. For the wearable sensors, we used two units of wearable sensors positioned on the left wrist and in the right pocket, and from each unit, acceleration, the rate of return along three perpendicular axes, and their total magnitudes were used as input signals. For the GP-based dynamic modeling, we proposed the auto-encoded Gaussian process dynamical modeling, which can map noisy high-dimensional input space to the low-dimensional latent feature space, and the resultant latent trajectories efficiently revealed intrinsic characteristics of human dynamic walking patterns. In the training phase of the auto-encoded GPDM, the objective function was defined as the variational approximation to the negative marginal log likelihood function. According to the latent trajectories found by the GP encoder, the low-dimensional latent manifolds associated with dynamic walking patterns were smooth, while the original input signals coming from wearable sensors were often abrupt and noisy. For the task of detecting falls while walking, we used the GP-based novelty detection for the latent samples resulting from the auto-encoded GPDM. Note that, since the latent samples generated by the training for the GPDM are recycled in our novelty detection step, the combination of these GP-based modules are seamlessly natural. In addition, the experiments showed that the proposed GP approach was able to distinguish falls from normal walking. Future work yet to be done includes more extensive experiments and comparative studies, which should reveal the strengths and weaknesses of the proposed approach, and further extension of the approach in several directions. Consideration of other kinds of auto-encoders for nonlinear dimensionality reduction, and the use of a larger number of wearable sensors for the problem are some of the topics to be covered along these lines.

Acknowledgments: This work was supported by the Industrial Convergence Core Technology Development Program (No. 10048474) funded by the Ministry of Trade, Industry & Energy (MOTIE), Korea.

Author Contributions: Jooyoung Park and Keehoon Sung conceived and designed the methodology of the paper; Taewan Kim, Jeongho Park, Seongman Heo, and Keehoon Sung wrote the computer program and performed the experiments; and Taewan Kim, Jeongho Park, and Jooyoung Park wrote the paper.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lan, M.; Samy, L.; Alshurafa, N.; Suh, M.K.; Ghasemzadeh, H.; Macabasco-O'Connell, A.; Sarrafzadeh, M. Wanda: An end-to-end remote health monitoring and analytics system for heart failure patients. In Proceedings of the Conference on Wireless Health, San Diego, CA, USA, 23–25 October 2012; pp. 1–8.
2. Ghasemzadeh, H.; Ostadabbas, S.; Guenterberg, E.; Pantelopoulos, A. Wireless medical-embedded systems: A review of signal-processing techniques for classification. *IEEE Sens. J.* **2013**, *13*, 423–437.

3. Özdemir, A.T.; Barshan, B. Detecting falls with wearable sensors using machine learning techniques. *Sensors* **2014**, *14*, 10691–10708.
4. Luštrek, M.; Kaluža, B. Fall detection and activity recognition with machine learning. *Informatica* **2009**, *33*, 197–204.
5. Liu, S.H.; Cheng, W.C. Fall detection with the support vector machine during scripted and continuous unscripted activities. *Sensors* **2012**, *12*, 12301–12316.
6. Dubey, R.; Ni, B.; Moulin, P. A depth camera based fall recognition system for the elderly. In Proceedings of the International Conference Image Analysis and Recognition, Aveiro, Portugal, 25–27 June 2012; pp. 106–113.
7. Jeon, A.; Park, G.; Ro, J.H.; Geon, G.R. Development of the algorithm for detecting falls during daily activity using 2 tri-axial accelerometers. *World Acad. Sci. Eng. Technol. Issue* **2012**, *61*, 19–23.
8. Ellis, D.; Sommerlade, E.; Reid, I. Modelling pedestrian trajectory patterns with gaussian processes. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision Workshops, Kyoto, Japan, 27 September–4 October 2009; pp. 1229–1234.
9. Ojetola, O.; Gaura, E.I.; Brusey, J. Fall detection with wearable sensors–safe (Smart Fall Detection). In Proceedings of the 2011 7th International Conference on Intelligent Environments, Nottingham, UK, 25–28 July 2011; Volume 5, pp. 318–321.
10. Choi, Y.; Ralhan, A.S.; Ko, S. A study on machine learning algorithms for fall detection and movement classification. In Proceedings of the 2011 International Conference on Information Science and Applications, Jeju Island, Jeju Province, Korea, 26–29 April 2011; Volume 5, pp. 1–8.
11. Abbate, S.; Avvenuti, M.; Bonatesta, F.; Cola, G.; Corsini, P.; Vecchio, A. A smartphone-based fall detection system. *Pervasive Mob. Comput.* **2012**, *8*, 883–899.
12. Wang, N.; Ambikairajah, E.; Lovell, N.H.; Celler, B.G. Accelerometry based classification of walking patterns using time-frequency analysis. In Proceedings of the 2007 29th Annual International Conference of the Engineering in Medicine and Biology Society (EMBS 2007), Lyon, France, 23–26 August 2007; Volume 5, pp. 4899–4902.
13. Sekine, M.; Tamura, T.; Fujimoto, T.; Fukui, Y. Classification of walking pattern using acceleration waveform in elderly people. In Proceedings of the 2000 22nd Annual International Conference of the Engineering in Medicine and Biology Society, Navy Pier Convention Center, Chicago, IL, USA, 23–28 July 2000; Volume 2, pp. 1356–1359.
14. Schölkopf, B.; Smola, A.J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*; MIT Press: Cambridge, MA, USA, 2002.
15. Rasmussen, C.E.; Williams, K.I. *Gaussian Processes for Machine Learning*; MIT Press: Cambridge, MA, USA, 2006.
16. Krauth, K.; Bonilla, E.V.; Cutajar, K.; Filippone, M. AutoGP: Exploring the Capabilities and Limitations of Gaussian Process Models. Available online: https://arxiv.org/pdf/1610.05392.pdf (accessed on 6 March 2017).
17. Lawrence, N.D. Gaussian process latent variable models for visualization of high dimensional data. *Adv. Neural Inf. Process. Syst. (Nips)* **2003**, *2*, 329–336.
18. Wang, J.; Fleet, D.J.; Hertzmann, A. Gaussian process dynamical models for human motion. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 283–298.
19. Urtasun, R.; Fleet, D.J.; Hertzmann, A.; Fua, P. Priors for people tracking from small training sets. In Proceedings of the IEEE International Conference on Computer Vision, Beijing hotel, Beijing, China, 15–21 October 2005; Volume 1, pp. 403–410.
20. Urtasun, R.; Fleet, D.J.; Fua, P. 3D people tracking with Gaussian process dynamical models. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, NY, USA, 17–22 June 2006; Volume 1, pp. 238–245.
21. Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; Manzagol, P.A. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.* **2010**, *11*, 3371–3408.
22. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. *The International Conference on Learning Representations (ICLR)* 2014. Available online: https://arxiv.org/pdf/1312.6114v10.pdf (accessed on 1 May 2014).

23. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer-Verlag: New York, NY, USA, 2006.
24. Roweis, S.T.; Saul, L.K. Nonlinear dimensionality reduction by locally linear embedding. *Science* **2000**, *290*, 2323–2326.
25. Tenenbaum, J.B.; De Silva, V.; Langford, J.C. A global geometric framework for nonlinear dimensionality reduction. *Science* **2000**, *290*, 2319–2323.
26. Schölkopf, B.; Smola, A.; Müller, K.R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.* **1998**, *10*, 1299–1319.
27. Kemmler, M.; Rodner, E.; Wacker, E.S.; Kwok, J.T.; Denzler, J. One-class classification with Gaussian processes. *Pattern Recognit.* **2013**, *46*, 3507–3518.
28. Rodner, E.; Wacker, E.S.; Kemmler, M.; Denzler, J. One-class classification for anomaly detection in wire ropes with Gaussian processes in a few lines of code. *Training* **2011**, *1*, 1–5.
29. Bodesheim, P.; Freytag, A.; Rodner, E.; Kemmler, M.; Denzler, J. Kernel null space methods for novelty detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Oregon, Portland, 25–27 June 2013; pp. 3374–3381.
30. McGeer, T. Powered flight, child's play, silly wheels and walking machines. In Proceedings of the 1989 IEEE International Conference on Robotics and Automation, Scottsdale, AZ, USA, 14–19 May 1989; pp. 1592–1597.
31. Collins, S.H. Dynamic Walking Principles Applied to Human Gait. Ph.D. Thesis, University of Michigan, Ann Arbor, MI, USA, 2008.
32. Karantonis, D.M.; Narayanan, M.R.; Mathie, M.; Lovell, N.H.; Celler, B.G. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE Trans. Inf. Technol. Biomed.* **2006**, *10*, 156–167.
33. Bersch, S.D.; Azzi, D.; Khusainov, R.; Achumba, I.E.; Ries, J. Sensor data acquisition and processing parameters for human activity classification. *Sensors* **2014**, *14*, 4239–4270.
34. Kocijan, J.; Girard, A.; Banko, B.; Murray-Smith, R. Dynamic systems identification with Gaussian processes. *Math. Comput. Model. Dyn. Syst.* **2005**, *11*, 411–424.
35. Van Vaerenbergh, S.; Via, J.; Santamaria, I. Nonlinear system identification using a new sliding-window kernel RLS algorithm. *JCM* **2007**, *2*, 1–8.
36. Van Vaerenbergh, S.; Via, J.; Santamaría, I. A sliding-window kernel RLS algorithm and its application to nonlinear channel identification. In Proceedings of the 2006 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006), Toulouse Pierre Baudis Congress Center, Toulouse, France, 14–19 May 2006; Volume 5, pp. 789–792.
37. Titsias, M.K. Variational learning of inducing variables in sparse Gaussian processes. *AISTATS* **2009**, *5*, 567–574.
38. Watter, M.; Springenberg, J.; Boedecker, J.; Riedmiller, M. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in Neural Information Processing Systems*; Palais des Congres de Montreal, Montreal, CANADA, 7–12 December 2015; pp. 2746–2754.
39. Rudin, W. *Real and Complex Analysis*; Tata McGraw-Hill Education: 2 Pennsylvania Plaza New York, NY, USA, 1987.
40. Park, J.; Kang, D.; Kim, J.; Kwok, J.T.; Tsang, I.W. SVDD-based pattern denoising. *Neural Comput.* **2007**, *19*, 1919–1938.
41. Rätsch, G.; Mika, S.; Schölkopf, B.; Müller, K.R. Constructing boosting algorithms from SVMs: An application to one-class classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 1184–1199.
42. Hunter, J.D. Matplotlib: A 2D graphics environment. *Comput. Sci. Eng.* **2007**, *9*, 90–95.
43. Auto-encoded GPDM. Available online: https://www.youtube.com/watch?v=ZQSdYvSHcU8 (accessed on 16 February 2017).