

Article

# Non-Parametric Bayesian Human Motion Recognition Using a Single MEMS Tri-Axial Accelerometer

M. Ejaz Ahmed and Ju Bin Song \*

Department of Electronics and Radio Engineering, Kyung Hee University, Yongin 446-701, Korea; E-Mail: ejaz629@gmail.com

\* Author to whom correspondence should be addressed; E-Mail: jsong@khu.ac.kr;  
Tel.: +82-31-201-2031; Fax: +82-31-205-1775.

Received: 10 August 2012; in revised form: 19 September 2012 / Accepted: 19 September 2012 /  
Published: 27 September 2012

---

**Abstract:** In this paper, we propose a non-parametric clustering method to recognize the number of human motions using features which are obtained from a single microelectromechanical system (MEMS) accelerometer. Since the number of human motions under consideration is not known *a priori* and because of the unsupervised nature of the proposed technique, there is no need to collect training data for the human motions. The infinite Gaussian mixture model (IGMM) and collapsed Gibbs sampler are adopted to cluster the human motions using extracted features. From the experimental results, we show that the unanticipated human motions are detected and recognized with significant accuracy, as compared with the parametric Fuzzy C-Mean (FCM) technique, the unsupervised K-means algorithm, and the non-parametric mean-shift method.

**Keywords:** MEMS application; human motion recognition; non-parametric Bayesian inference; infinite Gaussian mixture model; Gibbs sampler

---

## 1. Introduction

Human motion recognition (HMR) is an important topic currently being researched due to its large number of applications in tracking, personal navigation, health care, personal life log, surveillance, and sports, among other things. Human motion recognition [1–13] systems have been proposed to efficiently detect numerous human motions, and they are widely applicable in the above mentioned domains.

However, the motion recognition system has some limitations that cannot be mitigated. One of these limitations is inherent in the dynamic nature of human motions; that is, the daily motions of a person are not limited but rather vary from a small to a large set of motions. The introduction of new human motions makes detection and recognition challenging. Therefore, there is a need for a HMR system that is adaptable, accurate, and robust to the dynamic nature of the daily motions performed by a human.

In the health care domain, motions of patients are monitored via wearable sensors. This is useful for three reasons: (1) to keep track of the movements performed by a patient during the medical examination period, (2) to reduce the number of patient visits to medical facilities, and (3) automated motions retrieval and management to facilitate the documentation of a patient's history.

Two main technologies being currently used for motion recognition are body-mounted sensors (accelerometers) [1,2] or image-based recognition (cameras) [3,5] for monitoring a subject's motions. Both methodologies have their own pros and cons. The benefit of camera aided motion identification is that multiple subjects can be monitored simultaneously without the need for any other device. However, the drawbacks of such a technology include (1) failure to achieve higher accuracy due to clutter, light fluctuation under different settings, and varied activities in real environments [4]; (2) image-based motion recognition can be require extensive resources and complex processing [3], and (3) it is not cost effective as it requires at least one calibrated camera in each room setting. On the other hand, the pros of body-mounted sensors are that (1) sensed data is independent of environmental conditions and is person-specific; (2) sensors are compact and portable; and (3) processing data does not require a great deal of resources. Alternatively, the downside of body-mounted sensors is service issues in terms of early battery exhaustion. Numerous HMR techniques developed in the past have focused on the acceleration signals in different directions that enable the classification of physical motions [1,2,6]. Motion recognition using multiple sensors instead of relying on a single sensor is also discussed in the literature [6,8,10–13]. However, these approaches are impractical due to the difficulty and inconvenience associated with affixing these sensors to a subject's body and/or clothing. Previous research about the incorporation of multiple sensors for motion detection results in recognition accuracy ranging from 83% to 95% [11]. In [14], the detection accuracy is shown to be less variable for data acquired in the laboratory compared to data collected in a real life environment. The detection accuracy drops from 95% in the laboratory environment to 65% [14] in the real world.

In the literature regarding the features, most studies incorporated fast Fourier transform (FFT) [15], while others used wavelet transform [6,8], support vector machines (SVM), signal-magnitude area (SMA), mean, variance, entropy, and correlations [1,2,5,11]. However, these features are computed using a wide time window, which is not very effective for detecting rapid transitional movements, such as sitting to standing. In [1], short-term transition motions are detected with 95% accuracy.

Methodologies investigated to this point follow heuristic classifiers, Gaussian mixture models (GMM) [16], support vector machines [12], and hidden Markov model (HMM) [17,18]. However, those approaches consider a fixed number of human motions. In other words, most studies follow parametric classification with a known number of motions. In a daily routine, human motions are not fixed—instead they may vary, and new human motion can also occur. Therefore, motion recognition becomes more challenging when the non-parametric nature of human motions is considered. Non-parametric behavior refers to the fact that the number of motions observed may increase as the amount of data increases,

and may not remain fixed or could be unknown *a priori*. Therefore, parametric approaches for motion detection fail to achieve higher accuracy when new or transitional motions are observed, such as state transition motions (sitting to standing) for which the motion recognition accuracy is 78% [1,19].

In this paper, we propose motion-dependent sensor metrics to identify human motion. Sensors produce a set of unique signal metrics for various human motions. A signal metric is a component of the sensor readings, such as acceleration in the  $x$ ,  $y$  and  $z$  axes of a MEMS accelerometer. These motion-dependent unique signals enable the identification of various motions, including standing, walking, running, taking the stairs, riding an elevator, and more. We considered the autocorrelation function of the tilt angle and the variance of acceleration against the  $x$  and  $y$  axes as features for clustering. Our proposed technique differs from the conventional technique in the following ways:

- Selected features are independent of the sensor device, but only dependent of a particular motion.
- It is an observation-based detection system that does not require any protocol or any active coordination among devices.
- Unsupervised clustering is used without any prior knowledge about the number of clusters.

Our contributions in this study are as follows:

- We propose a non-parametric human motion recognition technique that can detect and recognize an unbounded number of motions (clusters). By unbounded, we mean that the number of clusters (activities) is not fixed. Our techniques can automatically cluster those motions without any prior information.
- The accuracy of the motion detection ranges from 97% to 99% with an unknown number of clusters.
- We compute the Kullback–Leibler divergence (KLD) for newly detected motion using already recognized motions. This enables the system to draw inferences regarding the newly detected motions and cluster them.
- The proposed cluster algorithm collects no prior information about the number of motions, and achieves higher accuracy in detecting the number of clusters compared to the conventional method.

The rest of the paper is organized as follows: Section 2 presents current literature describing various motion recognition techniques. Section 3 presents the system model and feature space with an explanation of each feature. Section 4 presents the non-parametric inference and Section 5 explains the non-parametric inference for motion recognition. Section 6 presents the experimental setup, Section 7 demonstrates experimental results, and Section 8 concludes this paper.

## 2. Literature Review

In this section, we review recently developed approaches in the motion recognition domain. Khan *et al.* [19] used autoregressive (AR) modeling of three-dimensional data from the accelerometer. In their work, AR was combined with the signal-magnitude-area and tilt angle obtained from the sensor readings; all of these parameters formed a feature space for the classification of different gaits. These features were utilized to classify a fixed number of human motions, such as standing, walking, running,

and lying down. A significant improvement in the detection rate was achieved (95% to 99%); however, the detection rate decreased as new motions were incorporated [1].

Adil *et al.* [1] proposed a system that addressed motion recognition for more than ten gaits with a significant detection rate. Linear discriminant analysis was incorporated in order to extract features. They proposed a hierarchical recognition method for the classification and also explored transitional state change detection. The broad set of motions was divided into three categories: (1) static (stand, sit); (2) transitional (sit-stand, walk-stand, *etc.*); (3) dynamic (run, walk, *etc.*). The achieved accuracy was 97%; however, a fixed number of motions was considered for classification.

Ling *et al.* [11] used multiple sensors (five) for motion recognition. The mean, energy, frequency-domain entropy, and correlation of the data were computed, serving as a source for motion detection. Decision trees outperformed other techniques, attaining an accuracy of 84%. Accelerometers mounted on a subject's thigh and wrists were used to detect their daily motions in a naturalistic way. The fast Fourier transform (FFT) and decision tree algorithm were used to classify different human gaits. Frequency-domain entropy was calculated as the normalized information entropy of the FFT component magnitudes of the signals. The decision tree algorithm was able to recognize the motion style with a higher number of labels, such as walking slowly and walking briskly. However, the overall detection accuracy was 80%.

Rodrigo *et al.* [7] proposed sensor-based human motion recognition using the hidden Markov model (HMM) with a large feature set. Candidate features were obtained from the feature space, while the HMM classifier was used to classify the data points for different motions. The multiple sensor-based approach was used to gather data features, and then only those features that are relevant were utilized for the target classes. Genetic algorithms were incorporated to explore the feature set then to select and utilize only those features that are most relevant to the target motion. However, the accuracy of predicting human motion decreases with an increase in the number of motions to be predicted. This is due to the uncertainty that arises from a greater number of classes. Therefore, the probability of assigning a data point to its correct class decreases.

Slyper *et al.* [8] demonstrated an animation system for e-textile application using multiple MEMS accelerometers that are embedded in a shirt. The accelerometer readings are continuously matched against accelerations computed from existing motion capture data, and an avatar is animated with the closest match using wavelet transform, which is a parametric approach.

Ravi *et al.* [9] formulated human motion recognition as a parametric classification problem. Authors proposed an approach to classify different human motions and selected only those features that are important for classification. The following features were used to classify motions: correlation, mean, standard deviation, and energy. The correlation was calculated between each pair of axes as the ratio of covariance and the product of standard deviation. Motions were classified with reasonable accuracy using the above mentioned features.

Practically, human motions are dynamic in that the number of motions varies with time. For example, a person at a particular moment is standing with friend, then after some time he or she starts walking. He or she may then start cycling and during cycling he or she may fall. The flow of different motions performed by a person can fluctuate over time and vary by individual, making it very likely that a motion recognition system will encounter unfamiliar motions. In other words, the number of motions shows

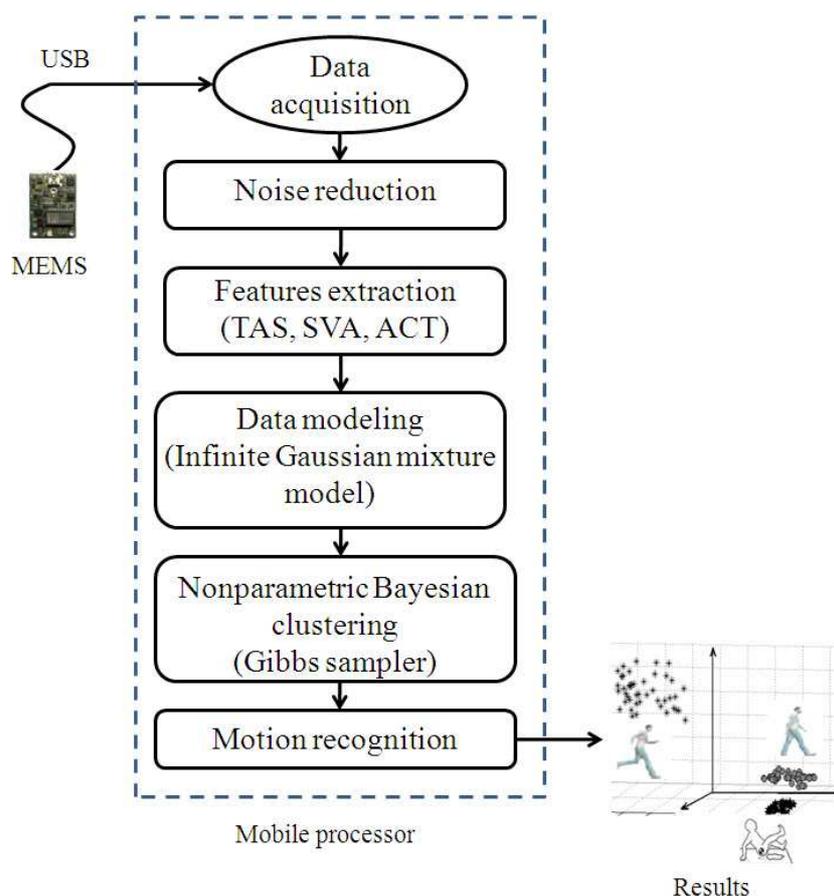
non-parametric behavior. Therefore, a flexible system that can incorporate the time varying behavior of human motions, instead of relying on a fixed number of motions, is needed. In this paper, we propose a non-parametric Bayesian technique that is flexible enough to accommodate the detection and clustering of newly observed motions in order to account for the dynamic and unexpected nature of human motions that vary with time.

System architecture and feature space for the proposed non-parametric human motion recognition will be described in the next section.

### 3. System Architecture and Feature Space for Motion Recognition

The architecture of the proposed system for human motion recognition is illustrated in Figure 1. Each rectangle in the system architecture corresponds to a component responsible for performing a particular task. The system architecture is divided into the following components.

**Figure 1.** System architecture with components and feature space for the proposed motion recognition.

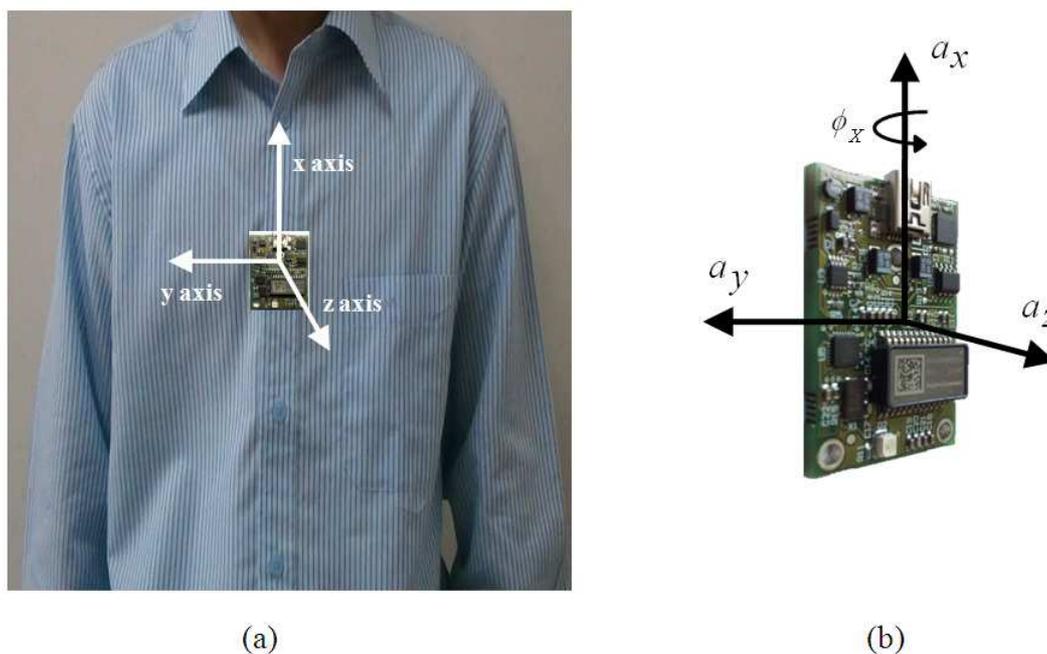


While recording real-time data from a MEMS sensor, the output includes noise. It is important to remove this noise before extracting the features from the sensor output. This component tends to remove signal outliers by filtering the signal. The feature space for motion recognition is described in the following subsection, and infinite Gaussian modeling and non-parametric Bayesian inference modeling

are discussed in detail in the following sections. The Gibbs sampler results in clusters, where each cluster corresponds to a particular human motion. The clusters obtained from the Gibbs sampler are further mapped to a particular recognized human motion.

In this section, we introduce the features used in the proposed motion recognition system. To distinguish among different motions, we need to identify unique features that govern various human motions. These features can be extracted from the readings taken by the MEMS sensor attached to the human body. We used an inertial MEMS sensor system (SD777) with a three-axis accelerometer and a single-axis gyroscope. The inertial MEMS sensor (SD777) is a device that measures its own acceleration as a four-dimensional (4D) vector. This 4D vector includes two measurement ranges for the gyroscope,  $\pm 100$  °/s and  $\pm 300$  °/s, and one measurement range accelerometer (three axes) from  $\pm 1$  g to  $\pm 5$  g. The acceleration can be defined as the rate of change of speed, ( $m/s^2$ ). The acceleration corresponding to each axis can be recorded by a sensor mounted on the subject's chest, as shown in Figure 2.

**Figure 2.** The experimental setup for MEMS sensor: (a) A MEMS sensor tagged on the subject's chest; (b) Three axes of MEMS accelerometers for experimental setup.



For data collection, we placed the accelerometer device on the chest of the subject. We obtained a 4D data set from the accelerometer, consisting of the acceleration on three axes and one gyroscope reading. We propose using the following features for human motion recognition: (1) cumulative sum of a gyroscope's angular speed, tilt angle  $\phi$ ; (2) sum of the variance of the acceleration in the  $x$  and  $y$  axes; and (3) the autocorrelation function of the tilt angle,  $\phi$ . The selected features are unique to each human motion, and all of the features are dependent on human motion. In this study, we assumed that all of the features follow a Gaussian distribution; however the proposed algorithm is not very sensitive to the exact distribution. Each feature is described in detail in the following subsections.

### 3.1. Tilt Angle Sum (TAS)

The tilt angle refers to the tilt of a body in space. The tilt angle can be defined as the angle between the  $x$  axis and gravitational vector  $g$ . Accelerometers are often used to calculate a tilt angle. The angle can be calculated from the acceleration as follows:

$$\phi = \arcsin(a) \quad (1)$$

where  $a = [a_x, a_y, a_z]$  and  $a_x, a_y, a_z$  are the accelerations along the  $x, y,$  and  $z$  axes, respectively. To get an accurate tilt angle, they are often combined with one or more gyros, and a combination of data is used to calculate the angle. Both  $x$  and  $y$  tilt angles can be sensed simultaneously using the output of all three axes, as shown below.

$$\phi = \arctan\left(\frac{a_x}{\sqrt{a_y^2 + a_z^2}}\right) \quad (2)$$

and

$$\rho = \arctan\left(\frac{a_y}{\sqrt{a_x^2 + a_z^2}}\right) \quad (3)$$

The cumulative sum of the tilt angle,  $\phi$ , is an important feature that follows a unique behavior for most human motions. The tilt angle shows distinguishing behavior in a static condition (stay, lift-up, lift-down) compared with dynamic (run, walk, fall) human motions. The TAS can be represented as follows:

$$\mathcal{O}(n) = \sum_{i=1}^W (|\phi|), \quad (4)$$

where  $n$  is a data point from the set of data points  $N$ . A data point is the observation recorded from the attached MEMS accelerometer, while  $W$  is the window size of 512 observations.

### 3.2. The Sum of the Variance of Accelerations (SVA)

Most dynamic gaits (walk, run, fall) produce similar signal amplitude readings for the acceleration in the  $x$  and  $y$  directions, but the variance of each distinguishes them from one another. The SVA plays a significant role in distinguishing different motions. To distinguish between the resting state and motion, all three axis reading changes are represented by this feature. In [20], the signal magnitude area (SMA) is used to distinguish between static and dynamic movements; however, we use different version of SMA. We tend to include the variance of all three axes, which separates dynamic and static human motions in an efficient way. Dynamic motions such as walking, running and falling follow similar behavior, while the variance of the acceleration along different axes differs, which helps in motion detection. We define SVA as the signal magnitude area, that is, the sum of the variance of the areas under the moduli of integrals in the time domain. The SVA can be represented as follows:

$$\Psi(n) = \sum_{i=1}^W \left( \text{var} \left( \int_t a_x(t) dt \right) + \text{var} \left( \int_t a_y(t) dt \right) + \text{var} \left( \int_t a_z(t) dt \right) \right) \quad (5)$$

where  $\Psi(n)$  is the value calculated for a single window.  $a_x(t), a_y(t),$  and  $a_z(t)$  are the body acceleration components in the  $x$ -axis,  $y$ -axis, and  $z$ -axis samples, respectively. This feature is calculated by summing

each sampled value progressively. The value obtained from this feature varies for the dynamic and static movements of the human under consideration.

### 3.3. Autocorrelation Coefficient of Tilt Angle (ACT)

The autocorrelation of a signal measures the similarity between observations as a function of the time separation between them. The goal is to identify a repeating pattern in the time-domain signal. Let  $\phi$  be a repeatable random process, and  $i$  be the time point after the start of the process, then  $\phi_i$  is the realization of the process at time stamp  $i$ . Assume that the process is known to have values for the mean ( $\mu$ ) and variance ( $\sigma$ ) for all times  $i$ . The autocorrelation between time  $a$  and  $b$  can be defined as:

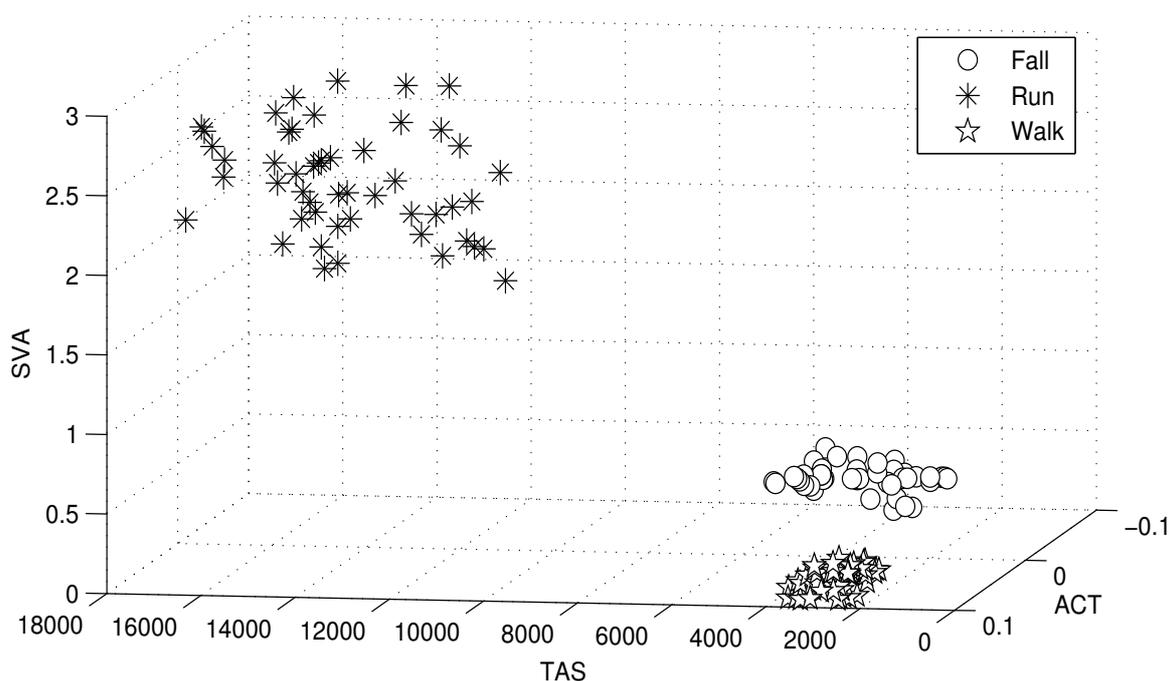
$$R(\phi_a, \phi_b) = \frac{E[(\phi_b - \mu_b)(\phi_a - \mu_a)]}{\sigma_b \sigma_a} \quad (6)$$

where  $E$  is the expected value, and the expressions for the autocorrelation are not well defined because the variance may be zero or infinity. However, in our data set, the variance is bounded by a well-defined limit. The above mentioned features play an important role in distinguishing among various static and dynamic motions; however, there are some motions that show low similarity with each other in terms of the time-domain signal for the tilt angle,  $\phi$  (e.g., walk, run), but some motions have a smooth curve when the tilt angle is plotted as a function of time (picking up, putting down). Therefore, the ACT is an important metric in capturing this distinct and motion-specific variation. The ACT is represented as follows:

$$\Delta(n) = \sum_{i=2}^W \frac{R(\phi_i, \phi_{i-1})}{W-1} \quad (7)$$

where  $i$  is the realization of the  $\phi$  from the time-domain signal.

**Figure 3.** 3D feature space with three human motions recognized.



The clusters are modeled as the distribution of a unique hyper-parameter set. Each parameter set represents a unique cluster. The cluster may refer to a particular human motion with a unique hyper-parameter set. Figure 3 illustrates human motion recognition via a 3D feature space, and further demonstrates how the three human motions (walking, running and falling) shown in the figure are dependent on the features used for clustering. Figure 3 shows that the selected features cluster human motions efficiently.

#### 4. Non-Parametric Bayesian Inference

Suppose we have  $N$  data points (observations) constituting all of the above mentioned features,  $\vec{O} = [\vec{o}_1, \vec{o}_2, \vec{o}_3, \dots, \vec{o}_N]$ , where each data point is a vector of  $D = 3$ , representing each feature's value. Our goal is then to build a posterior distribution for the set of observations where the posterior distribution represents the total number of human motions (clusters) in the data set, which can also be used to infer which data point belongs to which motion (cluster). Two generic models exist to represent and model the data points: (a) generative models and (b) discriminant models. We prefer to model our feature space using the generative model [21] rather than a discriminant model for three reasons: (1) discriminant models do not allow one to generate samples from joint distributions of two or more variables; (2) generative models are more flexible in expressing dependencies in a more complex learning environment; and (3) discriminant models are inherently supervised and cannot be extended to unsupervised learning.

The generative model is a model used to randomly generate observable data given hidden parameters. It specifies a joint distribution over observations and labels. Generative models serve two purposes in machine learning: (1) modeling data directly and (2) forming a conditional probability density function. GMM, HMM, naive Bayes (NB), and latent Dirichlet allocation are some examples of generative models. We intend to use the GMM for clustering data points, as the GMM is flexible and can be easily extended to the case in which the number of hidden clusters is unknown. Two models exist in the GMM literature: the finite Gaussian mixture model (FGMM) and the infinite Gaussian mixture model (IGMM). When the number of clusters is known *a priori*, the FGMM is used. On the other hand, if there is no prior knowledge about the number of clusters, then the IGMM can be used. Since we do not limit the number of human motions (clusters) to any fixed number, we intend to use the IGMM rather than the FGMM. However, both models are closely related to each other, so to grasp the idea of the IGMM, an understanding of the FGMM is also necessary. Before explaining the GMMs, we discuss the Dirichlet distribution that governs both models.

##### 4.1. Dirichlet Distribution

The Dirichlet distribution is a continuous multivariate distribution parameterized by the vector  $\alpha$  of the positive real. Furthermore, the multivariate generalization of the beta distribution is also a Dirichlet distribution [22]. The Dirichlet distribution is used as a prior distribution in the Bayes inference engine due to the following two reasons: (1) the Dirichlet distribution is the conjugate prior of the categorical and multinomial distribution; (2) the probability distribution function of the Dirichlet distribution results in the belief about  $K$  components (clusters) that constitutes the Gaussian mixture model.

The Dirichlet distribution of order  $K \geq 2$  with parameters  $\alpha_1, \alpha_2, \dots, \alpha_K > 0$  has a probability density function with respect to the Lebesgue measure on the Euclidean space  $R^{K-1}$  given by:

$$f(o_1, o_2, o_3, \dots, o_{K-1}) = \frac{1}{B(\alpha)} \prod_{i=1}^K o_i^{\alpha_i-1} \quad (8)$$

for all  $o_1, o_2, o_3, \dots, o_{K-1} > 0$  satisfying  $o_1 + o_2, \dots, o_{K-1} < 1$  [21,23]. The normalizing constant is a multinomial beta function, which can be expressed in terms of the gamma function:

$$B(\alpha) = \frac{\prod_{i=1}^K \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^K \alpha_i)}, \quad \alpha = (\alpha_1, \alpha_2, \dots, \alpha_K) \quad (9)$$

#### 4.2. Finite Gaussian Mixture Model

A FGMM is a hidden variable probabilistic model based on weighted multivariate Gaussian random variables. The FGMM provides an accurate approximation for multi-modal probability density estimation and clusters data points if the hidden variables are interpreted as class labels. The FGMM assumes that all of the data points are generated from a finite number of Gaussians with unknown parameters.

Recall from the previous section that we know matrix  $\vec{O} = [\vec{o}_1, \vec{o}_2, \vec{o}_3, \dots, \vec{o}_N]$ , with  $N$  number of data points where each  $\vec{o}_i$  represents a vector with  $D$  dimensions. The number of dimensions  $D$  is fixed at three in our problem, representing the features space discussed in the previous section. Figure 4 illustrates the finite Gaussian mixture model. Each cluster in the mixture constitutes weight  $\vec{\omega}_k$ , which is the probability of assigning a data point to one of cluster ( $K$ ) and  $\vec{\omega} = [\omega_1, \omega_2, \omega_3, \dots, \omega_N]$ . Since the Gaussian mixture is a multi-modal probability distribution with a different set of parameters corresponding to each cluster in the mixture, each cluster follows a Gaussian distribution with parameters  $\vec{\theta}_k$ , where  $\vec{\theta}_k$  is the vector with a mean,  $\vec{\mu}_k$  and covariance,  $\Sigma_k$ . Each data point is associated with  $c_i$ , indicating to which cluster  $K$  the data point  $o_i$  belongs. The  $c_i$  belongs to cluster  $k$  with probability  $\omega_k$ . The FGMM can be represented by:

$$\vec{\omega} | \alpha \sim Dir(\frac{\alpha}{K}, \frac{\alpha}{K}, \dots, \frac{\alpha}{K}); \vec{\theta}_k \sim \vec{H}; c_i | \vec{\omega} \sim Multinomial(\cdot | \vec{\omega}) \quad (10)$$

$$\vec{x}_i | \vec{\theta}_k \sim Gaussian(\cdot | \vec{\theta}_k) \quad (11)$$

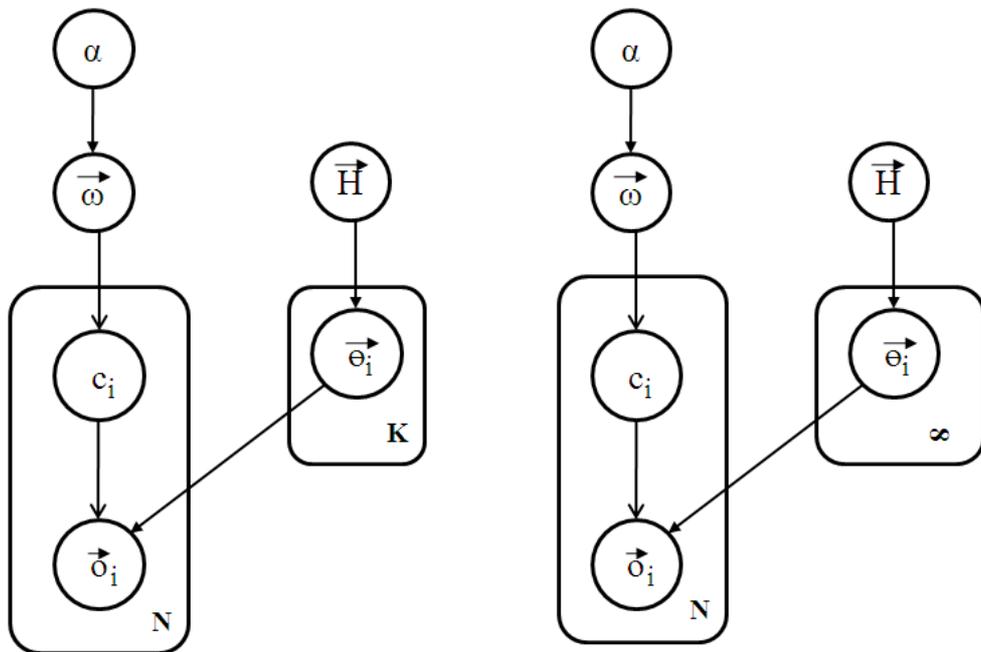
The hyper-parameters  $\vec{H}$  are the parameters representing prior knowledge, in this case incorporating our prior knowledge about the data points. It realizes the degree of our belief about the underlying system of parameters. They arise when the use of a conjugate prior is necessary to simplify the calculations for posterior estimation. The actual parameters that govern the underlying system are  $\vec{\theta}_k$ , while the hyper-parameters tend to accurately estimate the true data. Let  $\vec{L}$  be  $[\frac{\alpha}{K}, \frac{\alpha}{K}, \dots, \frac{\alpha}{K}]$ .

The FGMM is effective when the number of labels is known, but in reality, we do not know how many clusters are in the mixture. Therefore, we need to have a flexible model that does not have a fixed prior  $\vec{\theta}$ . Therefore,  $\vec{\theta}$  follows the base distribution  $\vec{H}$ , which tends to give the model flexibility.

The problem becomes challenging when we do model selection for the FGMM. In our problem, if we have knowledge about the number of human motions, then we could apply the FGMM. However, there

is no bound on the number of human motions; they may grow with time. Since it is not appropriate to use the FGMM, the IGMM is introduced in the next subsection for model selection.

**Figure 4.** Finite and Infinite Gaussian Mixture Models.



(a) Finite Gaussian Mixture Model (FGMM)      (b) Infinite Gaussian Mixture Model (IGMM)

### 4.3. Infinite Gaussian Mixture Model

The IGMM is an extension of the FGMM, where  $K \rightarrow \infty$ . It is assumed that the number of clusters tends to go to countably infinity because the number of human motions a person can perform is bounded (finite). Therefore, the IGMM can be utilized to model our problem in a generative way for a given data set. Figure 4(b) shows the IGMM as the number of clusters goes to infinity. In the FGMM, the term  $\vec{\omega}$  is dependent on  $K$ ; as the number of clusters  $K$  increases, the value of  $\vec{\omega}$  is affected. It is possible to work on the infinite dimensional model, which will integrate out  $\vec{\omega}$ . Therefore,  $\vec{\omega}$  can be marginalized out due to the Dirichlet prior because the Dirichlet prior is the conjugate of the discrete multinomial likelihood. The use of a conjugate prior helps us to integrate our complex integrations. The IGMM can be represented as follows:

$$\vec{\omega}|\alpha \sim \text{Stick}(\alpha); c_i|\vec{\omega} \sim \text{Multinomial}(\cdot|\vec{\omega}); \vec{\theta} \sim \vec{H}; o_i|c_k, \vec{\theta}_k \sim \text{Gaussian}(\cdot|\vec{\mu}_k, \Sigma_k) \quad (12)$$

where  $\vec{\theta}_k \sim \vec{H}$  and the conjugate prior for multivariate normal is inverse Wishart random variable [24].

$$\Sigma_k \sim \text{inverseWishart}(\Lambda_o); \vec{\mu}_k \sim G(\vec{\mu}_o, \frac{\Sigma_k}{K_o}) \quad (13)$$

where the Stick breaking follows beta distribution and is given as:

$$\vec{\omega} \sim \text{Beta}(1, \alpha); \omega_k = \omega'_k \prod_{j=1}^{K-1} (1 - \omega'_j); K \rightarrow \infty \quad (14)$$

where  $\omega'$  is an initial probability for each cluster. The conjugate prior for the multivariate normal distribution is the inverse Wishart, playing a significant role in the posterior estimation for labels  $c_i$ . The prior hyper-parameters for the model are given as  $\vec{H} = \{\mu_o, \kappa_o, \nu_o, \Lambda_o\}$ .

Stick breaking process is used in the IGMM to model and imparts flexibility in terms of a variable number of clusters. In the stick breaking process, a stick of unit length is assumed, which can be represented as  $\sum_{i=1}^K \omega_i = 1$ . The stick breaking process begins by breaking the stick into two parts, modeled using the Beta distribution [24], where the length of one of the parts is  $\omega_1$ , which corresponds to the weight. The same process is repeated for the remaining part of the stick ( $1 - \omega_1$ ). The countably infinite concept that we discussed in the previous subsection is realized here in the form of the Stick breaking process.

The IGMM fully models the problem under consideration, where the number of human motions is not known. Each type of motion forms a cluster with parameters  $\vec{\theta}_k$ . The parameter vector consists of  $\vec{\mu}_k$  and  $\Sigma_k$ . The mixing weights ( $\vec{\omega}$ ) are obtained from the Stick breaking process, where  $\vec{\omega}_k$  is the probability that a data point belongs to cluster  $k$ . The  $\alpha$  represents our confidence in the model parameters. Data modeling is done in this section, and we need a non-parametric clustering approach that clusters the data points based on the prior information. In the next section, we explain how we cluster data into different clusters based on non-parametric Bayesian inference using Gibbs sampling.

## 5. Non-Parametric Bayesian Inference for Motion Recognition

In this section, we focus on the non-parametric Bayesian inference model for motion recognition. We define the labels  $\vec{C} = [c_1, c_2, \dots, c_N]$  for each data point  $\vec{O}$ , where  $c_i$  indicates the cluster to which the data point  $o_i$  belongs. In this section, we answer the following questions: (1) *How many human motions have generated the data set?* (2) *Which human motion does each data point  $\vec{o}_i$  result from?* Hence, the parameter of interest is  $\vec{C}$ ; once we find  $\vec{C}$ , then we are able to answer the above questions.

The IGMM is a generic model that can be extended by numerous approaches for parameter estimation described in the literature. For example, for parameter estimation, the following methods can be utilized with the IGMM for clustering: expectation maximization (EM), Markov chain Monte Carlo (MCMC), moment matching, spectral methods, *etc.* In our problem, we restrict ourselves to the use of MCMC due to its clustering accuracy. The realization of MCMC is the Gibbs sampling algorithm [25] that we incorporate to draw inferences about the data points. In the Gibbs sampling approach, parameters are integrated out, which results in lower complexity. The Gibbs sampler is used to obtain a sequence of random samples from a joint probability distribution of more than one random variable. The random samples obtained can be used to approximate the joint probability distribution, to approximate unknown parameters, or to compute integrals [25].

Recall that we are interested in the  $\vec{C}$ , which is difficult to find due to the complex integration process. Therefore, instead of finding the joint probability distribution, we tend to calculate  $P(c_i = k | \vec{O}, \vec{C}_{-i}, \alpha; \vec{H})$ . Therefore, applying the Bayes rule, we get:

$$P(c_i = k | \vec{O}, \vec{C}_{-i}, \alpha; \vec{H}) = P(c_i = k | \vec{O}_{-i}, \alpha, \vec{\theta}_k, \vec{H}, \vec{o}_i) \quad (15)$$

$$P(c_i = k | \vec{O}, \vec{C}_{-i}, \alpha; \vec{H}) \propto P(o_i | \vec{O}_{-i}; \vec{H}) P(c_i = k | \vec{C}_{-i}, \alpha) \quad (16)$$

where  $\sim$  is introduced to replace the normalization constant from the above equation. In the equation above, the term  $\vec{H}$  is a Student t-distribution, which can be computed easily. The Student t-distribution is used to compute the mean of a normally distributed population where the sample size is small and the population standard deviation is unknown [24]. The term  $P(c_i = k | \vec{C}_{-i}, \alpha)$  in the equation is unknown, and will be calculated in the following subsection.

### 5.1. FGMM and Motion Recognition

In the context of FGMM, the number of clusters are fixed and assumed to be  $K$ . The value of interest here is  $P(c_i = k | \vec{C}_{-i}, \alpha)$  under  $K$  clusters. The FGMM is modeled in Equation (10). Let us assume that  $N$  data points are clustered and we received  $\vec{o}_{N+1}$ , then the probability of assigning this new data point to a cluster  $k$  is given below:

$$P(c_{N+1} = k | c_{1..N}; \alpha, \vec{L}) = \int P(c_{N+1} = k | \vec{\omega}) P(\vec{\omega} | c_{1..N}; \alpha, \vec{L}) d\vec{\omega} \quad (17)$$

$$= \int P(c_{N+1} = k | \vec{\omega}) P(\vec{\omega}; \alpha^*, \vec{L}^*) d\vec{\omega} \quad (18)$$

$$= E(P(c_{N+1} = k | \vec{\omega})) \quad (19)$$

$$= \frac{\alpha^* m^*}{\sum_{i=1}^K \alpha^* m_i^*} = m_k^* \quad (20)$$

Equation (17) is the marginal distribution, where  $\vec{\omega}$  is integrated out. Equation (19) is the expected value of  $\vec{\omega}$ . Equation (20) gives the marginal probability of allocating a new data point to the clusters already present, where  $m_k = \frac{\alpha}{K}$  is the number of data points in the cluster  $k$ . The posterior distribution of the weights is the Dirichlet distribution with updated prior parameters [25],

$$\alpha^* = \alpha + N; \vec{L}^* = \frac{\alpha \vec{L} + NF}{\alpha + N} \quad (21)$$

where  $F$  is empirical distribution. The cluster assignment for the  $o_{N+1}$  can be given by [21,23]:

$$P(c_{N+1} = k | c_{1..N}; \alpha, \vec{L}) = \frac{\alpha/K + n_k}{\alpha + N} \quad (22)$$

where  $n_k$  is the number of data points originating from the same cluster.

### 5.2. IGMM and Motion Recognition

Now, we will relax the limit over the  $K$  and let  $K \rightarrow \infty$ . In this section, we extend the model from finite  $K$  to infinity, therefore, when  $K \rightarrow \infty$  the FGMM transforms to the IGMM. From [21,23], we know that,

$$P(c_{N+1} = k | c_{1..N}; \alpha, \vec{L}) = \frac{n_k}{\alpha + N} \quad (23)$$

where the right side of Equation (23) is the probability that data point  $N + 1$  belongs to cluster  $k$ . However, from the property of exchangeability of the non-parametric Bayesian [24], we can replace  $N + 1$  with any value of  $i$  without changing the joint probability. Therefore, we have

$$P(c_i = k | \vec{C}_{-i}; \alpha) = \frac{n_{k,-i}}{\alpha + N - 1} \quad (24)$$

where  $n_{k,-i}$  is the number of data points assigned to the  $k$ th cluster. However, the left side of Equation (24) represents the probability of assigning a data point to an existing cluster (human motion). However, there is also the possibility that new human motion is detected and, as discussed in previous sections, that new human motion detection will follow different feature values. Therefore, we can model the new motion detection with the probability given by:

$$P(c_i \neq c_j, \forall j \neq i | \vec{C}_{-i}, \alpha) = 1 - \frac{\sum_{j=1}^K n_{j,-i}}{\alpha + N - 1} = \frac{\alpha}{\alpha + N - 1} \quad (25)$$

From Equation (25), we can see that probability of assigning a data point to a new cluster is equal to 1 minus the sum of all probabilities assigned to existing clusters. This process is called Chinese restaurant process (CRP) [26]. In the CRP the number of tables (clusters) and the number of customers sitting at a particular table (data points) can be infinity. The first customer arriving at the restaurant will always sit at the first table. However, the second customer arriving at the restaurant will want to sit at the first table or he may choose a new table, and this process continues for the rest of the customers arriving at the restaurant. The probability that a customer will sit at an already occupied table depends on the number of customers already seated at the table  $m_k$ ; given in Equation (24). However, the probability of sitting at a new table is proportional to  $\alpha$ ; given in Equation (25).

At this stage, we need Gibbs sampler to obtain the samples of  $\vec{C}$ . After observing  $N - 1$  data points, we can update priors using Bayes rule. Therefore, the probability of assigning a data point to the represented cluster is given as:

$$P(c_i = k | \vec{C}_{-i}, \alpha, \vec{\theta}, \vec{H}, \vec{O}) = P(c_i = k | \vec{C}_{-i}, \alpha) P(\vec{o}_i | \vec{\theta}_k) \quad (26)$$

where  $\vec{C}_{-i}$  is the set of observations currently assigned to cluster  $i$  except  $o_i$ ,  $P(c_i = k | \vec{C}_{-i}, \alpha)$  in the Equation (26) follows the Chinese restaurant process and  $P(\vec{o}_i | \vec{\theta}_k)$  follows Gaussian with parameters  $\vec{\theta}_k$ . The probability that a data point belongs to a represented cluster is given by:

$$P(c_i \neq j, \forall j \neq i | \vec{C}_{-i}, \alpha, \vec{\theta}, \vec{H}, \vec{O}) = \frac{\alpha}{\alpha + N - 1} \int_{\vec{\theta}} P(\vec{o}_i | \vec{\theta}) P(\vec{\theta} | \vec{H}) d\vec{\theta} \quad (27)$$

The integration is the marginal probability, where  $\vec{\theta}$  is integrated out. The use of the conjugate prior has made the integration tractable. Moreover, we know that the conjugate prior of the multivariate normal distribution is the inverse Wishart. By integrating out the parameter  $\vec{\theta}$  the sample size of the problem is reduced and the Gibbs sampler will converge early. Therefore, we adopt collapsed Gibbs sampler [27] to increase the performance in terms of reduced complexity.

### 5.3. Collapsed Gibbs Sampler

Remember that our goal is to estimate the posterior distribution for the infinite Gaussian mixture model using the Gibbs sampler. Therefore, we have already chosen the inverse Wishart distribution which make it possible integrate out parameters  $\vec{\theta}_k$ . In the collapsed Gibbs sampling approach, the parameter  $\vec{\theta}_k$  is integrated out. Therefore, it is possible to obtain a closed-form solution for the posterior distribution. From [23], we have,

$$P(c_i = k | \vec{C}_{-i}, \alpha, \vec{H}, \vec{O}) = P(c_i = k | \vec{O}_{-i}, \alpha) P(\vec{o}_i | \vec{O}_{k,-i}; \vec{H}) \tag{28}$$

The first term on the right side of Equation (28) follows the Chinese restaurant process and is given by Equation (23). However, the second term follows the multivariate Student-t [24]. The hyper-parameters  $\vec{H}$  is used to simplify the integration steps in the posterior estimation. Therefore, we choose the inverse Wishart as a prior for the  $\Sigma_k$  distribution and Gaussian distribution for  $\vec{\mu}_k$ .

$$\Sigma_k \sim \text{inverseWishart}_{v_o}(\Lambda_o^{-1}) \tag{29}$$

$$\vec{\mu}_k \sim \text{Gaussian}(\vec{\mu}_o, \Sigma_k / \kappa_o) \tag{30}$$

From [24], we can write the second term in the Equation (28) as,

$$P(\vec{o}_i | \vec{O}_{k,-i}; \vec{H}) \sim t_{v_n - D + 1} \left[ \vec{\mu}_n, \frac{\Lambda_n, (\kappa_n + 1)}{\kappa_n (v_n - D + 1)} \right] \tag{31}$$

where

$$\vec{\mu}_n = \frac{\kappa_o}{\kappa_o + N} \vec{\mu}_o + \frac{N}{\kappa_o + N} \vec{O} \tag{32}$$

$$\kappa_n = \kappa_o + N, v_n = v_o + N \tag{33}$$

$$\Lambda_n = \Lambda_o + S + \frac{\kappa_o n}{\kappa_o + N} ((\vec{O}) - \vec{\mu}_o)(\vec{O} - \vec{\mu}_o)^T \tag{34}$$

$$\vec{O} = (\vec{o}_1, \vec{o}_2, \dots, \vec{o}_N) / N \tag{35}$$

where  $D$  is the number of dimension,  $S$  is the sample variance of the observations  $\vec{O}$ , and  $\vec{\mu}_n, \kappa_n, v_n, \Lambda_n$  are the updated hyper-parameters.  $n$  is the number of observations in that particular cluster. The advantage of integrating out the parameters is that it reduces the sample space, resulting in a quick convergence [26]. Using Equation (31), the distribution can be determined as a multivariate Student-t distribution with hyper-parameters,

$$P(\vec{o}_i; \vec{H}) \sim t_{v_n - D + 1} \left[ \vec{\mu}_n, \frac{\Lambda_n, (\kappa_n + 1)}{\kappa_n (v_n - D + 1)} \right] \tag{36}$$

The above results in two cases for clustering. One is the case when data point belongs to one of the represented cluster from  $K$  clusters. In the second case, the data point belongs to a new cluster. The first case can be represented as,

$$P(c_i = k | \vec{C}_{-i}, \vec{O}, \alpha, \vec{H}) \sim \frac{m_{k,-i}}{\alpha + N - 1} t_{v_n - D + 1} \left[ \vec{\mu}_n, \frac{\Lambda_n, (\kappa_n + 1)}{\kappa_n (v_n - D + 1)} \right] \tag{37}$$

In the second case when the data point observed is assigned a new cluster, it can be represented as,

$$P(c_i \neq j, \forall j \neq i | \vec{C}_{-i}, \vec{O}, \alpha, \vec{H}) \sim \frac{\alpha}{\alpha + N - 1} t_{v_n - D + 1} \left[ \vec{\mu}_n, \frac{\Lambda_n, (\kappa_n + 1)}{\kappa_n (v_n - D + 1)} \right] \quad (38)$$

The posterior derived is used in the collapsed Gibbs sampler.

---

**Algorithm 1:** Non-parametric Bayesian clustering algorithm for human motion recognition using collapsed Gibbs sampler

---

**Input:**  $\vec{O}(\vec{o}_i, D)$ , Sweeps,  $\Gamma(a, b)$ ,  $\vec{H}$

**Output:**  $K, c_i$

```

1 begin
2    $\vec{C} \leftarrow c_1, c_2, \dots, c_K$ 
3    $K_+ \leftarrow 0$ ;
4   foreach  $s$  in Sweeps do
5      $C_s \leftarrow C_{s-1}$ 
6     foreach  $s$  in Sweeps do
7        $m_{-i} \leftarrow \sum_{j=1}^K (c_j == c_i) - 1$ ;
8       if  $m_{-i} == 0$  then
9          $c_j \leftarrow c_j - 1; \forall j > i$ 
10         $K_+ \leftarrow K_+ - 1$ 
11      end
12       $\text{dim} \leftarrow \text{length}(\mu_D)$ 
13      foreach  $i$  in  $K$  do
14         $\text{covariances}(i, \text{dim}) \leftarrow \text{iWishart}(\lambda, v)$ 
15         $\text{means} \leftarrow \text{MVRnd}(\mu, \text{covariances}(i, \text{dim}))$ 
16         $m_k = \sum_{i=1}^R \mathbb{I}(c_i = k)$ 
17         $c_i \leftarrow \text{MVRnd}(\text{means}, \text{covariances})$ 
18      end
19      /* Estimate the prior  $P(\vec{o}_i | \vec{O}; \vec{H})$  using Equation (31) */
20      if  $c_i > K_+$  then
21         $K_+ \leftarrow K_+ + 1$ 
22      end
23      /* Estimate the prior  $P(c_i = k | \vec{C}_{-i}, \alpha)$  using Equations (23) and (24) */
24    end
25  end
26  /* Collapsed Gibbs Sampling */
27  /* Estimate the posterior  $P(c_i = k | \vec{O}, \vec{C}_{-i}, \alpha; \vec{H})$  using Equation (37) */
28
29 end

```

---

In Algorithm 1, input parameters are the measured data point from MEMS sensor, the number of sweep to monitor the convergence of algorithm, and hyper-parameters. The data model is estimated using the observed data and hyper-parameters. Algorithm 1 runs for the specified number of sweeps (steps) till convergence after initialization procedure. For example, the loop in the step 6 runs for 100 sweeps. The samples obtained for each data point from the conditional distribution are used to estimate

the joint distribution of all variables. In Algorithm 1, priors are estimated and updated for each step in the Gibbs sampler. After repeated estimation and update process, the final posterior estimation can be obtained at the last step in Algorithm 1.

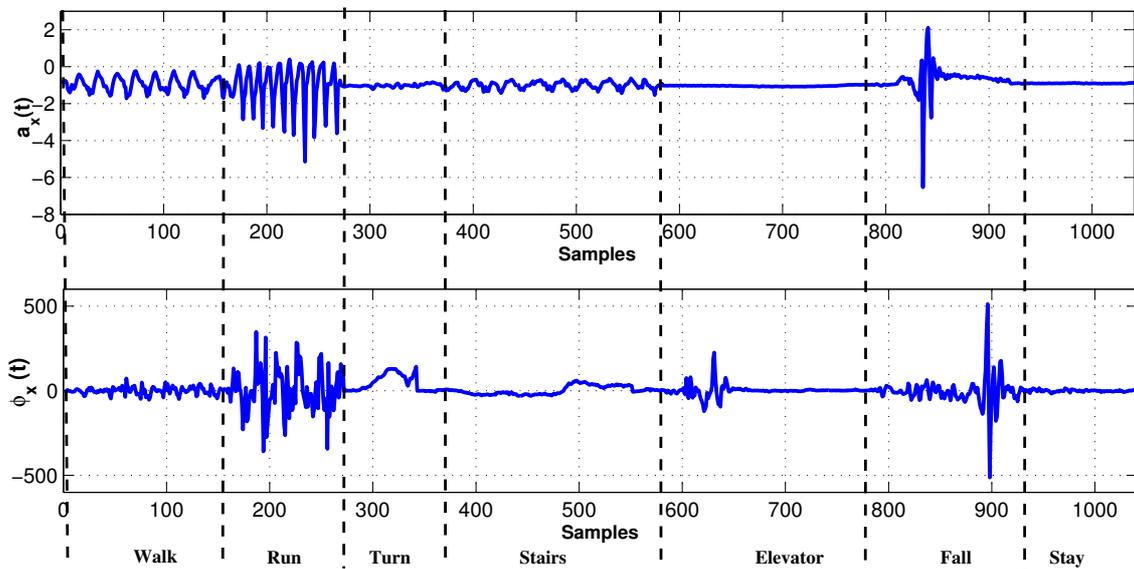
#### 5.4. Motion Recognition Method

The result we get from the collapsed Gibbs sampler is the indicators  $\vec{C}$ , and the indicators are associated with every data point that is input into the collapsed Gibbs sampler. The indicator  $\vec{C}$  obtained as an output from the collapsed Gibbs sampler answers the two questions that we are supposed to answer. Recall from Section 5, the questions we have to answer are (1) how many clusters represent the data set and (2) which data point belongs to which cluster. As every data point is associated with its respective indicator ( $c_i, i = 1 : N$ ), the indicator demonstrates to what cluster the data point ( $\vec{o}_i, i = 1 : N$ ) belongs. The first question can be answered by counting the unique values of  $c_i$ . The last step in motion recognition is motion mapping. The mean and variance of each cluster can be used to identify human motion. We assume that we are given each motion-dependent value for the feature space. We compare those with the values obtained from the collapsed Gibbs sampler, and the similarity between the two distributions is measured via the Kullback–Leibler divergence. The KLD value is computed between each cluster obtained from the non-parametric Bayesian and the reference training data. The decision is made over a minimum value of KLD.

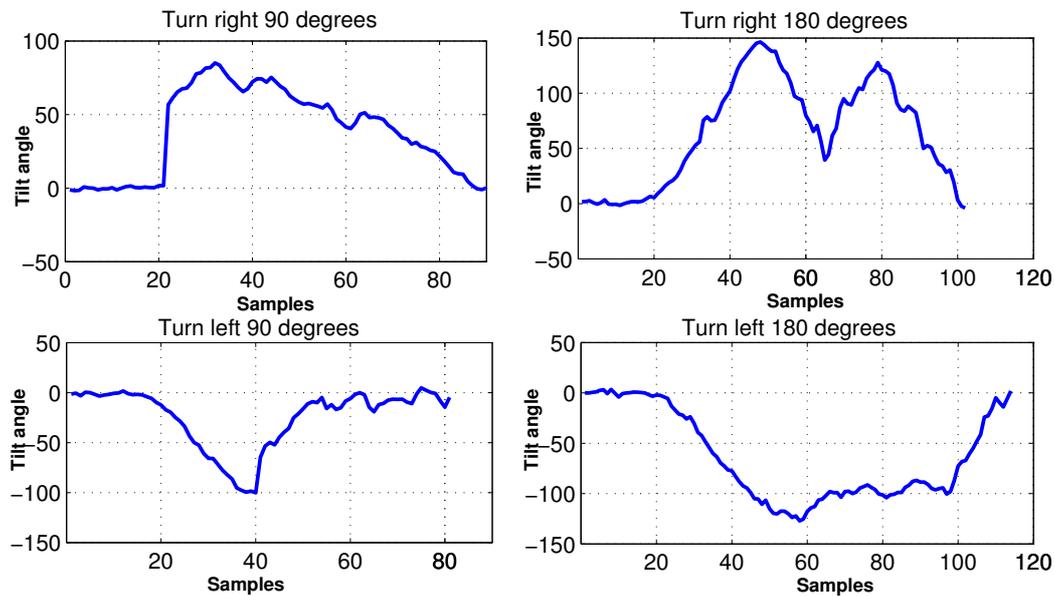
## 6. Experimental Setup

In experiment, the MEMS accelerometer device was mounted to the chest of the individual as shown in Figure 2. The data set for our experiments was collected in an unsupervised way. Every motion was performed by fifty different individuals without any supervision. In consequence, fifty data sets are acquired for each motion, where the distance for walk and run was 10 meters. Data sets were acquired using USB (Universal Serial Bus) interface from the MEMS device. The sampling rate for data acquisition was 45 Hz/sample. We considered human motions such as standing, walking, running, turning, falling, taking the stairs, and taking the elevator, all of which can happen in a building environment. The acceleration ( $a_x$ ) and tilt angle ( $\phi_x$ ) in  $x$  axis are depicted for different human motions in Figure 5. It is evident from Figure 5 that different motions show different patterns. In Figure 6, the plots for tilt angle are shown for left and right turn. It is clear from the plots in Figure 6 that the left and right turns show distinguishing characteristics for tilt angle. The tilt angle is a useful feature to cluster human motions.

**Figure 5.** The acceleration ( $a_x$ ) and tilt angle ( $\phi_x$ ) signals of human motions obtained from the MEMS sensor.



**Figure 6.** Tilt angle signals for turning motions.



### 6.1. Hyper-Parameters

In this section, we will explain the hyper-parameters that are useful in carrying out the clustering procedure. Since we rely on the IGMM, we have to set the hyper-parameters in such a way that represents our actual true data set. The complexity of the Gibbs sampler grows significantly as the number of data points increases [21,23]. Therefore, it is desirable to set hyper-parameters such that the true data is obtained faster in order to reduce the Gibbs sampler complexity.

Since we have employed the generative model, we have to setup the following hyper-parameters,  $\vec{H} = \{\vec{\mu}_o, \kappa_o, \Lambda_o, \nu_o\}$  and  $\alpha$ , where  $\alpha$  is the concentration parameter that encodes the number of clusters

in the data set. As the value of  $\alpha$  increases, the number of clusters also increases, which is evident from Equation (25). In our scenario we keep the value of  $\alpha$  small initially because the number of human motions in the initial stage is assumed to be smaller.

The mean vector  $\vec{\mu}_o$  of a cluster follows a Gaussian distribution with mean  $\vec{\mu}_o$  and covariance  $\Sigma_o/\kappa_o$ . Therefore, the two hyper-parameters  $\vec{\mu}_o$  and  $\kappa_o$  represent the mean for the cluster. Where  $\kappa_o$  is the dispersion measure of the cluster, the higher is the value of  $\kappa_o$ , the closer the clusters are to each other, and vice versa.

The covariance matrix of the cluster depends on the following two hyper-parameters,  $v_o$  and  $\Lambda_o$ . These parameters are related to the inverse Wishart distribution. Parameter  $v_o$  is the degree of freedom while  $\Lambda_o$  quantifies the variability around the mean of the feature space. Suppose that feature point TAS shows a wide range of variability from its mean value, which is specific to a particular human motion such as running. Therefore, this variability is quantified by  $\Lambda_o$ . The value of  $v_o$  represents our confidence about  $\Lambda_o$ .

## 6.2. Mapping Cluster to Motion

The Gibbs sampler results in a number of clusters that constitute the data set, as well as each data point's association with a particular cluster. We assume that the mean values of the features are available for all of the human motions. After clustering, we compare and then map the clusters to the human motion.

## 7. Experimental Results

In this section, we present the performance analysis of the proposed motion detection technique, in addition to comparing our proposed technique with the parametric Fuzzy C-Means (FCM) [28], the unsupervised K-Means clustering algorithm [29], the non-parametric mean-shift clustering method [30]. In the performance analysis, we study the effect of varying the number of human motions. The results demonstrate the effect of introducing new human motion on the clustering accuracy. In the following subsection, we briefly discuss the K-means and mean-shift clustering approaches.

### 7.1. K-means Clustering

In the K-means clustering algorithm, the  $N$  data points are partitioned into  $k$  clusters, where each observation belongs to the cluster with the nearest mean. Given a set of observations  $\vec{o}_1, \vec{o}_2, \dots, \vec{o}_N$ , where each observation  $\vec{o}_i$  is a  $D$ -dimensional vector, the K-means clustering algorithm tends to cluster those  $N$  observations into  $K$  sets, where ( $K \leq N$ ) and  $K = [k_1, k_2, \dots, k_K]$ , in order to minimize the intra-cluster sum of squares.

$$\arg \min \sum_{i=1}^K \sum_{\vec{o}_j \in k_i} \|\vec{o}_j - \vec{\mu}_i\|^2 \quad (39)$$

where  $\vec{\mu}_i$  is the mean of points in  $k_i$ .

The overall algorithm proceeds in two steps. Initially it is assumed that the initial means for the  $K$  clusters are given by  $h_1^1, h_2^1, \dots, h_K^1$ , where the two steps for the algorithm are the assignment step and the update step. In the assignment step each observation is assigned to a cluster with minimum mean, represented as,

$$k_i^{(t)} = \{\vec{o}_p : \|\vec{o}_p - \vec{\mu}_i^{(t)}\| \leq \|\vec{o}_p - \vec{\mu}_j^{(t)}\| \forall 1 \leq j \leq k\} \quad (40)$$

where each  $\vec{o}_p$  goes to one of the  $k_i^{(t)}$ . In the update step new means are calculated based on the observations.

$$h_i^{(t+1)} = \frac{1}{|k_i^{(t)}|} \sum_{\vec{o}_j \in k_i^{(t)}} \vec{o}_j \quad (41)$$

The algorithm converges when there are no updates. The k-means in  $D$ -dimensions is NP hard.

### 7.2. Mean-Shift Clustering

Mean-shift is a non-parametric clustering algorithm [30]. The mean-shift procedure is obtained by successive computation of the means-shift vector  $\vec{\theta}_i$  and translation of the kernel  $G(\vec{o}_j)$  by  $\vec{\theta}_i$ . The mean-shift procedure is guaranteed to converge at a nearby point where the kernel estimate has zero gradient [30]. It is useful for detecting modes of the density function, which is iterative method with initial start. Here we use Gaussian kernel function for the distance to the current estimate given as,  $G(\vec{o}_j - \vec{\mu}_i) = \exp(-c\|\vec{o}_j - \vec{\mu}_i\|^2)$ . The weighted mean of the density for cluster  $k$  is given as,

$$\vec{\theta}_i = \frac{\sum_{\vec{o}_i \in N(\vec{o}_i)} \exp(-c\|\vec{o}_j - \vec{\mu}_i\|^2) \vec{o}_i}{\sum_{\vec{o}_i \in N(\vec{o}_i)} \exp(-c\|\vec{o}_j - \vec{\mu}_i\|^2)} - \vec{o}_j \quad (42)$$

where  $N(\vec{o}_i)$  is a set of data points for which  $G(\vec{o}_i) \neq 0$ . Similarly, assignment is done  $\vec{o}_i = \vec{\theta}_i$  in iterative way until the convergence point is reached.

### 7.3. Performance Evaluation Criteria

The accuracy of the proposed non-parametric Bayesian inference can be evaluated using the following three metrics:

- (1) The hit rate for detecting the right number of human motions in the data set. The percentage is calculated for the correct number of human motions detected over the total number of trials performed.
- (2) The hit rate for each data point is realized by assigning every feature point to its correct cluster. It is the percentage of feature points assigned to its correct cluster over the total number of feature points.
- (3) The false alarm rate can be computed by counting the data points that are assigned to incorrect clusters.

#### 7.4. Unforeseen Motion Detection with Limited Data

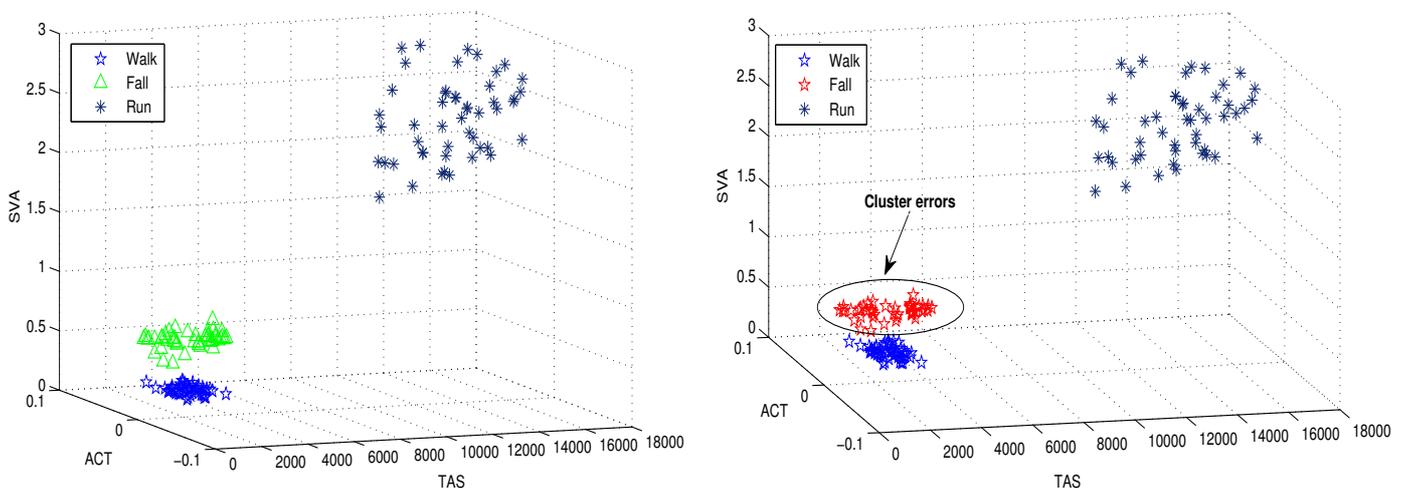
In this subsection, we will show the efficacy of the proposed motion detection method against unforeseen motions. As discussed in the previous section, human motions vary with time and encountering new motions is anticipated. Moreover, the proposed HMR approach is robust in terms of new motion detection and recognition. For example, a person is walking and instantly falls due to some unavoidable cause. The person's fall is a new event that should be detected and recognized precisely. Therefore, we compared our proposed approach with a parametric Fuzzy C-Means clustering algorithm [28]. In Fuzzy clustering, the data point can belong to more than a single cluster, and a set of membership levels can be associated with each data point based on the strength of a data point's association with a particular cluster. In FCM, each data point  $\vec{o}_i$  has a coefficient that represents its association with a particular cluster  $\vec{\omega}_k(\vec{o}_i)$ . In FCM, the centroid of a cluster of the mean of all data points is a weighted measure belonging to the cluster. This can be represented as

$$Q_k = \frac{\sum_{\vec{o}} \omega_k(\vec{o}) \vec{o}}{\sum_{\vec{o}} \omega_k(\vec{o})} \quad (43)$$

The algorithm also tend to minimize intra-cluster variance.

The accuracies of the proposed and FCM approaches are given in Figure 7. It is evident from the Figure 7 that the FCM clustering approach fails to recognize the new motion (fall) due to its parametric clustering procedure. However, the proposed technique accurately identifies the new motion with 100% accuracy. The performance evaluation is shown in Table 1. In the FCM approach, the new motion is not detected and all of the data points are classified into two clusters (motions). However, the proposed HMR detects the unanticipated fall and clusters with 99.33% accuracy. For the performance metric 1, the proposed approach clusters the data points into three clusters with 100% accuracy, where the PCM clusters the same data points into two clusters.

**Figure 7.** Unforeseen motion detection results by proposed (left) and FCM clustering algorithm (right).



**Table 1.** Performance metrics results for FCM and proposed HMR.

Motion	FCM		Proposed HMR	
	Type 2: Hit Accuracy	Type 3: False Alarm	Type 2: Hit Accuracy	Type 3: False Alarm
	(%)	(%)	(%)	(%)
Walk	100	66.67	100	0
Run	100	0	100	0
Fall (new)	0	100	99.33	0.67

### 7.5. Recognition Results for Numerous Motions

In this subsection, we show the recognition accuracies for routine daily motions. We compare the proposed HMR technique with the K-means, and observe the performance gains in terms of the performance metric criteria discussed above. Since we used a generative model, we need to set the hyper-parameters in such a way that the hit rate is maximized with a minimum number of errors. Note that the clustering results are highly sensitive to the hyper-parameters. Therefore, the hyper-parameters must be set carefully in order to reduce the chance of cluster errors. We set the values of the hyper-parameters as  $\vec{H} = \{\Lambda_o, \vec{\mu}_o, \kappa_o, \nu_o\}$ , and  $\kappa_o = 0.1$ ,  $\nu = 4$ ,  $\Lambda = \text{diag}(0.3)$ , and  $\Gamma(3, 2)$ .

### 7.6. Varying Number of Human Motions

In the simulations, we compared the proposed approach with the K-Means clustering algorithm. Figure 8 shows the clustering results for the proposed human motion approach and K-Means clustering. The clustering errors produced by the K-Means are shown in Figure 8. The proposed approach efficiently clustered the human motions.

**Figure 8.** Clustered results for the proposed algorithm (left) and the K-Means method (right) with six human motions.

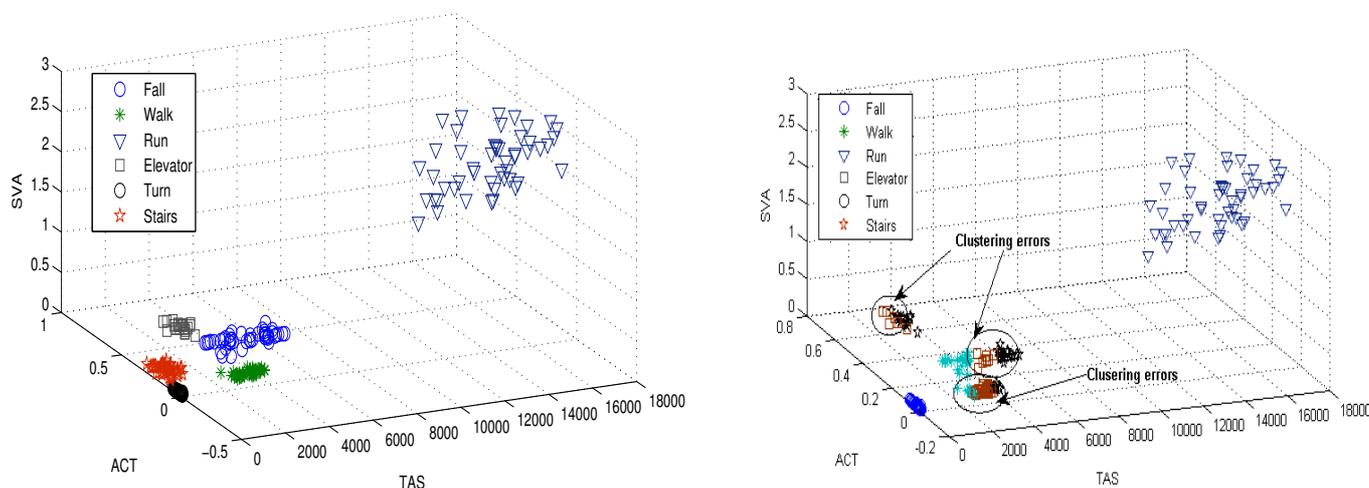
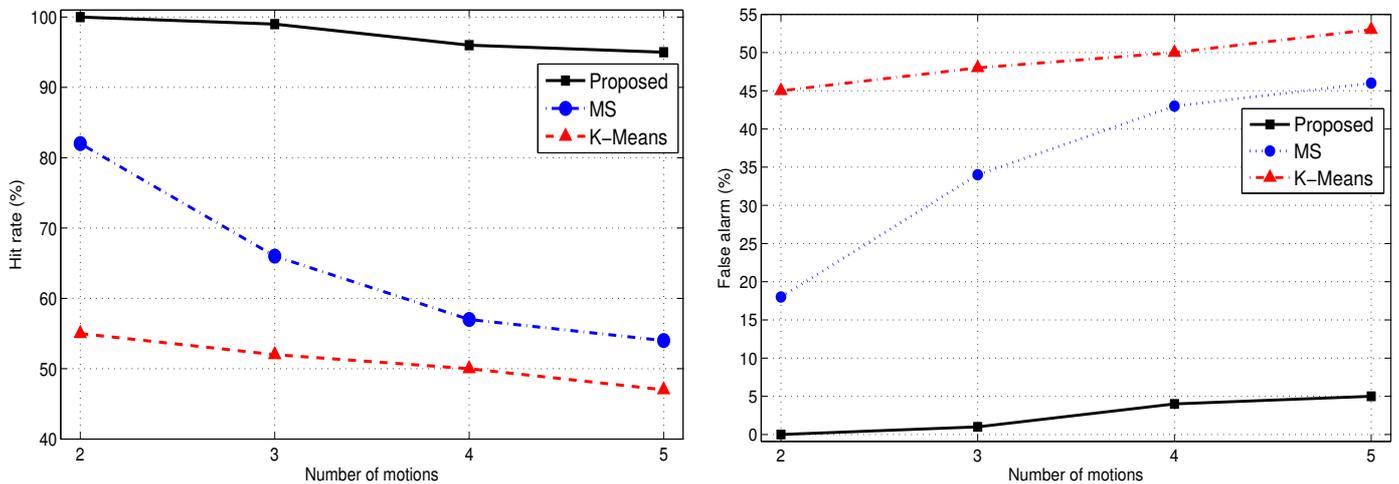


Figure 9 demonstrates the accuracy and clustering errors for the proposed method, K-Means, and mean-shift clustering approaches. The Type 2 hit rate of the proposed approach outperforms that of

the K-Means and mean-shift approaches in terms of accurate data point clustering. The unsupervised K-Means and mean-shift algorithms suffer a significant loss in accuracy.

**Figure 9.** Hit rate (left) and false alarm (right) results for the proposed algorithm, the K-Means method, and the mean-shift method with varying number of motions.

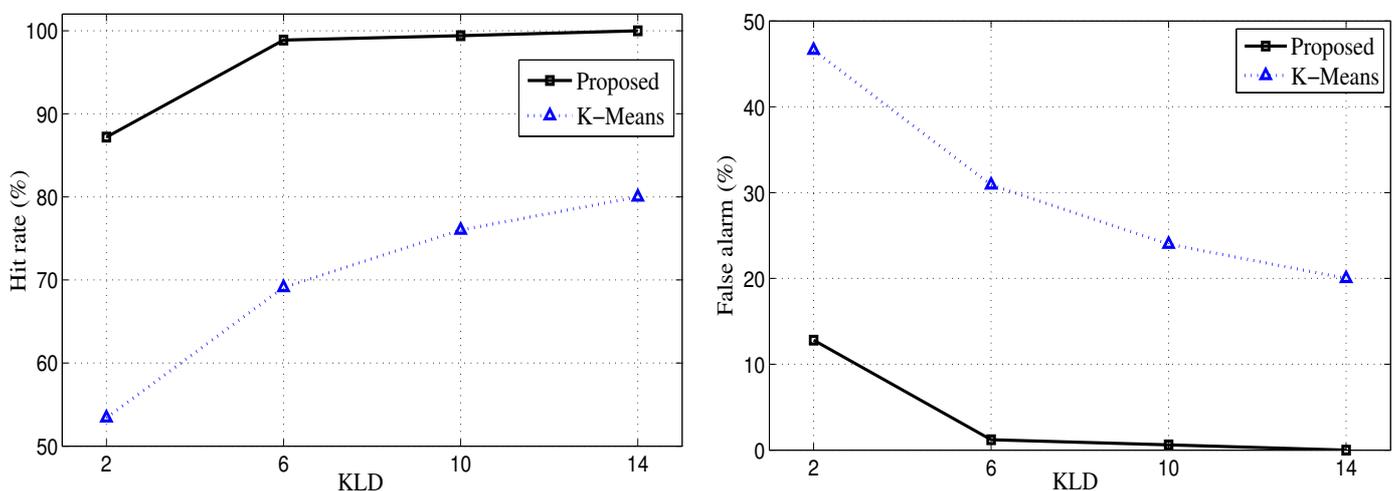


7.7. Kullback–Leibler Divergence and Clustering Accuracy

In this section, we compared the accuracy of the proposed approach with that of the K-Means approach with a varying Kullback–Leibler divergence value. The Kullback–Leibler divergence is the measure of the difference between two probability distributions. Equation (44) represents the measure of the KLD, where D is the number of dimensions in the data set.

$$KLD = \frac{1}{2} [trace(\Sigma_2^{-1}\Sigma_1) + (\vec{\mu}_2 - \vec{\mu}_1)^T \Sigma_2^{-1} (\vec{\mu}_2 - \vec{\mu}_1) - \ln(\frac{det\Sigma_1}{det\Sigma_2} - D)] \tag{44}$$

**Figure 10.** Clustering accuracy (left, hit rate; right, false alarm) with KLD variations for the proposed algorithm and the K-Means method.

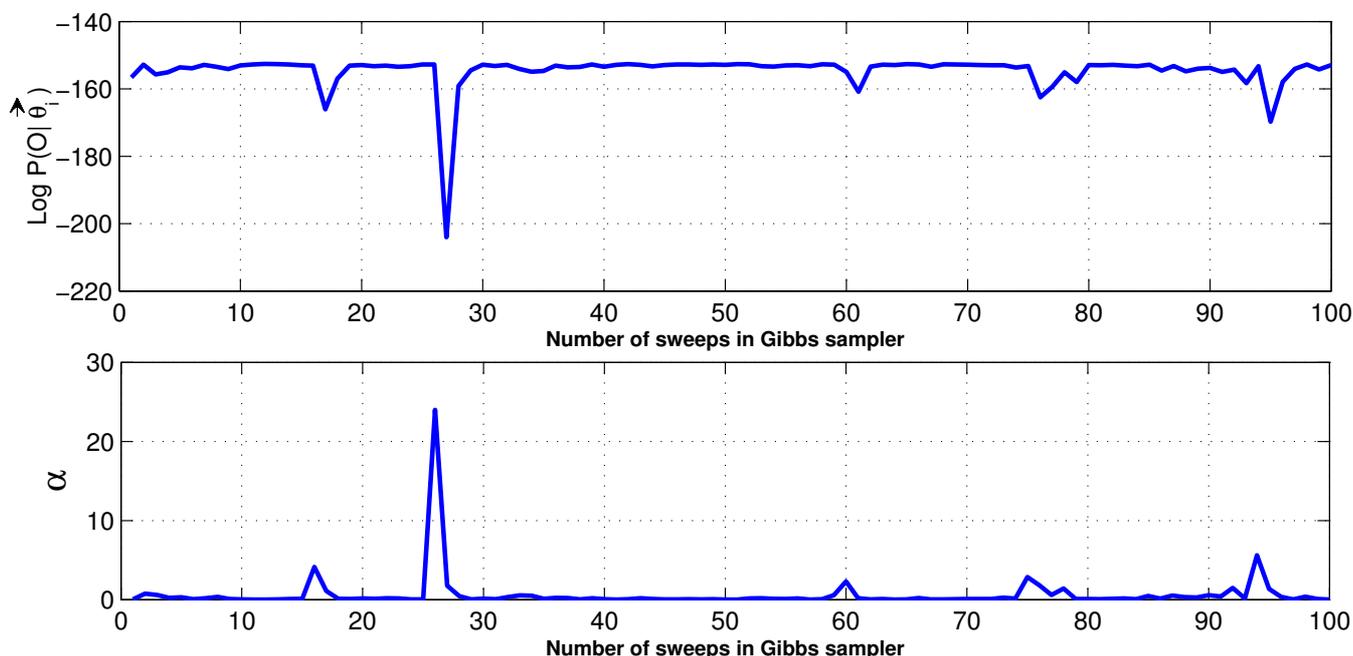


In this section, we show how the clustering results are affected by varying the KLD for the proposed and K-Means clustering approaches. In Figure 10, it is shown that the proposed approach outperforms the K-Means algorithm in terms of its hit rate and false alarm. It is evident from Figure 10 that the hit accuracy of the proposed approach is increased to 100% when the KLD increases, but the K-Means algorithm fails to achieve such high hit rate. The proposed approach attains 100% accuracy at KLD=14 for the clustering data points as shown in Figure 10.

### 7.8. Convergence

From the simulation results, it is apparent that the proposed collapsed Gibbs sampler converges to a stable state after a few iterations. Figure 11 demonstrates the convergence rate of the collapsed Gibbs sampler. The plot of the random variables are generated, for the log probability of training data under the model and the distribution over the Dirichlet hyper-parameter  $\alpha$ , in Figure 11. The two graphs are useful in diagnosing the convergence of the Gibbs sampler. Even though the Gibbs sampler converges [31], it is difficult to diagnose the convergence of Gibbs sampler in practice; therefore, visual graphs can help to decide on the convergence. It is clear that the convergence is attained after 30 iterations for the given data set as shown in Figure 11.

**Figure 11.** Convergence characteristics of the collapsed Gibbs sampler.



## 8. Conclusions

In this study, we proposed a non-parametric Bayesian approach for detecting and clustering various human motions. The proposed work exploits the motion-dependent signal features to model an available data set using the infinite Gaussian mixture model. The collapsed Gibbs sampler is utilized to classify the available data set into various human motions. The experimental results show that the proposed human motion recognition approach significantly outperforms methods including the Fuzzy-C Means,

K-Means, and mean-shift approaches. The unsupervised nature of the proposed scheme relaxes the upper bound for the number of human motions under consideration. Therefore, the proposed approach can be extended to many other applications in which the number of underlying clusters is unknown.

## References

1. Khan, A.M.; Lee, Y.K.; Lee, S.Y. A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer. *IEEE Trans. Inf. Technol. Biomed.* **2010**, *14*, 1166–1172.
2. Lee, M.W.; Khan, A.M.; Kim, T.S. A single tri-axial accelerometer-based real-time personal life log system capable of human activity recognition and exercise information generation. *Pers. Ubiquitous Comput.* **2011**, *15*, 887–898.
3. Ji, X.; Liu, H. Advances in view-invariant human motion analysis: A review. *IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev.* **2010**, *40*, 13–24.
4. Turaga, P.; Chellappa, R.; Subrahmanian, V.S.; Udreă, O. Machine recognition of human activities: A survey. *IEEE Trans. Circuits Syst. Video Technol.* **2008**, *11*, 1473–1488.
5. Tapia, E.; Intille, S.; Larson, K. Activity recognition in the home using simple and ubiquitous sensors. *Pervasive Comput.* **2004**, *11*, 158–175.
6. Sekine, M.; Tamura, T.; Akay, M.; Fujimoto, T.; Togawa, T.; Fukui, Y. Discrimination of walking patterns using wavelet-based fractal analysis. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2002**, *10*, 188–196.
7. Cilla, R.; Patricio, M.A.; Garcia, J.; Berlanga, A.; Molina, J.M. Recognizing human activities from sensors using hidden markov models constructed by feature selection techniques. *Algorithms* **2009**, *2*, 282–300.
8. Slyper, R.; Hodgins, J.K. Action Capture with Accelerometers. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Dublin, Ireland, 7–9 July 2008; pp. 193–199.
9. Ravi, N.; Dandekar, N.; Mysore, P.; Littman, M.L. Activity Recognition from Accelerometer Data. In *Proceedings of the National Conference on Artificial Intelligence*, Pittsburgh, PA, USA, 9–13 July 2005; Volume 20, pp. 1541–1546.
10. Mantyjarvi, J.; Himberg, J.; Seppanen, T. Recognizing Human Motion with Multiple Acceleration Sensors. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, Tucson, AZ, USA, 7–10 October 2001; Volume 2; pp. 747–752.
11. Bao, L.; Intille, S. Discrimination of walking patterns using wavelet-based fractal analysis. *Act. Recognit. User-Annot. Accel. Data* **2004**, *10*, 1–17.
12. Ugolotti, R.; Sassi, F.; Mordonini, M.; Cagnoni, S. Multi-sensor system for detection and classification of human activities. *J. Ambient Intell. Humaniz. Comput.* **2011**, *1*, 1–15.
13. Bao, L.; Intille, S. Activity recognition from user-annotated acceleration data. *Pervasive Comput.* **2004**, *1*, 1–17.
14. Foerster, F.; Smeja, M.; Smeja, M. Detection of posture and motion by accelerometry: A validation study in ambulatory monitoring. *Comput. Hum. Behav.* **1999**, *15*, 571–583.

15. Ermes, M.; Parkka, J.; Mantyjarvi, J.; Korhonen, I. Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. *IEEE Trans. Inf. Technol. Biomed.* **2008**, *12*, 20–26.
16. Field, M.; Stirling, D.; Naghdy, F.; Pan, Z. Mixture Model Segmentation for Gait Recognition. In *Proceedings of the 2008 ECSIS Symposium on Learning and Adaptive Behaviors for Robotic Systems, IEEE LAB-RS'08*, Edinburgh, Scotland, 6–8 August 2008; pp. 3–8.
17. Shi, G.; Zou, Y.; Jin, Y.; Cui, X.; Li, W.J. Towards HMM Based Human Motion Recognition Using MEMS Inertial Sensors. In *Proceedings of the IEEE International Conference on Robotics and Biomimetics, ROBIO*, Bangkok, Thailand, 14–17 December 2009; Volume 12, pp. 1762–1766.
18. Herzog, D.; Krger, V.; Grest, D. Parametric Hidden Markov Models for Recognition and Synthesis of Movements. In *Proceedings of the British Machine Vision Conference*, Leeds, UK, 1–4 September 2008; Volume 12, pp. 162–176.
19. Khan, A.M.; Lee, Y.K.; Kim, T.S. Accelerometer Signal-Based Human Activity Recognition Using Augmented Autoregressive Model Coefficients and Artificial Neural Nets. In *Proceedings of the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, Vancouver, BC, Canada, 20–25 August 2008; Volume 12, pp. 5172–5175.
20. Bouten, C.V.; Koekkoek, K.T.; Verduin, M.; Kodde, R.; Janssen, J.D. A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity. *IEEE Trans. Biomed. Eng.* **1997**, *44*, 36–147.
21. Ahmed, M.E.; Song, J.B.; Nguyen, N.T.; Han, Z. Nonparametric Bayesian Identification of Primary Users' Payloads in Cognitive Radio Networks. In *Proceedings of the IEEE International Conference on Communication, ICC12*, Ottawa, ON, Canada, 10–15 June 2012.
22. Samuel, K.; Balakrishnan, N.; Norman, L. *Continuous Multivariate Distributions*, 2nd ed.; Wiley: Weinheim, Germany, 2000.
23. Nguyen, N.; Zheng, R.; Han, Z. On Identifying primary user emulation attacks in cognitive radio systems using nonparametric bayesian classification. *IEEE Trans. Signal Process.* **2012**, *99*, 5172–5175.
24. Gelman, A.; Carlin, J.B.; Stern, H.S.; Rubin, D.B. *Bayesian Data Analysis*; CRC Press: Boca Raton, FL, USA, 2004; p. 88.
25. Ishwaran, H.; James, L.F. Gibbs sampling methods for stick-breaking priors. *J. Am. Stat. Assoc.* **2001**, *96*, 161–173.
26. Wood, F.; Black, M.J. A nonparametric Bayesian alternative to spike sorting. *J. Neurosci. Methods* **2008**, *173*, 1–12.
27. Liu, J.S. The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem. *J. Am. Stat. Assoc.* **1994**, *89*, 958–966.
28. Hopner, F.; Hoppner, F.; Klawonn, F.; Kruse, R. *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*; John Wiley & Sons: Hoboken, NJ, USA, 1999.
29. Hartigan, J.A.; Wong, M.A. Algorithm AS 136: A k-means clustering algorithm. *J. R. Stat. Soc. Series C Appl. Stat.* **1979**, *28*, 100–108.

30. Comaniciu, D.; Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *5*, 603–619.
31. Sujit, K.S.; Gareth, O.R. On Convergence of the EM algorithm and the Gibbs sampler. *Stat. Comput.* **1999**, *9*, 55–64.

© 2012 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).