Sensor-based Human Activity Mining Using Dirichlet Process Mixtures of Directional Statistical Models

Lei Fang*, Juan Ye† and Simon Dobson‡
School of Computer Science, University of St Andrews
St Andrews, UK
Email: *lf28@st-andrews.ac.uk, †juan.ye@st-andrews.ac.uk, ‡simon.dobson@st-andrews.ac.uk

***Abstract*—We have witnessed an increasing number of activity-aware applications being deployed in real-world environments, including smart home and mobile healthcare. The key enabler to these applications is sensor-based human activity recognition; that is, recognising and analysing human daily activities from wearable and ambient sensors. With the power of machine learning we can recognise complex correlations between various types of sensor data and the activities being observed. However the challenges still remain: (1) they often rely on a large amount of labelled training data to build the model, and (2) they cannot dynamically adapt the model with emerging or changing activity patterns over time. To directly address these challenges, we propose a Bayesian nonparametric model, i.e. Dirichlet process mixture of conditionally independent von Mises Fisher models, to enable both unsupervised and semi-supervised dynamic learning of human activities. The Bayesian nonparametric model can dynamically adapt itself to the evolving activity patterns without human intervention and the learning results can be used to alleviate the annotation effort. We evaluate our approach against real-world, third-party smart home datasets, and demonstrate significant improvements over the state-of-the-art techniques in both unsupervised and supervised settings.**

I. Introduction

The European Commission has predicted that by 2025, the United Kingdom alone will see a rise of 44% in people over 60 years of age. This motivates the development of new solutions to improve the quality of life and independence for elderly people. Ambient assisted living is one promising solution, which is enabled by sensor-based human activity recognition (HAR) – unobtrusively monitoring and inferring human activities from a collection of ambient and wearable sensors [1].

Due to its potential in healthcare, HAR has been extensively studied and numerous machine learning techniques have been applied therein. Most of them require a large number of training data well annotated with activity labels and assume a fixed model; *i.e*, once trained, an activity model will stay the same. However, this methodology and assumption does not reflect the complexity of real-world deployment. First of all, annotating sensor data with activity labels is known to be an intrusive, tedious, and time-consuming task. Secondly, users often change their behaviour over time; *e.g.*, starting a new type of exercise, or changing the cooking style due to health conditions.

Unsupervised techniques such as clustering can be useful to remedy the problematic situation. They can be employed, for example, to mine clusters from the raw data for annotation, which alleviates the problem of missing labels [2]. However, most existing clustering algorithms often require some pre-knowledge from the data: for example, k-means or any other mixture model, needs to pre-fix the number of activities in advance. Moreover, once fixed, the model is hard to adapt to the evolving human behaviours over time. As a result, the practicability and performance of existing solutions are compromised. HAR system also faces other challenges including the complexity of the sensor generated data: sensor-based HAR usually employs a large number of sensors in different modalities. The high-dimensionality further complicates the learning task.

To tackle this problem we propose a Bayesian nonparametric directional statistical model: specifically, a Dirichlet process mixture of conditionally-independent von Mises Fisher distributions (DP-MoCIvMFs). Our solution benefits from the properties of Bayesian nonparametrics (BNP) and directional statistics on high-dimensional data, and so can dynamically discover activity patterns and automatically infer the hidden activity cluster sizes at the same time. To the best of our knowledge this is the first work that employs Dirichlet process mixture models and von Mises Fisher models together in the HAR domain, and the proposed DP-MoCIvMFs model has never been studied before in existing literature. To be specific, we claim the following novelties and contributions:

- a novel statistical model based on Dirichlet process mixture and conditionally independent directional statistical models for HAR activity modelling;
- a method by which activity patterns and cluster size are learnt adaptively from the data under an unsupervised setting, whereas the performance is significantly better than the state-of-the-art algorithms;
- a partially-collapsed Gibbs sampler algorithm that can make efficient on-line inference over the model and its Bayesian hierarchical extension;
- an inference algorithm to train hierarchical mixture of conditionally independent vMFs model as an activity classifier; and the classification performance is on par

with the state-of-the-art machine learning algorithms.

The rest of the paper is organised as follows. Section II reviews the literature and compares and contrasts our approach with the existing work. Section III introduces the theory of von Mises-Fisher distributions and Bayesian mixture models. Section IV describes our proposed approach, which is then evaluated in Section V. We conclude our work in Section VI and point to some future work.

II. Related work

In this paper, we propose a novel generative model for unsupervised and semi-supervised learning that combines Bayesian nonparametrics and directional statistical models. In the following, we will survey the related literatures on activity recognition, BNP, directional statistical models and existing techniques in learning new types of activities and those devoted to reducing the labelling effort.

A. Activity Recognition

Activity recognition based on wearable and environmental sensing technologies has been extensively researched in the last decades and a few recent surveys have broadly reviewed the existing techniques [1], [3]–[5]. In general, sensor-based activity recognition techniques can be grouped into knowledge- and data-driven approach, and the data-driven approach can be further classified into supervised and unsupervised learning techniques. A knowledge-driven technique leverage expert knowledge ranging from the early attempt on a small scale of common sense knowledge [6] to a more advanced and formal approach on a large scale of knowledge base such as ontologies [7] and WordNet [8], [9], and apply reasoning engines to infer activities from sensor data. A data-driven technique apply the off-the-shelf machine learning and data mining techniques to automatically establish the correlation between sensor data and activity labels. Hidden Markov Models (HMM) and recent deep neural networks are the most popular techniques [1], [10].

B. Unsupervised Learning

Unsupervised learning automatically partitions and characterises sensor data into patterns that can be mapped to different activities without the need of annotated training data. Pattern mining and clustering are the two mostly used techniques that support unsupervised activity recognition. Gu et al. have applied emerging patterns to mine the sequential patterns for interleaved and concurrent activities [11]. Rashidi et al. propose a method to discover the activity patterns and then manually group them into activity definitions [12]. Based on the patterns, they create a boosted version of a HMM to represent the activities and their variations in order to recognise activities in real time. Similarly, Ye et al. have combined the sequential mining and clustering algorithms to discover representative sensor events for activities. Different from the work in [12], they have applied the generic ontologies to automatically map the discovered sensor sequential patterns to activity labels through a semantic matching process [13]. Yordanova et al. have also applied domain knowledge in rule-based systems to generate probabilistic models for activity recognition [14], [15].

From statistical modelling perspective, clustering problem, or unsupervised learning can be solved by mixture models. The main focus has traditionally been on Gaussian and multinomial models. Banerjee et al. proposed an EM based inference procedure for finite mixture of von Mises Fisher (vMF) [16]. Gopal and Yang derived the Bayesian learning inferences on a finite mixture of vMFs and some other extensions like Hierarchical mixtures of vMFs [17]. Taghia et al. did similar research on Bayesian learning on vMF mixture models via variational inference [18]. The infinite mixture extension of the vMFs mixture model is first studied by Bangert et al. [19] to cluster treatment beam in external radiation therapy; while later Roge et al. propose an alternative Collapsed Gibbs sampler to infer the same infinite mixture model [20]. Qin et al. [21] developed a reverse jump Markov Chain Monte Carlo algorithm to learn trans-dimensional model of von Mises Fisher models. The major difference between our model and theirs is the component density is assumed as multiple conditional independent vMFs that accommodate both sensor and time features rather than a singular vMF.

C. Semi-supervised Learning

One of the most common semi-supervised learning techniques is active learning, so called "query learning", a subfield of machine learning. It is motivated by the scenario when there is a large amount of unlabelled data but a limited and insufficient amount of labelled data. As the labelling process is tedious, time-consuming and expensive in real-world applications, active learning methods are employed to alleviate the labelling effort by selecting the most informative instances to be annotated [22].

Cheng et al. apply a density-weighted method that combines both uncertainty and density measure into an objective function to select the most representative instances for user annotation, which has been demonstrated to improve activity recognition accuracy with the minimal labelling effort [23]. Similarly, Hossain et al. combine the uncertainty measure and Silhouette coefficient to select the most informative instances as a way to discover new activities [24].

Alemdar et al. apply active learning strategies to select the most uncertain instances to be annotated; that is, the instances sit at the boundaries of different activity classes [25]. The annotated instances are used to iteratively update a HMM to infer daily activities in a home setting. Their experimental results have demonstrated that active learning strategies have improved recognition accuracies, compared to random selection. Fang et al. combine hierarchical mixture models of directional statistical models with active learning strategies to form an incremental and on-line learning framework for activity recognition [26]. Their solution demonstrates good emerging activity detection and model update accuracies. However, their mixture model's size is prefixed and model

update procedures are based on some specific form of EM algorithm rather than a formal statistical model.

III. Background

This section introduces the background on von Mises-Fisher distribution and its Bayesian finite mixture model, which forms the foundation of our proposed approach.

A. von Mises-Fisher Distribution

A von Mises-Fisher (vMF) distribution is a probability distribution with support on the unit hypersphere, whose density can be defined as

$$f(x|\mu,\kappa) = c_D(\kappa)e^{\kappa\mu^T x}, c_D(\kappa) = \frac{\kappa^{D/2-1}}{(2\pi)^{D/2}I_{D/2-1}(\kappa)}$$

where $x \in R^D$ is a $D-$dimensional vector with unit length, *i.e.* $\|x\|_2 = 1$, I_ν is the modified Bessel function of the first kind at order ν, $\mu \in R^D, \|\mu\|_2 = 1$ is the mean direction and $\kappa > 0$ is a concentration parameter indicating how concentrated the samples are generated against μ. When κ is large, the samples are closely aligned with μ, which tends to a point density; when κ is small, or close to zero, the model degenerates to the uniform distribution on the sphere [27]. Fig. 1 shows samples from three vMFs in a three dimensional setting. Note that as κ decreases, the distribution is more uniformly spread over the sphere. vMF is a good alternative to other commonly used distributions, like Gaussian, for high dimensional data. vMF based model has been successfully applied in high dimensional data analysis, like document topic modeling [16]–[18], gene expressions [16], [18], and fMRI time series [28] etc.

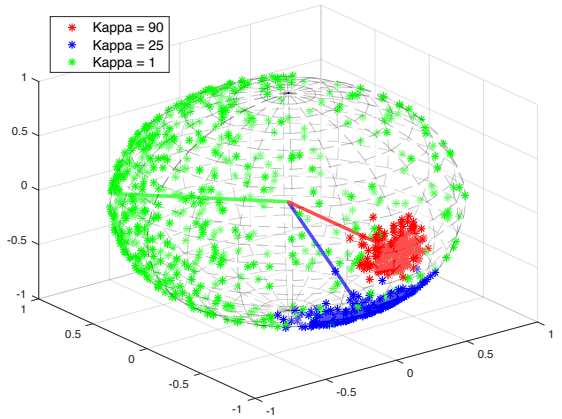


Fig. 1: von Mises-Fishers with different parameters; as κ decreases, the distribution is more widespread with respect to their mean vectors μ over the sphere.

B. Bayesian Mixture Model Specification

A finite mixture model assumes the data samples are independently generated by a fixed number of $K \geq 1$ components. The model implicitly assumes hidden categorical variables $z_i \in \{1,\ldots,K\}$, indicating which component originally generates x_i, $i = 1,\ldots,N$. The generative model of a finite mixture of vMFs can be written as:

$$z_i \sim Multi(\cdot|\pi)$$
$$x_i \sim vMF(\cdot|\mu_{z_i},\kappa_{z_i})$$

where the k-th component's vMF is defined by $\{\mu_k,\kappa_k\}$, and π is the mixture proportion.

A Bayesian extension of the finite mixture model can be defined by imposing additional prior distributions on the model parameters, *e.g.*

$$\pi \sim Dirichlet(\alpha)$$
$$z_i \sim Multi(\pi),\ i = 1\ldots N$$
$$\mu_k \sim vMF(m_0,C_0),\ ,k = 1\ldots K$$
$$\kappa_k \sim P_0^+,\ k = 1\ldots K$$
$$x_i \sim vMF(\mu_{z_i},\kappa_{z_i}),\ i = 1\ldots N$$

The model assumes the mixture proportion π is drawn from a symmetric Dirichlet distribution with parameter α; and each cluster's μ_k and κ_k are commonly drawn from a vMF prior with mean and concentration parameters $\{m_0,C_0\}$ and some prior distribution P_0^+ with a strictly positive support respectively. The probabilistic graphical model (PGM) representation of the Bayesian mixture of von Mises Fishers (B-movMF) is listed in Fig. 2. The Bayesian model holds various advantages over its likelihood-based counterpart, including parameter shrinkage, stability, inclusion of expert prior knowledge *etc.* [17]. Note that the above models require a pre-specified mixture size K, which is usually not feasible in real-world applications.

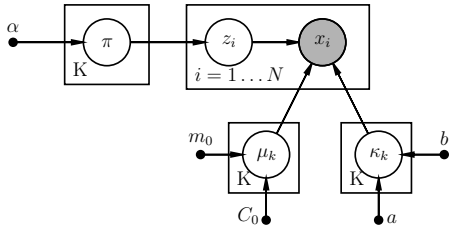


Fig. 2: The Bayesian finite mixture of vMFs in a standard probabilistic graphical model notation; where P_0^+ is assumed be identified by two parameters a, b.[1]

IV. Proposed Approach

We begin this section by describing the proposed statistical model, with the inference algorithm presented afterwards.

A. The Proposed Model

Sensor-based HAR, usually employing a large number of sensors, entails high-dimensional datasets. Inspired by the successful applications of vMFs on high-dimensional data in other domains, we propose to model sensor based activities

[1]The PGM diagrams in this paper are generated by DAFT http://daft-pgm.org/.

(after appropriate sensor feature extraction and transformation) by vMFs.

However, different classes of features are needed to differentiate the underlying activities, and the mixture of singular vMFs is not sufficiently flexible to capture all the characteristics. In particular, the time feature and other sensor features should ideally be treated separately as they naturally differ in many ways. Note that the time feature, upon the following cyclic transformation:

$$x^t = (\cos\theta, \sin\theta), \text{where } \theta = (h - h_0) \times (2\pi/24), \quad (1)$$

where $h - h_0$ is the elapsed time units between h and any fixed reference point h_0, is actually 2d vMF distributed whereas other sensor features, after appropriate feature extraction and transformation detailed in V-A, are of much higher dimensional directional vectors (depending on the number of deployed sensors). A more flexible approach is to treat them as multiple directional vectors on two spheres with different dimensions instead of a singular hypersphere vector.

In light of this, we propose the following conditionally independent (CI) component density. That is, conditioning on mixture identity z_i (or activity identity), we assume x_i is generated by independent vMFs, *i.e.*, decomposing x_i as the sensor features x_i^s and time feature x_i^t s.t. $x_i = [x_i^s, x_i^t]$, where x_i^s and x_i^t are unit vectors with d' and 2 dimensions respectively. The implied CI density of each cluster component becomes

$$f_{CI}(x_i|\cdot) = vMF(x_i^s; \mu_k^s, \kappa_k^s) vMF(x_i^t; \mu_k^t, \kappa_k^t). \quad (2)$$

Note that at the *mixture* level the different data components, assumed independent at *component* level, are *not* independent, due to the mixture model specification [29], implying the statistical correlations between the time and sensor features can be captured [26].

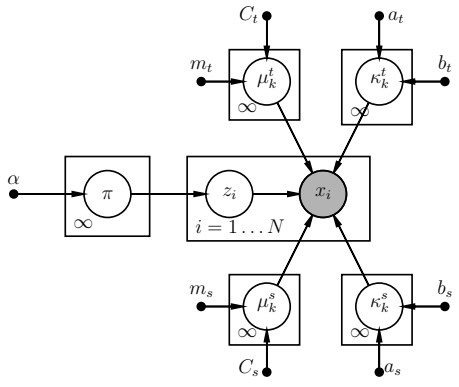


Fig. 3: The proposed DP-MCIvMF model in PGM representation.

1) Dirichlet Process Mixture of CI vMFs: So far, we have assumed a finite mixture of CIvMFs (moCIvMF), where the mixture size K is constant and has to be pre-specified manually. To resolve this problem, we propose the Bayesian non-parametric extension of the mixture model, *i.e.*, an infinite mixture of CIvMFs, where the mixture size is assumed infinitely large, or $K \to \infty$ [30]. It can be shown that the infinite mixture model induces the prior proportion π being generated from a Chinese Restaurant Process (CRP), whereas the cluster component parameters are drawn from their corresponding prior distribution (or equivalently the base measure of the Dirichlet process) [30]–[32]. The CRP representation based nonparametric model can be written as:

$$\begin{gathered}
\pi \sim CRP(\alpha), \\
z_i \sim Multi(\pi),\ i = 1 \dots N \\
\mu_k^s \sim vMF(m_s, C_s),\ \kappa_k^s \sim G(a_s, b_s),\ k = 1 \dots \infty \\
\mu_k^t \sim vMF(m_t, C_t),\ \kappa_k^t \sim G(a_t, b_t),\ k = 1 \dots \infty \\
x_i \sim f_{CI}(\mu_{z_i}^t, \kappa_{z_i}^t \mu_{z_i}^s, \kappa_{z_i}^s),\ i = 1 \dots N,
\end{gathered}$$

where the prior parameters are $\{\alpha, m_s, C_s, m_t, C_t, a_s, b_s, a_t, b_t\}$. A few explanations on the prior choices are given here. We use a Gamma distribution for κ because it has the same support as the concentration parameter (a positive real number) and it has also been shown that the likelihood function of κ closely resembles a Gamma form [18]. The vMF priors for μ are used because they have the matching support and are also conjugate to vMF likelihood. The model essentially is a Dirichlet Process Mixture model with a concentration parameter α and a base measure induced by the prior $p(\mu_k^s, \mu_k^t, \kappa_k^s, \kappa_k^t)$. This model is therefore denoted DP-MoCIvMFs hereafter. The equivalent PGM representation of the model is listed in Fig. 3. Note its differences with the Bayesian finite mixture model with singular vMF as component distribution, which is shown in Fig. 2.

B. Inference Algorithm

Based on the CRP mixture model representation, Gibbs sampling, a class of Markov Chain Monte Carlo method, can be used to make approximate Bayesian inference over the nonparametric infinite sized model [31], [32]. To improve the sampling efficiency, we employ the collapsing strategy and derive a partially collapsed Gibbs sampler [33]: the state of the chain to sample are $Z = \{z_i\}$ and $\{\kappa_k^s, \kappa_k^t\}$ with the mean directions $\{\mu_k^s, \mu_k^t\}$ integrated out analytically. Collapsing or integrating out analytically the mean direction parameters does not only saves the computation of sampling them (of high dimensional vectors, which can be expensive), but also improves the rate of convergence according to the Rao-Blackwell theorem.

As a general summary, the sampler iterate:

- For $i = 1, \dots, N$, iteratively sample each z_i conditioning on the rest of the chain state, i.e. $Z_{/i}$, $\{\kappa_k^s, \kappa_k^t\}$ and the observed data X;
- Sample $\{\kappa_k^s, \kappa_k^t\}$ conditioning on Z and X;
- Update prior-parameters if necessary;

The required conditional distributions for the sampling steps are:

$$p(z_i = k|\cdot) \propto n_{k,-i} \cdot c_2(\kappa_k^t) \frac{c_2(\|m_t C_t + \kappa_k^t \sum_{j \in \mathcal{Z}^k_{/i}} x_j^t\|)}{c_2(\|m_t C_t + \kappa_k^t \sum_{j \in \mathcal{Z}^k} x_i^t\|)}$$
$$\cdot c_{d'}(\kappa_k^s) \frac{c_{d'}(\|m_s C_s + \kappa_k^s \sum_{j \in \mathcal{Z}^k_{/i}} x_j^s\|)}{c_{d'}(\|m_s C_s + \kappa_k^s \sum_{j \in \mathcal{Z}^k} x_j^s\|)},\ k = 1 \dots K' \quad (3)$$

$$p(z_i = k|\cdot) \propto \alpha \cdot \frac{1}{M} \sum_{m=1}^{M} f_{CI}(x_i|\theta^s_{(m)}, \theta^t_{(m)}),\ k = K' + 1 \quad (4)$$

$$p(\kappa_k^s|\cdot) \propto \frac{c_{d'}(\kappa_k^s)^{n_k}}{c_{d'}(\|\kappa_k^s \sum_{j \in \mathcal{Z}^k} x_j^s + C_s m_s\|)} G(a_s, b_s) \quad (5)$$

$$p(\kappa_k^t|\cdot) \propto \frac{c_2(\kappa_k^t)^{n_k}}{c_2(\|\kappa_k^t \sum_{j \in \mathcal{Z}^k} x_j^t + C_t m_t\|)} G(a_t, b_t); \quad (6)$$

where K' denote the size of the occupied clusters at the current iteration; $\mathcal{Z}^k = \{j : z_j = k\}$ and $\mathcal{Z}^k_{/i} = \{j \neq i : z_j = k\}$; $n_k = |\mathcal{Z}^k|$ and $n_{k,-i} = |\mathcal{Z}^k_{/i}|$, i.e. the size of the observations in the kth cluster. And the derivation of some key results are given in the supplemental file. Some explanations over the sampling steps are given below.

Sampling z_i: The sampling step for $z_i \in \{1, \dots, K'+1\}$ updates its cluster membership according to its conditional distribution. When $z_i = K' + 1$, *i.e.* eq. (4), it denotes the probability of the observation occupying a new cluster (or new table in the CRP metaphor). Based on the CRP prior and the convoluted CIvMF base measure, the probability is proportional to

$$\alpha \cdot \int \int f_{CI}(x_i|\theta^s, \theta^t) p(\theta^s, \theta^t) d\theta^s d\theta^t,$$

where $\theta^s = \{\mu^s, \kappa^s\}$, $\theta^t = \{\mu^t, \kappa^t\}$. As this integral has no closed form solution, we resort to Monte Carlo (MC) approximation, where the MC sample size is M, and $\theta^s_{(m)}$ and $\theta^t_{(m)}$ denote the m-th sample generated from the prior distribution. To sample from the vMF prior, we have used the Wood method [34]. We find $M = 1$ works well in most cases, while a larger M leads to better converging rate in general (see the result part for some analysis on the effect of M). Note that $\{\theta^s, \theta^t\}$ can be pre-sampled and cached for reuse in the Gibbs iterations, which greatly reduces the computation effort.

Sampling κ: The conditional distribution on κ, i.e. eq. (5) and (6) are not of standard forms but one-dimensional distributions that can be evaluated up to some unknown constants; we therefore use slice sampler to sample them with initial starting values set as the current state [35]. Note that a slice sampler is efficient for univariate distribution sampling and it only needs to evaluate the distribution proportional to some constant.

1) On-line Inference: An important advantage of the proposed method is its capability to deal with on-line inference: *i.e.* incorporation of new sensor data into the learning process as time progresses. As the time feature is explicitly incorporated into the mixture component, the temporal order of the recorded activity data no long matters as opposed to other time series models, such as HMMs. Therefore, new data samples can be included into the sampling procedure simply as the last arriving customers of the CRP metaphor, which is in line with the exchangeability assumption of the DP mixture model [36]. Computationally speaking, as the sampler considers each data sample marginally (eq. (3) (4)), which implies new observations' cluster memberships z_is can be sampled at the end of the hidden membership sampling step conditioning on the status of the sitting arrangements of the existing customers, and the sampling procedure can resume as normal but with expanded data size N at the following iterations.

2) Prior specification and hierarchical Bayesian model: The prior parameters $\{\alpha, m_s, C_s, m_t, C_t, a_s, b_s, a_t, b_t\}$ can either be elicited from expert knowledge or learnt from data. To minimise human input, we impose a hierarchical Bayesian model to learn the prior parameters [37]. In particular, the following hyper-priors are used:

$$\alpha^{-1} \sim G(1, 1),$$
$$m_s \sim vMF(\bar{m}_s, 0.01),\ m_t \sim vMF(\bar{m}_t, 0.01),$$
$$b_t \sim G(0.01, 0.01),\ b_s \sim G(0.01, 0.01),$$

where $\bar{m}_s$, $\bar{m}_t$ are the maximum likelihood (ML) estimators of the mean directions from the whole data set; while the others are fixed as constants $a_t = a_s = 1$ (a standard practice for noninformative Gamma prior), $C_s = C_t = 0.1$ (noninformative priors on the mean directions). The prior parameter update procedures can be derived based on conjugacy, which are detailed in a supplemental file that is made publicly available along with the implementation code [3].

C. (Semi-)supervised Learning

To use the DP-MoCIvMFs as a classifier, we only need to slightly modify the algorithm by treating the testing data's labels as missing value. In an overview, a DP-MoCIvMFs can be learnt on each labelled dataset by running the Gibbs sampler in parallel; as a result, a DP-MoCIvMFs for the whole dataset can be formed; then the unlabelled data (test data) can be classified by running the Gibbs sampler to update their labels (and only their labels). The detail is listed below in Algorithm 1. The algorithm essentially creates a (flattened) hierarchical mixture of CIvMFs model for classification, where each activity is a mixture model. The novelty here is that each mixture's size is learnt from the data rather than pre-fixed.

V. Experiment and Evaluation

We now present an assessment of the performance of our proposed solution. Both synthetic and real-world data are used: the synthetic data analysis is mainly used to demonstrate the correctness of the proposed inference algorithm, whereas the real-world analysis tries to access the algorithm's realistic applicability in HAR.

Algorithm 1 Semi-supervised learning of DP-MoCIvMFs

Input labelled training data $\{D_c\}_1^C$ and testing data X_{test}

1: **for** each labelled dataset D_c, $c = 1 \dots C$ **do**
2: Run DP-MoCIvMFs Gibbs sampler on D_c
3: Create a map from the K_c clusters to class c
4: **end for**
5: Form a DP-MoCIvMFs on $\{D_c\}_1^C$ with $\sum_{c=1}^{C} K_c$ clusters
6: Run the Gibbs sampler on X_{test}
7: Map Z_{test} to their corresponding classes

A. Datasets and Sensor Data Pre-processing

We perform the evaluation on two real-world smart home datasets. The first dataset (House A) is collected by the University of Amsterdam from a single-resident house instrumented with a wireless sensor network [38]. The dataset has 16 dimensions (sensors) and 7 activities. The second dataset is collected from a testbed at Washington State University [2]. This dataset has 32 sensors and 9 different activities.

We segment sensor events into time slots of a fixed interval. For each time slot, we extract features from the sensor data and associated timestamps. A sensor feature vector is represented as $x_s = [x_1, x_2, ..., x_S]$, where S is the number of sensors being installed, and each x_i $(1 \leq i \leq S)$ (possibly a vector by itself depending on the sensor type and feature extraction technique) is the extracted feature of the ith sensor. If a sensor is binary (*e.g.*, an RFID, switch sensor, or passive infra-red motion sensor) x_i is the frequency of this sensor being activated over the interval: that is, n_i/n, where n_i is the number of times the ith sensor being activated and n is the total number of sensor events reported in this time slot. For the timestamps, instead of treating them as real-valued scalar feature, we apply the transformation listed in (1).

B. Metrics and Baselines

The proposed model, DP-MoCIvMFs, is implemented in Matlab, which is made publicly available[3]. We have chosen a range of Bayesian/maximum likelihood based and parametric/nonparametric models as baselines to give a comprehensive comparison. For non-parametric models or finite mixture models, like K-means, the K is set as the true cluster size. The details of the baselines are:

- DP-MovMF: Dirichlet Process Mixture of vMFs [19], implemented in Matlab [3];
- DP-MoG: Dirichlet Process Mixture of Gaussians, the DP base measure is the regular conjugate Normal-Inverse Wishart distribution [4];
- DP-MoCIG: Dirichlet Process Mixture of CI Gaussians, the algorithm is implemented based on the existing DP-MoG program [4];
- K-means: the standard k-means with Euclidean distance as distance measure [5];
- MovMF (EM): Mixture of vMFs estimated by expectation-maximization (EM) algorithm[6] [16];
- MoG (EM): Mixture of Multivariate Gaussians estimated by EM algorithm [4].

To evaluate the unsupervised learning performance, we use five standard measures for clustering algorithms [39]: mutual information (MI), normalised mutual information (NMI), Rand Index (RI), adjusted Rand Index (ARI), and Purity.

To demonstrate the classification performance of the proposed model, we use two criteria that are commonly used in existing literature [38] [40] to access the activity recognition accuracy, namely time-sliced wise accuracy (A_t) and class wise accuracy (A_c); that is,

$$A_t = \frac{N_a}{N}, \quad A_c = \frac{1}{K}\sum_{a=1}^{K} A_a$$

where N_a is the number of times that an activity is correctly classified, and N is total time slice count; A_a is the classification sensitivity rate with respect to activity a, i.e. $A_a = \frac{TP_a}{TP_a + FN_a}$, where TP_a and FP_a denote the true positive and false positive counts of the classifier with respect to activity a. Therefore, A_c measures the averaged by class accuracy among all class labels. We also report F-score to help compare the performance on both precision and sensitivity, where

$$F\text{-}score_a = \frac{2 \times TP_a}{2 \times TP_a + FP_a + FN_a}.$$

C. Synthetic Data Analysis

We demonstrate the effectiveness of the derived sampling algorithm on two synthetically generated datasets, denoted $\mathcal{D}_1$ and $\mathcal{D}_2$. In particular, we want to examine whether the sampler can discover the hidden clusters and infer the correct cluster size at the same time. For $\mathcal{D}_1$, three datasets of mixture of $K = 4$ conditionally independent vMFs are generated. The dimensions of the two conditionally independent vMFs are 5 and 2 respectively. The concentration parameters of the three data sets are set as $\{5, 10, 25\}$ to simulate high noise, median noise and low noise scenarios. Each dataset has $N = 100$ data samples and each cluster has equal size, i.e. 25 for each cluster. The datasets for the three noise variates are collectively denoted as $\mathcal{D}_1$. To further challenge the algorithm and make the generated data more similar to the real world HAR data, we generate another suite of datasets, collectively denoted as $\mathcal{D}_2$, with cluster size $K = 10$, dimension $D = 20$, and dataset size $N = 300$. Furthermore, the cluster sizes are in-balanced among the 10 clusters to mimic the in-balanced distribution of human activities, where the size ratio between the largest and smallest cluster varies around 8. The concentration parameters are varied again among the three values to denote the high, median and low noise cases.

Table I lists the the average of 10 independent runs on $\mathcal{D}_1$ where $K = 4$ and $D = 7$. The initialised value of K for

[2]http://ailab.wsu.edu/casas/datasets/
[3]https://leo.host.cs.st-andrews.ac.uk
[4]http://prml.github.io/
[5]https://uk.mathworks.com/products/statistics.html

[6]The initialisation step is modified as random assignment to avoid the converging problem of the original implementation (especially for high dimensional data).

TABLE I: Comparison of DP-MoCIvMF on different synthetic datasets $\mathcal{D}_1$ with various noise levels. The correct cluster size is $K = 4$; dimension $D = 7$. The paired t-test results against DP-MoCIvMF, M=30 are denoted by a $*$ for significance at 5% , and † for 1% level.

Dataset		High Noise		Med Noise		Low Noise	
Method/Metrics		NMI	K	NMI	K	NMI	K
DP-MoCIvMF	M=1	.735	3.04	.849	**4.08**	.981	**4.0**
	M=30	**.751**	**3.8**	**.87**	4.46	**.983**	**4.0**
DP-MoG		.692†	**3.8**	.761†	3.14	.636†	2.14
DP-MoCIG		.719†	3.54	.758†	2.44	.786†	2.44
K-means (K= 4)		.732†	NA	.866	NA	.941†	NA

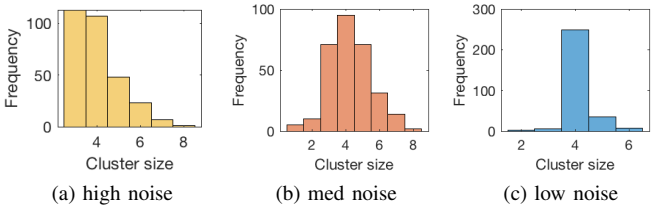
TABLE II: Comparison of DP-MoCIvMF on different synthetic datasets $\mathcal{D}_2$ with various noise levels. The correct cluster size is $K = 10$; dimension $D = 20$; The paired t-test results against DP-MoCIvMF, M=30 are denoted by a $*$ for significance at 5% , and † for 1% level.

Dataset		High Noise		Med Noise		Low Noise	
Method/Metrics		NMI	K	NMI	K	NMI	K
DP-MoCIvMF	M=1	.636	8.4	0.806	**8.9**	.952*	9.1
	M=30	**.657**	**11.1**	**0.807**	**11.1**	**.984**	**9.9**
DP-MoG		.509†	5.8	.663†	7.3	855†	6.3
DP-MoCIG		.569†	4.6	.669†	5.5	.808†	5.4
K-means (K= 4)		.641	NA	.768†	NA	.911†	NA

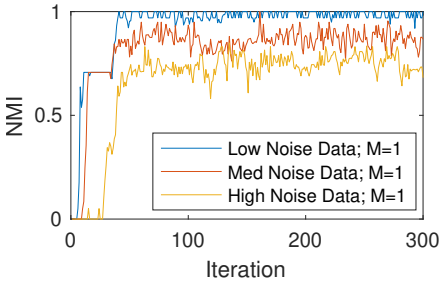
the sampler is set as 1; *i.e.* all the data are from one cluster. Each chain runs 200 iterations. The reported Ks are the mean of the 10 modes of the ten chains and NMIs are the average of the ten runs with the first half of each sample discarded as burn-in. Note that NMI and the inferred K value together gives a complete assessment of the clustering performance. It can be seen that the correct cluster size is recovered by the algorithm for both median and low noise cases while the algorithm's performance on high noise data deteriorates slightly. The Monte Carlo sample size M does not affect the performance much, as the deviance is not significant. The results of some other models/algorithms on the same datasets are also listed for reference.

To better understand the distribution or uncertainty of K and the effect of the Monte Carlo sample size M, results from some individual runs are plotted in Fig 4. The histograms of the inferred cluster size with the sampler with $M = 1$ are shown in the top row (the results with $M = 30$ is very similar). It is evident that the uncertainty grows as the data becomes noisier. But the correct size 4 is always within the highest credible interval, which shows the proposed algorithm can successfully infer the cluster size from the data. The lower two figures show the NMI traces of two samplers with $M = 1$ and 30 respectively. In general, the sampler with $M = 1$ converges slower (stuck at some local maximum initially) but eventually mix well, and there is no significant difference between the converged results of the two settings.

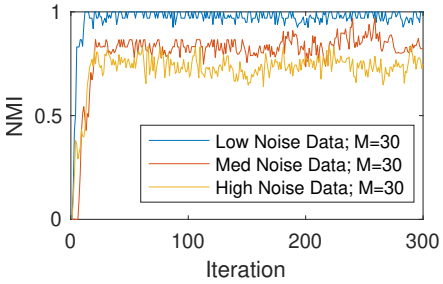
Table II lists the results on $\mathcal{D}_2$ where $K = 10$ and $D = 20$. The overall results show a very similar pattern as the $\mathcal{D}_1$'s, although the Monte Carlo sample size M seems affect the performance a bit more for this more complicated dataset, as the deviance is slightly greater. Also the correct cluster size can be inferred from the data by the algorithm for the low-noise case while the algorithm's performance on median-noise data is slightly off the target although the difference is minor (within 1 on average).



(a) high noise (b) med noise (c) low noise



(d) NMI traces under the integration Monte Carlo sample size $M = 1$



(e) NMI traces under the integration Monte Carlo sample size $M = 30$

Fig. 4: Evaluations on synthetic data set $\mathcal{D}_1$; the top three figures show the inferred cluster size under the three data sets, and the Monte Carlo sample size $M = 1$ is used; the lower two figures show the NMI against the running iterations on both $M = 1$ (left) and $M = 30$ (right) settings.

D. Unsupervised Learning on Real World Sensor Data

In this section, we apply the proposed method on real world HAR data to investigate the research question whether the solution can cope with the complexity. The results are shown in Table III and IV respectively on House A and Washington datasets. Each result is averaged over 10 different starting values for the algorithms, where the initial K is randomly set between 1 to 10. Bold face numbers indicate the best performing method with respect to the corresponding

TABLE III: Experiment results on House A data. The paired t-test results are denoted by $*$ for significance at 5% , and † for 1% level.

Method/Metric	NMI	MI	Rand Index	Adjusted RI	Purity
DP-MoCIvMFs	**.690 (.032)**	2.159 (.118)	**.875 (.008)**	**.493 (.039)**	.892 (.034)
DP-MoCIvMFs on-line	.690 (.011)	**2.185 (.0437)**	.872 (.004)	.471 (.025)	**.901 (.015)**
DP-MovMF	.619 (.047)†	2.089 (.252)†	.835 (.016)†	.280 (.041)†	.857 (.076)*
DP-MoG	.530 (.060)†	1.144 (.151)†	.739 (.058)†	.3783 (.092)†	.580 (.037)†
DP-MoCIG	.566 (.049)†	1.315 (.146)†	.804 (.042)†	.475 (.081)	.616 (.035)†
K-means	.519 (.039)†	1.354 (.114)†	.791 (.018)†	.304 (.045)†	.619(.053)†
MovMF	.474 (.045)†	1.190 (.127)†	.756 (.022)†	.251 (.046)†	.592 (.041)†
MoG	.489 (.062)†	1.061 (.192)†	.713 (.086)†	.344 (.117)†	.536 (.082)†

TABLE IV: Experiment results on Washington data. The paired t-test results are denoted by $*$ for significance at 5% , and † for 1% level.

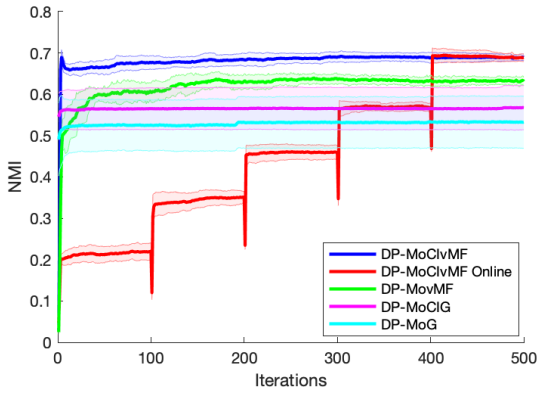
Method/Metric	NMI	MI	Rand Index	Adjusted RI	Purity
DP-MoCIvMFs	**.755 (.031)**	2.357 (.120)	**.899 (.009)**	**.586 (.044)**	.908 (.032)
DP-MoCIvMFs on-line	.752 (.022)	**2.427 (.023)**	.893 (.001)	.550 (.027)	**.929 (.011)**
DP-MovMF	.641 (.010)†	2.354 (.049)	.844 (.001)†	.232 (.005)†	.905 (.016)
DP-MoG	.512 (.055)†	1.116 (.164)†	.720 (.043)†	.323 (.068)†	.495 (.050)†
DP-MoCIG	.642 (.058)†	1.626 (.155)†	.830 (.032)†	.487 (.093)*	.664 (.067)†
K-means	.564 (.026)†	1.629 (.074)†	.842 (.008)†	.390 (.031)†	.723 (.036)†
MovMF (EM)	.461 (.034)†	1.300 (.098)†	.787 (.016)†	.232 (.035)†	.630 (.037)†
MoG (EM)	.515 (.028)†	1.356 (.087)†	.828 (.019)†	.448 (.064)†	.617 (.028)†

evaluation metric. Statistical significance tests results against DP-MoCIvMF are denoted by a $*$ for significance at 5% level and † for 1%. The reported values are the means and standard deviations of the ten runs with the the first half of the chains discarded as burn in.

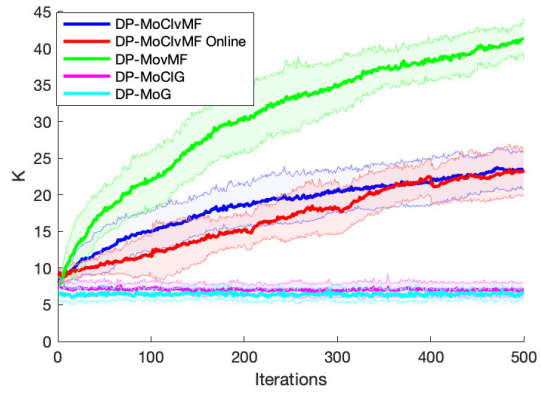
To assess the algorithm's performance on on-line inference, we also simulate the scenario by segmenting the whole dataset randomly into equal subsets and feed the algorithm incrementally at some sampling frequency (every 100 iterations). In reality, it is similar to adding operational observations at some fixed frequency *e.g.* every 60 mins.

Based on the results, it is evident that the proposed model, either on-line or off-line, perform the best among the listed clustering methods across the five metrics. It is interesting to note that the vMF based methods outperform their Gaussian equivalences, which supports our claim vMFs are suitable for sensor based human activity modelling. The difference between the Gaussian models and CIvMF models is greater in Washington dataset where the data dimension is larger and Gaussian models struggles to fit (whose parameter size grows in $(O(D^2))$ comparing to $O(D)$ for vMF). Comparing with other vMF based methods, DP-MoCIvMFs also achieves better results, which demonstrates the effectiveness of the DP-mixture and CI assumption. The purity measures how pure each formed cluster with respect to the true label. The good performance of DP-MoCIvMFs on this metric indicates the algorithm's potential in alleviating data annotation effort. The on-line inference of DP-MoCIvMFs achieves comparable results as its off-line counterpart, where the differences are insignificant; this shows the proposed solution's capability in handling the on-line learning scenario, a desirable property for long term deployed HAR systems.

To better understand how the algorithm evolves, the chain traces are plotted in Fig. 5 for the House A data set (the Washington data result is similar therefore omitted). The bold coloured lines are the mean of the ten runs where the shaded intervals are the $+/-$ standard deviations at each iteration. Note that the standard deviations plotted here are against the means at each iteration rather than the overall mean across the runs and iterations. Note the fluctuation of the on-line algorithm where new data is fed at every 100 iterations: the algorithm can always quickly learn the new data and achieves comparable result as the off-line algorithm at the end. Comparing vMF models against their Gaussian counterparts, the Gaussian models struggle to allocate or expand any new clusters at a very early stage; and their NMI values stuck at some local maximum as well, which probably can be attributed to the curse of dimension. The DP-MoCIvMF outperforms the DP-MovMF methods in NMI as they converge to better clustering configurations sooner; while the inferred cluster size is significantly smaller than their vMF counterpart. This is probably because the single vMF model is not flexible enough to accommodate the various cluster patterns showcased in the data so it has to spawn new clusters to compensate constantly, which leads to over-sized cluster size, although the purity performance is still not as good as the proposed method despite the extra allocated clusters. This also demonstrates the effectiveness of the proposed CI assumption.



(a) NMI traces



(b) K traces

Fig. 5: NMI traces and inferred K values against iteration on House A Dataset; the bold colored lines are the mean values at each iteration while the shaded area are $+/-$ standard deviation against the iteration means.

E. Semi-supervised Learning

In this section, we evaluate how the proposed solution works as an activity classifier under the supervised learning setting (as detailed in IV-C). We evaluate the method again on the two real world datasets. The results are reported in Table V and VI respectively. We compare the proposed solution against a wide range of generative and discriminative classification algorithms. In particular, we compare it with three other statistical model based classifiers, namely Mixture of Gaussians (MoG) and Mixture of von Mises Fisher (MovMFs) both of which are estimated by maximum likelihood method, while DP-MovMF and DP-MoCIvMFs are learnt based on the proposed Gibbs sampler based algorithm. In addition, we also list the results of a few widely used discriminative classifiers: Neural Networks (NNet), Support Vector Machine (SVM), K-Nearest Neighbour (KNN), and Random Forest. All the listed discriminative classifiers are implemented in Matlab's Statistic and Machine Learning toolbox. A five-fold cross validation is used for this comparison.

According to the results, the overall stronger performance of the discriminative classifiers over generative ones echos existing research findings [41]. Nevertheless, it is evident that the DP-MoCIvMFs is a strong candidate for activity classification. Its performance is the best among all generative models and comparable or better than most of the discriminative classifiers.

TABLE V: Comparing classification accuracy on House A data.

Method	By Time Slice	By Class	F-score
NNet	.912 (.031)	.874 (.051)	.874 (.043)
SVM	.92 (.019)	.847 (.013)	.851 (.02)
KNN	.912 (.03)	.88 (.051)	.884 (.054)
Random Forest	.926 (.03)	.899 (.06)	.895 (.036)
MoG	.887 (.044)	.875 (.058)	.857 (.048)
MovMF	.796 (.024)	.823 (.04)	.793 (.031)
DP-MovMF	.895 (.022)	.853 (.047)	.85 (.059)
DP-MoCIvMFs	**.932 (.03)**	**.901 (.052)**	**.91 (.054)**

TABLE VI: Comparing classification accuracy on Washington data.

Method	By Time Slice	By Class	F-score
NNet	.905 (.023)	.8 (.051)	.787 (.038)
SVM	.927 (.011)	.803 (.022)	.801 (.014)
KNN	.922 (.021)	.814 (.061)	.819 (.059)
Random Forest	.926 (.015)	.822 (.035)	.815 (.029)
MoG	.65 (.05)	.678 (.065)	.618 (.073)
MovMF	.754 (.036)	.739 (.049)	.675 (.039)
DP-MovMF	.884 (.028)	.812 (.053)	.783 (.039))
DP-MoCIvMFs	**.939 (.019)**	**.852 (.059)**	**.836 (.048)**

VI. CONCLUSION AND FUTURE WORK

This paper proposes a novel generative statistical model for human activity mining. It supports unsupervised and semi-supervised learning for human activity recognition, without the need for any pre-knowledge on the number or the profiles of potential activities. It can not only reduce the burden of labelling sensor data, but also support inference over dynamically evolving activities. We have evaluated the proposed approach on synthesised and real-world smart home datasets and compared with a wide range of alternative approaches. The evaluation results have demonstrated the proposed solution's capabilities in both unsupervisedly clustering HAR data without fixing the cluster size and supervisedly learning the label correctly.

In the future, we will assess the proposed model's performance on more datasets. And apply the algorithm in real-world on-line experiment. At the meantime, to further improve the derived Gibbs sampler's performance, the effect of choice of priors, other alternative choices of hyper-prior specification together with the prior parameter update will be carefully investigated.

ACKNOWLEDGEMENT

This work has been partially supported by the UK EPSRC under grant number EP/N007565/1, "Science of Sensor Systems Software".

REFERENCES

[1] J. Ye, S. Dobson, and S. McKeever, "Situation identification techniques in pervasive computing: a review," *Pervasive and mobile computing*, vol. 8, pp. 36–66, Feb. 2012.

[2] J. Ye, G. Stevenson, and S. Dobson, "Usmart," *ACM Transactions on Interactive Intelligent Systems*, vol. 4, no. 4, pp. 1–27, Nov 2014. [Online]. Available: http://dx.doi.org/10.1145/2662870

[3] J. Aggarwal and M. Ryoo, "Human activity analysis: A review," *ACM Comput. Surv.*, vol. 43, no. 3, pp. 16:1–16:43, Apr. 2011. [Online]. Available: http://doi.acm.org/10.1145/1922649.1922653

[4] L. Chen, J. Hoey, C. Nugent, D. Cook, and Z. Yu, "Sensor-based activity recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 42, no. 6, pp. 790–808, 2012.

[5] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Communications Surveys Tutorials*, vol. 15, no. 3, pp. 1192–1209, Third 2013.

[6] S. W. Loke, "Representing and reasoning with situations for context-aware pervasive computing: a logic programming perspective," *The Knowledge Engineering Review*, vol. 19, no. 03, pp. 213–233, 2004.

[7] J. Ye, S. Dasiopoulou, G. Stevenson, G. Meditskos, E. Kontopoulos, I. Kompatsiaris, and S. Dobson, "Semantic web technologies in pervasive computing: A survey and research roadmap," *Pervasive and Mobile Computing*, vol. 23, pp. 1 – 25, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574119214001989

[8] W. Kleiminger, F. Mattern, and S. Santini, "Predicting household occupancy for smart heating control: A comparative performance analysis of state-of-the-art approaches," *Energy and Buildings*, vol. 85, pp. 493 – 505, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S037877881400783X

[9] E. Tapia, T. Choudhury, and M. Philipose, "Building reliable activity models using hierarchical shrinkage and mined ontology," in *Pervasive '06*. Springer Berlin Heidelberg, 2006, pp. 17–32.

[10] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognition Letters*, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016786551830045X

[11] T. Gu, Z. Wu, X. Tao, H. K. Pung, and J. Lu, "epsicar: An emerging patterns based approach to sequential, interleaved and concurrent activity recognition," in *2009 IEEE International Conference on Pervasive Computing and Communications*, March 2009, pp. 1–9.

[12] P. Rashidi, D. J. Cook, L. B. Holder, and M. Schmitter-Edgecombe, "Discovering activities to recognize and track in a smart environment," *IEEE Trans. on Knowl. and Data Eng.*, vol. 23, no. 4, pp. 527–539, Apr. 2011.

[13] J. Ye, G. Stevenson, and S. Dobson, "Usmart: An unsupervised semantic mining activity recognition technique," *ACM Trans. Interact. Intell. Syst.*, vol. 4, no. 4, pp. 16:1–16:27, Nov. 2014. [Online]. Available: http://doi.acm.org/10.1145/2662870

[14] K. Yordanova and T. Kirste, "A process for systematic development of symbolic models for activity recognition," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, pp. 20:1–20:35, Dec. 2015. [Online]. Available: http://doi.acm.org/10.1145/2806893

[15] F. Kruger, M. Nyolt, K. Yordanova, A. Hein, and T. Kirste, "Computational state space models for activity and intention recognition. a feasibility study," *PLOS ONE*, vol. 9, pp. 1–24, 11 2014. [Online]. Available: https://doi.org/10.1371/journal.pone.0109381

[16] A. Banerjee, I. S. Dhillon, J. Ghosh, and S. Sra, "Clustering on the unit hypersphere using von mises-fisher distributions," *J. Mach. Learn. Res.*, vol. 6, pp. 1345–1382, Dec. 2005. [Online]. Available: http://dl.acm.org/citation.cfm?id=1046920.1088718

[17] S. Gopal and Y. Yang, "Von mises-fisher clustering models," in *International Conference on Machine Learning*, 2014, pp. 154–162.

[18] J. Taghia, Z. Ma, and A. Leijon, "Bayesian estimation of the von-mises fisher mixture model with variational inference," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 9, pp. 1701–1715, Sept 2014.

[19] M. Bangert, P. Hennig, and U. Oelfke, "Using an infinite von mises-fisher mixture model to cluster treatment beam directions in external radiation therapy," in *Machine Learning and Applications (ICMLA), 2010 Ninth International Conference on*. IEEE, 2010, pp. 746–751.

[20] E. Rogers, J. D. Kelleher, and R. J. Ross, "Using topic modelling algorithms for hierarchical activity discovery," in *Ambient Intelligence-Software and Applications–7th International Symposium on Ambient Intelligence (ISAmI 2016)*. Springer, 2016, pp. 41–48.

[21] X. Qin, P. Cunningham, and M. Salter-Townshend, "Online trans-dimensional von mises-fisher mixture models for user profiles," *Journal of Machine Learning Research*, vol. 17, no. 200, pp. 1–51, 2016. [Online]. Available: http://jmlr.org/papers/v17/15-454.html

[22] B. Settles, "Active learning literature survey," University of Wisconsin–Madison, Computer Sciences Technical Report 1648, 2009.

[23] H.-T. Cheng, F.-T. Sun, M. Griss, P. Davis, J. Li, and D. You, "Nuactiv: Recognizing unseen new activities using semantic attribute-based learning," in *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*. ACM, 2013, pp. 361–374.

[24] H. S. Hossain, N. Roy, and M. A. A. H. Khan, "Active learning enabled activity recognition," in *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2016, pp. 1–9.

[25] H. Alemdar, T. L. van Kasteren, and C. Ersoy, "Using active learning to allow activity recognition on a large scale," in *International Joint Conference on Ambient Intelligence*. Springer, 2011, pp. 105–114.

[26] L. Fang, J. Ye, and S. Dobson, "Discovery and recognition of emerging human activities using a hierarchical mixture of directional statistical models," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2019.

[27] K. V. Mardia and P. E. Jupp, *Directional Statistics*. Wiley, 2008.

[28] R. E. Røge, K. H. Madsen, M. N. Schmidt, and M. Mørup, "Infinite von mises-fisher mixture modeling of whole brain fmri data," *Neural Comput.*, vol. 29, no. 10, pp. 2712–2741, Oct. 2017. [Online]. Available: https://doi.org/10.1162/neco_a_01000

[29] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[30] C. E. Rasmussen, "The infinite gaussian mixture model," in *In Advances in Neural Information Processing Systems 12*. MIT Press, 2000, pp. 554–560.

[31] M. D. Escobar and M. West, "Bayesian density estimation and inference using mixtures," *Journal of the American Statistical Association*, vol. 90, no. 430, pp. 577–588, 1995. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/01621459.1995.10476550

[32] R. M. Neal, "Markov chain sampling methods for dirichlet process mixture models," *Journal of Computational and Graphical Statistics*, vol. 9, no. 2, pp. 249–265, 2000. [Online]. Available: http://www.jstor.org/stable/1390653

[33] J. S. Liu, "The collapsed gibbs sampler in bayesian computations with applications to a gene regulation problem," *Journal of the American Statistical Association*, vol. 89, no. 427, pp. 958–966, 1994.

[34] A. T. Wood, "Simulation of the von mises fisher distribution," *Communications in statistics-simulation and computation*, vol. 23, no. 1, pp. 157–164, 1994.

[35] R. M. Neal, "Slice sampling," *Ann. Statist.*, vol. 31, no. 3, pp. 705–767, 06 2003. [Online]. Available: https://doi.org/10.1214/aos/1056562461

[36] D. J. Aldous, "Exchangeability and related topics," in *École d'été de probabilités de Saint-Flour, XIII—1983*, ser. Lecture Notes in Math. Berlin: Springer, 1985, vol. 1117, pp. 1–198. [Online]. Available: http://www.springerlink.com/content/c31v17440871210x/fulltext.pdf

[37] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*, 2nd ed. Chapman and Hall/CRC, 2004.

[38] T. van Kasteren, A. Noulas, G. Englebienne, and B. Kröse, "Accurate activity recognition in a home setting," in *UbiComp '08: Proceedings of the 10th International Conference on Ubiquitous Computing*. Seoul, Korea: ACM, Sep. 2008, pp. 1–9.

[39] D. M. Christopher, R. Prabhakar, and S. Hinrich, "Introduction to information retrieval," *An Introduction To Information Retrieval*, vol. 151, no. 177, p. 5, 2008.

[40] N. C. Krishnan and D. J. Cook, "Activity recognition on streaming sensor data," *Pervasive and Mobile Computing*, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574119212000776

[41] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," in *Advances in neural information processing systems*, 2002, pp. 841–848.