Discovery and Recognition of Emerging Human Activities Using a Hierarchical Mixture of Directional Statistical Models

Lei Fang, *Member, IEEE,* Juan Ye, and Simon Dobson, *Senior Member, IEEE*

Abstract—Human activity recognition plays a significant role in enabling pervasive applications as it abstracts low-level noisy sensor data into high-level human activities, which applications can respond to. With more and more activity-aware applications deployed in real-world environments, a research challenge emerges – discovering and learning new activities that have not been pre-defined or observed in the training phase. This paper tackles this challenge by proposing a hierarchical mixture of directional statistical models. The model supports incrementally, continuously updating the activity model over time with the reduced annotation effort and without the need for storing historical sensor data. We have validated this solution on four publicly available, third-party smart home datasets, and have demonstrated up to 91.5% accuracies of detecting and recognising new activities.

Index Terms—Activity recognition, Online learning, Incremental learning, Active learning, Semi-supervised learning, Mixture model, von Mises-Fisher distribution, Hierarchical mixture, Hierarchical clustering, Pervasive computing, Smart home

1 Introduction

We have witnessed a significantly increasing number of human activity-aware systems ranging from smart environments such as home and office to wearable and mobile computing. For example, smart energy applications are optimising heating configurations based on users activity routine, healthcare applications are making recommendations based on users' past physical activities, and intelligent notification systems are pushing notifications at opportune moments where users are most likely to attend according to their current context or task at hand, to name a few. These applications have an important implication in improving people's quality of life.

One of the enabling technologies for these applications is activity recognition; that is, the ability to recognise and predict user's activities from a wide range of sensors embedded in an environment or worn by users. Activity recognition has been a popular research topic in pervasive and ubiquitous computing for the last decade, and many knowledge- and data-driven techniques have been applied and they have achieved promising results on accurately recognising activities from training data. The success has encouraged moving an activity recognition system from lab settings to real-world home or office settings. This has introduced a new set of challenges in activity recognition and many assumptions that we have in the short-term, well-controlled lab-based experiments will not hold for a long-term, in-the-wild case study.

At the early stage, the study on activity recognition is composed of the following processes [1], [2]: (1) pre-define a closed set of activities of interest, and deploy a set of sensors that can potentially be used to identify them, (2) collect sensor data for a short period of time (say 2 weeks) and annotate them with activity labels, and (3) train a computational model with these collected data and then start recognising activities for the newly incoming sensor data. These processes assume a fixed set of activities whose patterns will not change over time.

However, neither the processes nor the assumption will hold when we deploy the system for a larger number of users for a much longer period of time. First of all, it is unrealistic to expect that users will only perform the pre-defined set of activities for a long time. When a user performs a new type of activity that has not been seen or learned during the training period, the pre-trained model might not be able to recognise it or simply mis-classify it. Secondly, user behaviours captured statically at the training period are limited, as user behaviours tend to evolve with time. Both the above situations could result in the degradation of the performance of activity recognition techniques, and thus could lead to undesirable services to the users.

This brings one important research question: *how to discover and recognise new emerging activities*. An activity recognition system that can continuously update its computational model with the change of activity pattern and the change of the set of activities of interest will enable a more sustainable deployment. Sensor based household applications usually generate high dimensional data. Inspired by the successful applications of von Mises Fisher distribution (vMF) on high dimensional data in other domains, we draw our attention to vMF and propose a novel statistical model, Hierarchical Mixture of Conditional Independent von Mises-Fisher distribution (HMCIvMFs) to solve this problem. The advantages of a generative machine learning technique, inherited by the proposed solution, are employed to effectively solve this dynamic learning problem and some other pressing problems that are common in real world deployments. To be more specific, we claim the following

- *L. Fang, J. Ye and S. Dobson are with the School of Computer Science, University of St Andrews, St Andrews, Fife, KY16 9SX, UK. E-mail: lf28@st-andrews.ac.uk.*

Manuscript received xxxx xx, xxxx; revised xxxx xx, xxxx.

novelties and contributions.

1) A novel statistical model features hierarchical mixture of conditional independent directional statistical models for activity recognition;
2) A model learning algorithm for the hierarchical mixture model is derived for both supervised and semi-supervised learning settings, i.e. available labelled training data is limited;
3) The generative model-based solution can be applied to accurately detect unknown activities;
4) It then can incrementally update the activity model to recognise new types of activities without the need of storing all the historic data;
5) Moreover, the algorithm is capable of carrying out model updates even when the data is sparely labelled;
6) To make the model update more efficient, various active learning strategies are investigated to minimise users' effort in manual annotation.

The rest of the paper is organised as follows. Section 2 introduces the background and challenges of discovering and recognising new emerging activities, and discusses the existing work that addresses towards this problem. Section 3 describes the background knowledge on vMF distribution and hierarchical mixture model. Section 4 illustrates the theoretical foundation of our approach, which is empirically validated and discussed in Section 5. The paper concludes in Section 6 and future directions are discussed.

2 Related Work

This paper aims towards discovery and recognition of new emerging activities that have not been observed in training data or defined *a priori*. Learning unseen pattern is naturally harder than traditional pattern recognition. First of all, it often lacks sufficient training data for these new patterns so it is difficult to build a robust model to recognise them. Secondly, imperfections in sensor data are a well-known problem [3], which can make new patterns indistinguishable from short-term noise. Moreover, humans can perform the same activity in different ways, resulting in a variety of patterns, which further complicates the discovery of emerging patterns. In recent years, different approaches have been attempted towards addressing this problem.

Abdallah et al. propose a system called *AnyNovel* that applies an incremental clustering model to discover and learn new physical activities from accelerometer data [4]. A clustering technique is used to cluster real-time data streams and identify a newborn cluster that lie outside the existing cluster decision boundary. Then a cohesion and separation criterion is utilised to validate this newborn cluster as a potential new activity through its density, weight, and temporal characteristics. Once validated, this new cluster will be annotated by the user with the support of active learning.

Similarly, Gjoreski et al. have used an agglomerative clustering technique to enable real-time clustering of streaming data [5]. To validate clusters for potential new activities, they have proposed two temporal assumptions on human activities; that is, a human activity usually lasts for a certain period of time and there should not be frequent transitions between activities. With these assumptions, they have been able to more accurately discover meaningful clusters.

Ye et al. use distance-based clustering to incrementally learn and recognise new daily routine activities in a smart home [6]. An activity profile is built on top of each pre-defined activity using training data and is modelled as a cluster with sufficient statistics to enable model drift without the need of storing historical data. Then each incoming sensor data will be assessed on each activity profile; if not matching any existing activity profile, it is considered as abnormal and taken for annotation query and an activity profile is built on this new cluster.

One-class support vector machines (OSVMs) are the mostly used technique in one-class classification, which have also been applied for discovering potential new activities [7], [8], [9]. In the training phase, a OSVM is first trained on the data of a set of pre-defined activities. In the test phase, the OSVM is used to discriminate whether the unlabelled instances conform to the observed data of pre-defined activities.

Yin et al. apply a OSVM with Gaussian Radian Basis Function (RBF) kernel to discover abnormal activities from body-worn sensors that collect light, temperature, sound, and acceleration data [7]. By selecting proper parameters, the OSVM can identify normal activity patterns with higher confidence and detect anomalies with low false negative rate. Similarly, Hu et al., under the assumption that abnormal activities occur with a lower likelihood, implement a OSVM with Fisher kernel to detect unseen abnormal activities by choosing parameters biased towards a low false negative rate [8].

Support vector data description (SVDD) [10], similar to OSVM, is also used for anomaly detection. By forming a hyperspace that encompasses all positive instances with minimal volume, SVDD can infer the anomaly of instances. For instance, Shin et al. use SVDD with Gaussian kernel to detect abnormal activities of elder people based on features extracted from infrared (IR) motion sensor data collected in houses [11]. Various kernel functions available to SVMs are one of the reasons for its success in novelty detection [7], [11], however tuning kernel parameters in high dimensional data to strike the balance between false positive and false negative rates is tricky in the absence negative instances.

Cheng et al. propose a two layer semantic attribute-based learning model to recognise new activities [9]. The low layer is composed by attribute detectors, where each attribute is a mid-level feature encoding specific semantic interpretations of low-level sensor features; for example, an attribute can be a physical activity such as running interpreted from accelerometer data. The mapping between sensor features with attributes can be learned through a data-driven model, such as OSVMs [9] or probabilistic graphical model [12]. The second layer is an activity recogniser, which is a knowledge-driven model where an activity-attribute matrix encodes the semantic relationship between activities and attributes. The inclusion of a new activities requires adding new attributes at the low layer and updating the activity-attribute matrix with a manually specified relationship between attributes and this new activity. By leveraging domain knowledge and using the shared attributes among activities, a zero-shot learner can be adapted to recognise new activity with limited training data.

Topic models have been applied to discover activity patterns from unseen sensor data in an unsupervised fashion [13]. The discovered activity patterns correspond to high-level activities that are composed by a set of low-level activity patterns [13], [14] or a set of sensor events [15], [16], [17]. The low-level activities, considered as words, are concrete and short-term activities such as body movement, location, object usage. The high-level activities, considered as topics, are abstract and complex activities such as daily routines (*working, having a meal*) [13], [14], location routines (*at work early in the morning*) [15], behaviours indicating functional health (*go to toilet at night*) [16].

Cook et al. have combined a compression-based sequence mining with clustering algorithms to discover new activity patterns on binary sensor events collected from a smart home environment [18], [19]. The key problem in sequence mining is how to find patterns of interest among a large number of patterns with various lengths. Cook et al. have applied the minimum description length principle to find patterns with sufficient frequency and length. The discovered patterns are those that best compress the dataset. Edit distance (Damerau-Levenshtein distance [20]) is further used to measure similarity between patterns and determine if a pattern is the variation of an existing pattern.

Directional statistical models especially von Mises Fisher based models have been studied and applied in different domains. Banerjee et al. proposed an EM based inference procedure for finite mixture of von Mises Fisher (vMF) [21]. Taghia et al. studied the variational learning on vMF mixture model [22]. Gopal and Yang derived the Bayesian learning on mixture of vMFs and some other extensions like Hierarchical mixture of vMFs (HMvMFs) [23]. The main difference between our model and their hierarchical model is the component density in our model admits multiple conditional independent vMFs rather than a singular one.

3 Background on Directional Statistics and Mixture Model

von Mises-Fisher (vMF) Distribution and mixture model form the foundation of our proposed approach, where we perceive each activity composed of a variety of patterns as a mixture of vMFs.

3.1 von Mises-Fisher distribution

A vMF distribution is a probability distribution on unit length hypersphere, whose density function can be defined as

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

where $\boldsymbol{x} \in R^D$ is a D dimensional vector with unit length ($\|\boldsymbol{x}\|_2 = 1$), $\mathcal{I}_\nu$ denotes the modified Bessel function of the first kind at order ν, $\boldsymbol{\mu}$ is the mean direction and κ is a concentration parameter indicating how concentrated the samples are generated against $\boldsymbol{\mu}$. When κ is large, the samples are closely aligned with $\boldsymbol{\mu}$, which tends to a point density [24]. When κ is small, or close to zero, the model degenerates to the uniform distribution on the sphere, implying each direction vector has the equal probability density.

A vMF distribution is one of the exponential family distributions, whose parameters can be parsimoniously summarised by its sufficient statistic, $\boldsymbol{\gamma} = \sum_{i=1}^n \boldsymbol{x}_i$; and can be estimated by Maximum Likelihood (ML) method. In particular, the mean direction $\boldsymbol{\mu}$ has a closed form estimator while κ can be estimated approximately [21]:

$$\hat{\boldsymbol{\mu}} = \frac{\boldsymbol{\gamma}}{\|\boldsymbol{\gamma}\|_2}, \;\; \hat{\kappa} = \frac{\bar{R}D - \bar{R}^3}{1-\bar{R}^2}, \tag{1}$$

where $\bar{R} = \frac{\|\boldsymbol{\gamma}\|_2}{n}$ is called the mean resultant length in directional statistics community. The above ML estimation serves the basis of the M step of the EM algorithm derived later.

3.2 Mixture and hierarchical mixture model

A mixture model is a probabilistic model to represent a complex population as a mixture of a finite number of different sub-populations. It assumes the data samples are independently generated by a number of $k (k \geq 1)$ components. The model implicitly assumes hidden categorical variables $z^i \in \{1, \ldots, k\}$ that tells which hidden component originally generates $\boldsymbol{x}^i$. Formally, the model has a probability density function

$$f(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{h=1}^{k} \pi_h f(\boldsymbol{x}|\theta_h),$$

where $\boldsymbol{\theta} = \{\pi_1, \ldots, \pi_k, \theta_1, \ldots, \theta_k\}$, and θ_h is the parameters of the hth component of the mixture model and the $\pi_h = p(z^i = h)$ is the prior probability of the membership. Given the number of components k, a mixture model of exponential family distributions can be learnt by Expectation-Maximisation (EM) algorithm [25], [26]. The algorithm iterates between E and M steps, and guarantees a local optimal estimate with respect to the likelihood. A mixture of von Mises Fishers distribution (MvMFs) and its model learning algorithm has been studied in [21], [27].

A hierarchical mixture model assumes that each subpopulation model is a finite mixture with possibly different sizes. The model admits a density

$$f(\boldsymbol{x}|\boldsymbol{\Theta}, \boldsymbol{\phi}) = \sum_{a=1}^{K} \phi_a f_a(\boldsymbol{x}|\boldsymbol{\theta}_a) = \sum_{a=1}^{K} \phi_a \sum_{h=1}^{k_a} \pi_h^a f(\boldsymbol{x}|\theta_h^a),$$

where $\boldsymbol{\phi} = \{\phi_1, \ldots, \phi_K\}$ is the mixing proportion of the top mixture and $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K\}$ is the collection of the parameters of the K lower level mixtures. Hierarchical mixtures of Gaussian have been proposed and studied for various applications [28], [29].

4 Proposed Solution

In this section, we first give an overview of the proposed solution framework by introducing the overall steps involved at a higher level. The proposed statistical model which is used in each of the steps is detailed later. Alternative model choices are also discussed such that the proposed model choice can be compared and justified.

4.1 Overview

A general process to discover and recognise unknown activities, proposed in this work, consists of the following three steps:

1) **Model Initialisation** – building a model for the existing known activities in the training data;
2) **Unknown Detection** – assessing whether newly collected sensor events match to any pre-known activity profile, and if so, inferring the activity label. Based on the assessment result, the unknown data is filtered out to form a candidate pool.
3) **Model Update** – updating the initial model to accommodate the emerging pattern. User input is required at this stage to help annotate emerging patterns with an activity label if necessary.

The above three steps are presented in 4.2, 4.3, and 4.4 respectively.

In real world applications, step 2) is applied to each incoming activity data record, while step 3) can be carried out at certain frequencies, say hourly or daily, depends on the application.

4.2 Activity modelling by HMCIvMFs

We propose a hierarchical mixture of conditional independent von Mises-Fisher distribution (HMCIvMFs) as a generative statistical model for activity modelling. In an overview, the model consists of two layers of mixture models. At the top level, a mixture of activity models is formed, where each mixture component represents a known activity, say cooking; at the second layer, a mixture of conditional independent von Mises-Fisher distributions is used to capture various sub-patterns of that activity. In the following sections, we elicit the lower-level activity model first, and complete the introduction of the full model. The model inference and learning algorithms are introduced at the end.

4.2.1 Mixture of CIvMFs for individual activity

As shown in the literature [30], users usually perform an activity in different ways; for example, a user might cook with different appliances, depending on the occasions or seasons. To capture this variety, we propose to model each activity as a finite mixture of vMFs, where each component corresponds to one sub-variate of the activity under consideration. The formal definition is given as follows:

$$f_a(\boldsymbol{x}|\boldsymbol{\theta}_a) = \sum_{h=1}^{k_a} \pi_h^a f_h(\boldsymbol{x}|\theta_h^a), \tag{2}$$

where $\boldsymbol{\theta}_a = \{\pi_1^a, \ldots, \pi_{k_a}^a, \theta_1^a, \ldots, \theta_{k_a}^a\}$, $\theta_h^a = \{\boldsymbol{\mu}_h^a, \kappa_h^a\}$ is the parameters of the hth vMF component and the π_h^a are the mixture proportions. For simplicity, when the context is clear, the activity label a is omitted in the following sections.

4.2.1.1 Conditional independent (CI) vMFs assumption: The above mixture model assumes that a sub-variate of an activity can be sufficiently modelled by a single vMF. In other words, each data point $\boldsymbol{x}^i$, conditioning on its mixture membership z^i, is a random draw from a D dimensional vMFs. This assumption is not enough to capture the full characteristics of the collected data. For example, the time feature and other sensor features should ideally be treated separately as they differ in many ways. For example, the time feature, upon the cyclic transformation (discussed in 5.2), is essentially a von Mises distribution (a 2-d vMF variate), whereas the sensor features are usually of much higher dimensions. A better approach, therefore, is to treat them as two directional vectors on two spheres.

In light of this, we propose the following conditional independence assumption; that is, conditioning on $z^i = h$, we assume $\boldsymbol{x}^i$ is generated by S independent vMFs, i.e.

$$f_h(\boldsymbol{x}|\theta_h) = \prod_{v=1}^{S} f_{h,v}(\boldsymbol{x}_v|\theta_{h,v}),$$

where $\boldsymbol{x} = [x_1, x_2, \ldots, x_S]^1$, i.e. $\boldsymbol{x}$ is partitioned as a collection of sub-vectors based on some criterion. In particular, the CI assumption made in this work is to decompose $\boldsymbol{x}$ as the sensor features $\boldsymbol{x}_s$, and time feature $\boldsymbol{x}_t$, i.e. $\boldsymbol{x} = [\boldsymbol{x}_s, \boldsymbol{x}_t]$, where $\boldsymbol{x}_s$ and $\boldsymbol{x}_t$ are unit vectors with $D-2$ and 2 dimensions respectively. The implied density becomes

$$f_h(\boldsymbol{x}|\theta_h) = f_{h,s}(\boldsymbol{x}_s; \boldsymbol{\mu}_{h,s}, \kappa_{h,s}) f_{h,t}(\boldsymbol{x}_t; \boldsymbol{\mu}_{h,t}, \kappa_{h,t}), \tag{3}$$

where $f_{h,s}$ and $f_{h,t}$ denote the individual vMFs matching the sensor/time components, and $\theta_h = \{\boldsymbol{\mu}_{h,s}, \kappa_{h,s}, \boldsymbol{\mu}_{h,t}, \kappa_{h,t}\}$.

The CI assumption, bringing in model flexibility, still ensures the model fitting algorithm is tractable. The derivation of the EM algorithm, listed in the supplemental file, shows that the complete data likelihood is decoupled which leads to a closed-form M step. It is also worth noting that the different data components, although assumed conditionally independent, are not independent at the mixture level. That is, the covariance matrix, $\mathrm{cov}[\boldsymbol{x}]$, for $\boldsymbol{x}$ admitting the CI mixture model is not block diagonal, where the off-diagonal non-zero entries indicate the cross-feature statistical dependence. This implies the possible correlations between the time and sensor features of the modelling activity can be captured. We prove this result and give an empirical analysis of the covariance in the supplemental file.

4.2.2 Hierarchical mixture model for activity ensemble

Based on the introduced activity model, the whole activity ensemble can be represented as a finite mixture model with the activity label as the mixture membership. Formally, the model can be written as

$$f(\boldsymbol{x}|\boldsymbol{\Theta}, \boldsymbol{\phi}) = \sum_{a=1}^{K} \phi_a f_a(\boldsymbol{x}|\boldsymbol{\theta}_a),$$

where $\boldsymbol{\phi} = \{\phi_1, \ldots, \phi_K\}, \boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K\}$, $\boldsymbol{\theta}_a$ is the parameters set for each activity's MCIvMFs model and K is the number of known activities.

4.2.3 Model learning

The model parameters $\{\boldsymbol{\Theta}, \boldsymbol{\phi}\}$ can be learnt by Maximum Likelihood estimation at two different stages. The top layer of the model is actually a generative mixture model with known data membership. A more general model fitting algorithm to handle incomplete data with missing labels is

[1] We have omitted the data index i to avoid cluttering notations.

presented later in 4.4.2. Denoting $\boldsymbol{X} = \{\boldsymbol{x}^i\}, \boldsymbol{Y} = \{y^i\}$ as the data set and their activity labels, the log-likelihood is

$$\ln P(\boldsymbol{X}, \boldsymbol{Y} | \boldsymbol{\Theta}, \boldsymbol{\phi}) = \sum_{i=1}^{n} \sum_{a=1}^{K} \mathbb{I}(y^i = a) \left(\ln \phi_a + \ln f_a(\boldsymbol{x}^i | \boldsymbol{\theta}_a) \right), \tag{4}$$

where $\mathbb{I}$ denotes the $1/0$ identity function. Upon maximisation with respect to $\boldsymbol{\phi}$ with the constraint $\sum_a^K \phi_a = 1,\ \phi_a > 0$, the ML estimator for $\phi_1, \ldots, \phi_K$ is just the observed frequencies of each activity class,

$$\phi_a = \frac{n_a}{\sum_{a=1}^K n_a},\ a = 1, \ldots, K \tag{5}$$

where $n_a = \sum_{i=1}^n \mathbb{I}(y^i = a)$ denotes the number of activity a's instances in the training data.

The ML estimator for each $\boldsymbol{\theta}_a$ however needs an iterative EM algorithm to learn as the class membership within each lower level mixture model is unobserved. The likelihood function to maximise with respect to $\boldsymbol{\theta}_a$ for each $a = 1, \ldots, K$ is

$$\begin{aligned} \ln P(\boldsymbol{X} | \boldsymbol{\theta}_a) &= \sum_{i=1}^{n} \mathbb{I}(y^i = a) \ln f_a(\boldsymbol{x}^i | \boldsymbol{\theta}_a) \\ &= \sum_{i: y^i = a} \ln f_a(\boldsymbol{x}^i | \boldsymbol{\theta}_a), \end{aligned}$$

which only involves activity a's data likelihood. Therefore, the ML estimators for $\boldsymbol{\theta}_a$ $(1 \leq a \leq K)$ can be fit independently with their corresponding data set $\boldsymbol{X}_a \triangleq \{\boldsymbol{x}^i : y^i = a\}$. The following EM algorithm can be applied to find the ML estimator of $\boldsymbol{\theta}_a$.

4.2.3.1 EM for MCIvMFs: An EM algorithm is derived to find the ML estimator of a MCIvMFs. The difference between this algorithm and the one derived in [21] is that our algorithm accommodates the case when the mixture component is multiple conditional independent vMFs.

For each individual activity, the following algorithm is used to estimate its model parameter $\boldsymbol{\theta}_a$ for a pre-specified k_a, which can be determined by an information criterion like BIC. The algorithm is initialised with a starting parameter set $\boldsymbol{\theta}_a$, and iterates through the following two steps until convergence:

[E step]

$$P(z_a^i = h | \boldsymbol{x}^i, \boldsymbol{\theta}_a) \equiv r_{i,h} \propto \pi_h f_h(\boldsymbol{x}^i | \theta_h),$$

where $f_h(\boldsymbol{x}^i | \theta_h) = f_{h,s}(\boldsymbol{x}_s^i; \boldsymbol{\mu}_{h,s}, \kappa_{h,s}) f_{h,t}(\boldsymbol{x}_t^i; \boldsymbol{\mu}_{h,t}, \kappa_{h,t})$;

[M step]

$$\pi_h = \frac{1}{n} \sum_{i=1}^{n} r_{i,h}, \quad \boldsymbol{\mu}_{h,s} = \frac{\tilde{\boldsymbol{x}}_{h,s}}{\|\tilde{\boldsymbol{x}}_{h,s}\|}, \quad \boldsymbol{\mu}_{h,t} = \frac{\tilde{\boldsymbol{x}}_{h,t}}{\|\tilde{\boldsymbol{x}}_{h,t}\|},$$

$$\kappa_{h,s} = \frac{(D-2)\bar{R}_{h,s} - \bar{R}_{h,s}^3}{1 - \bar{R}_{h,s}^2}, \quad \kappa_{h,t} = \frac{2 \times \bar{R}_{h,t} - \bar{R}_{h,t}^3}{1 - \bar{R}_{h,t}^2},$$

where

$$\tilde{\boldsymbol{x}}_{h,s} = \sum_{i=1}^{n} r_{i,h} \boldsymbol{x}_s^i, \quad \tilde{\boldsymbol{x}}_{h,t} = \sum_{i=1}^{n} r_{i,h} \boldsymbol{x}_t^i,$$

$$\bar{R}_{h,s} = \frac{\|\tilde{\boldsymbol{x}}_{h,s}\|}{\sum_{i=1}^n r_{i,h}}, \quad \bar{R}_{h,t} = \frac{\|\tilde{\boldsymbol{x}}_{h,t}\|}{\sum_{i=1}^n r_{i,t}}, \tag{6}$$

and D is the dimension of a feature vector $\boldsymbol{x}^i$.

The algorithm can be easily extended for more refined CI assumptions. The detailed derivation of the EM algorithm for a mixture of a general CI vMF model is listed in the supplemental file.

4.2.4 *Activity recognition inference*

The activity recognition problem can be formally solved as a probabilistic inference procedure. Based on Baye's theorem, given a data instance $\boldsymbol{x}^i$, the probability that it belongs to activity a is

$$P(y^i = a | \boldsymbol{\Theta}, \boldsymbol{\phi}, \boldsymbol{x}^i) \propto \phi_a f_a(\boldsymbol{x}^i | \boldsymbol{\theta}_a) = \phi_a \sum_{h=1}^{k_a} \pi_h^a f_h(\boldsymbol{x}^i | \theta_h^a),$$

for $a = 1, \ldots, K$. By Baye's optimal decision theory, the classified activity is just

$$y^i = \operatorname*{argmax}_a P(y^i = a | \boldsymbol{\Theta}, \boldsymbol{\phi}, \boldsymbol{x}^i).$$

4.3 Unknown activity detection

We make use one of the important features of vMF distribution, i.e. when $\kappa \to 0$, the distribution becomes uniform over the hypersphere. As unseen patterns can emerge from any direction over the sphere, we can represent an unknown activity as an additional uniformly distributed vMF. Based on this unique feature of vMFs, we come up with the following strategy.

We introduce an additional pseudo component $K+1$ at the top layer to represent any unknown data (i.e., the space has not been covered by the learnt K components in $\boldsymbol{\Theta}$). By calculating the posterior probability of $\boldsymbol{x}$ in each of these $K+1$ components, we can infer whether $\boldsymbol{x}$ is known or not. We model the $(K+1)$th component as $\boldsymbol{\theta}_{K+1} = \{\boldsymbol{\mu}_{K+1,s}, \kappa_{K+1,s}, \boldsymbol{\mu}_{K+1,t}, \kappa_{K+1,t}\}$, where

$$\boldsymbol{\mu}_{K+1,t} = -1 \times \frac{\sum_{i=1}^n \boldsymbol{x}_t^i}{\|\sum_{i=1}^n \boldsymbol{x}_t^i\|}, \quad \kappa_{K+1,t} = \epsilon_t$$

$$\boldsymbol{\mu}_{K+1,s} = -1 \times \frac{\sum_{i=1}^n \boldsymbol{x}_s^i}{\|\sum_{i=1}^n \boldsymbol{x}_s^i\|}, \quad \kappa_{K+1,s} = \epsilon_s$$

where ϵ_t and ϵ_s are two small constants close to zero. The above estimators are the ML estimators of vMF models with the data set $\{-\boldsymbol{x}^i\}$ and the known concentration parameter $\kappa = \epsilon$, according to (1). Note that $-1 \times \boldsymbol{x}^i$ is the vector that differs the most from $\boldsymbol{x}_i$ based on the cosine distance, which matches the belief that new activities should be different from any existing pattern. The diffused concentration κ also can be viewed as the lack of confidence regarding this unknown activity model, leading to a uniformly distributed density over the whole vector sphere.

In this work, we set ϵ by a simple heuristic rule: find κ such that

$$\frac{\min \text{vMF}_D(\kappa)}{\max \text{vMF}_D(\kappa)} \geq p,$$

here vMF_D denotes the density of a D-dimensional vMF, i.e. the minimum density of the vMF is at least $100 \times p\%$ of the maximum density. Based on the density function, it is easy to find the rule for κ:

$$\frac{\min \text{vMF}_D(\kappa)}{\max \text{vMF}_D(\kappa)} = \frac{f(-\boldsymbol{\mu} | \boldsymbol{\mu}, \kappa)}{f(\boldsymbol{\mu} | \boldsymbol{\mu}, \kappa)} = \frac{e^{-\kappa}}{e^{\kappa}} \geq p \Rightarrow \kappa \leq -\frac{1}{2} \ln P.$$

We find the results are insensitive to any $p \geq 0.5$. In this work, we set $p = 0.9$.

Based on the Baye's rule, we can find out those unknown events by calculating

$$P(\boldsymbol{x}^i \text{ unknown}|\boldsymbol{\Theta}, \boldsymbol{\phi}, \boldsymbol{x}^i) = P(y^i = K+1|\boldsymbol{\Theta}, \boldsymbol{\phi}, \boldsymbol{x}^i).$$

The vector $\boldsymbol{x}^i$ will be identified as an unknown activity if $\text{argmax}_a\, p(y^i = a|\boldsymbol{x}^i) = K+1$; i.e., the probability of $\boldsymbol{x}^i$ is from an unknown activity component is the largest. Otherwise, it is classified as known.

4.4 Model update and active learning

When the unknown activity instances are filtered out, the instances form an unknown data pool, denoted as $\boldsymbol{X}_{unkn}$. The unknown observations need to be presented to the user for annotation; then the annotated instances can be learnt and updated to the initial model. In this section, we present this model update procedure.

Instead of querying every single instance, we also explore the possibility of employing active learning strategies to minimize human annotation effort, i.e. only select a subset $\boldsymbol{X}' \subseteq \boldsymbol{X}_{unkn}$ to label; and the annotated labels are denoted as $\boldsymbol{Y}'$. We hope to maintain good activity recognition accuracies of the final updated model while keeping the annotated sample size small with the help of active learning.

4.4.1 Model update with fully annotated data

We first consider the case when all the data in the candidate pool is annotated and used for model update, i.e. $\boldsymbol{X}' = \boldsymbol{X}_{unkn}$. Note that there are two possible cases for the labels: one is that the annotated instance is a completely new activity or $y^{i'} \notin \{1, \dots, K\}$ where i' is used to index the model-update data set; the other is that the instance belongs to one of the existing activities, i.e. $y^{i'} \in \{1, \dots, K\}$.

4.4.1.1 New activity: The first case can be easily handled as the model for each activity is fit in parallel as the log likelihood is decoupled according to (4). Therefore, to update $\{\boldsymbol{\phi}, \boldsymbol{\Theta}\}$, one first needs to learn a MCIvMFs with parameter $\boldsymbol{\theta}_{K+1}$ for the new activity, and then combine it with the existing parameter set; the model update procedure for $\boldsymbol{\phi}$ can be easily derived based on (5) as

$$\boldsymbol{\phi} \leftarrow \left[\frac{N}{N+n'_{K+1}}\boldsymbol{\phi}^T, \frac{n'_{K+1}}{N+n'_{K+1}}\right]^T,$$
$$\boldsymbol{\Theta} \leftarrow \{\boldsymbol{\Theta}, \boldsymbol{\theta}_{K+1}\},\ N \leftarrow N + n'_{K+1},$$

where n'_{K+1} is the size of the data instances belonging to the new activity $K+1$ and $N = \sum_{a=1}^{K} n_a$ denotes the initial training data size. The extended vector is just the renormalised proportions of the activity instances based on (5). Note that to make the update possible, we need to keep the training data size N as an extra parameter.

4.4.1.2 New sub-pattern of existing activities: The second case deals with existing activities' model update, i.e. update $\boldsymbol{\theta}_a$ for $a = 1, \dots, K$ with the newly discovered data $\boldsymbol{X}'_a = \{\boldsymbol{x}^{i'} : y^{i'} = a\}$. A simple strategy is to rerun the EM algorithm with the extended data set $\boldsymbol{X}_a \cup \boldsymbol{X}'_a$. Although conceptually simple, this approach requires storing historic sensor data, which may not be feasible given the ongoing and continuous monitoring nature of the applications. We instead run the EM algorithm to fit a MCIvMFs on $\boldsymbol{X}'_a$ alone, which results in a k'_a mixture with fitted parameters $\boldsymbol{\theta}'_a = \{\pi'_1, \dots, \pi'_{k'_a}, \theta'_1, \dots, \theta'_{k'_a}\}$; then update the model by combining these two mixtures: the combined model has $k_a + k'_a$ mixture components with a renormalised mixing proportion

$$\boldsymbol{\pi}_a \leftarrow \left[\frac{n_a}{n_a+n'_a}\boldsymbol{\pi}_a^T, \frac{n'_a}{n_a+n'_a}\boldsymbol{\pi}_a'^T\right]^T,\ n_a \leftarrow n_a + n'_a,$$

and component parameters $\{\theta_1, \dots, \theta_{k_a}, \theta'_1, \dots, \theta'_{k'_a}\}$.

The above update procedure is actually a modified EM algorithm for $\boldsymbol{X}_a \cup \boldsymbol{X}'_a$ with a hard assignment E step, i.e. for $\boldsymbol{x} \in \boldsymbol{X}'_a$:

$$P(z_a = h|\boldsymbol{x}, \boldsymbol{\theta}_a) \propto \begin{cases} 0 & : h = 1, \dots, k_a \\ \pi'_h f_h(\boldsymbol{x}|\theta'_h) & : h = k_a+1, \dots, k_a + k'_a \end{cases};$$

while the M step is unchanged. This zero responsibility assignment is justified by the fact that $\boldsymbol{x} \in \boldsymbol{X}'_a$, where $\boldsymbol{X}'_a \subseteq \boldsymbol{X}'$ contains data filtered out by the unknown detection algorithm 4.3, which implies that the existing mixture components have shown little support for $\boldsymbol{x}$ by the unknown detection algorithm. Note that the activity proportion $\boldsymbol{\phi}$ for the hierarchical mixture needs to be updated as well at the end as the individual activity count is changed. One just needs to re-estimate $\boldsymbol{\phi}$ according to (5) with the updated n_a.

The above two update procedures are similar as they both require fitting MCIvMFs to the newly discovered data. But the former case adds to the top layer of the hierarchical mixture as a new activity (represented as a MCIvMFs) while the later updates one of the existing activity models.

4.4.2 Model update with incomplete labels

When only a subset of data in the candidate pool is annotated, the previous update procedure will not work as some parameters required for the update procedure like n'_{K+1}, n'_a becomes unobserved as well. In particular, we have a fully annotated data set $\boldsymbol{X}' \subseteq \boldsymbol{X}_{unkn}$ and the complement unlabelled data: $\bar{\boldsymbol{X}} \triangleq \boldsymbol{X}_{unkn}/\boldsymbol{X}$ while the missing labels are denoted as $\bar{\boldsymbol{Y}}$.

The problem can be solved by treating the missing labels as hidden random variables and applying another layer of EM algorithm to learn the HMCIvMFs. The model learning algorithm for a HMCIvMFs with missing labels is listed in Algorithm 1; the detailed derivation of this EM algorithm is given in the supplemental file. Note that by setting $\bar{\boldsymbol{X}} = \emptyset$ we expectedly recover the algorithm discussed in 4.2.3, i.e. the supervised version of this algorithm. The EM algorithm finds the ML estimator $\boldsymbol{\Theta}', \boldsymbol{\phi}'$ for the data with incomplete labels. Then the unobserved parameters required for the update procedures listed in 4.4.1 can be estimated by:

$$n'_{K+1} = N' \times \phi'_{K+1},\ n'_a = N' \times \phi'_a,$$

where N' is the size of $\boldsymbol{X}_{unkn}$. The other required parameters are readily available from the output of the algorithm.

Algorithm 1 Model Learning Algorithm for HMCIvMFs with Missing Labels

```
Initialize $\boldsymbol{\Theta}, \boldsymbol{\phi}$ on the labelled data $\boldsymbol{X}', \boldsymbol{Y}'$
$t_{ia} \leftarrow \mathbb{I}(y^i = a)$ for each $\boldsymbol{x}^i \in \boldsymbol{X}', a = 1, \ldots, K$
repeat
    1. Expectation step of the EM:
    $t_{ia} \leftarrow \frac{\phi_a f_a(\boldsymbol{x}^i|\boldsymbol{\theta}_a)}{\sum_{a=1}^{K} \phi_a f_a(\boldsymbol{x}^i|\boldsymbol{\theta}_a)}$ for $\boldsymbol{x}^i \in \bar{\boldsymbol{X}}, a = 1, \ldots, K$
    2. Maximization step of the EM:
    for $a = 1, \ldots, K$ do
        $\phi_a \leftarrow \frac{1}{n} \sum_{i=1}^{n} t_{ia}$
        repeat
            $\tilde{r}^a_{ih} \leftarrow t_{ia} \pi^a_h f_h(\boldsymbol{x}^i|\theta_h)$
            for $h = 1, \ldots, k_a$ do
                $\boldsymbol{\mu}^a_{h,s} \leftarrow \frac{\tilde{\boldsymbol{x}}^a_{h,s}}{\|\tilde{\boldsymbol{x}}^a_{h,s}\|_2}, \ \boldsymbol{\mu}^a_{h,t} \leftarrow \frac{\tilde{\boldsymbol{x}}^a_{h,t}}{\|\tilde{\boldsymbol{x}}^a_{h,t}\|_2}$
                $\kappa^a_{h,s} \leftarrow \frac{(D-2)\bar{R}^a_{h,s} - (\bar{R}^a_{h,s})^3}{1 - (\bar{R}^a_{h,s})^2}$
                $\kappa^a_{h,t} \leftarrow \frac{2 \times \bar{R}^a_{h,t} - (\bar{R}^a_{h,t})^3}{1 - (\bar{R}^a_{h,t})^2}$
                    ▷ where $\bar{R}^a_{h,\cdot}$ and $\tilde{\boldsymbol{x}}^a_{h,\cdot}$ are defined as (6)
            end for
            $\pi^a_h \leftarrow \frac{\sum_{i=1}^{n} r^a_{ih}}{\sum_{h=1}^{k_a} \sum_{i=1}^{n} r^a_{ih}}$
        until convergence
    end for
until convergence
```

4.4.3 Active learning based model update

Instead of random sampling, we also investigate the possibility of applying active learning strategies to alleviate the labelling effort. The objective is to minimize the number of queried instances while maintaining good recognition accuracies achieved by the updated model by selecting the most informative subset $\boldsymbol{X}'$ for annotation.

Model uncertainty based strategy Inspired by the active learning literature [31], we devised a simple two step sampling procedure. First, by querying the user with an initial subset, say half of the data planned to be queried, a probabilistic classifier, say a neural network with a softmax output unit, is trained where the class membership can be represented as a probability vector $P(z|x)$. Second, according to the class membership probability assigned by the classifier, the other half of the data points from the rest candidate pool are selected based on different uncertainty sampling strategies. The commonly used strategies are: least confidence [31], margin sampling [32] and entropy-based measure [33].

Data diversity based strategy The uncertainty sampling method can be limited as it ignores the diversity of the selection, which might lead to an unrepresentative sample. For example, the criteria might point all selections to one particular uncertain class but ignores the other instances. Moreover, most classifiers are trained with the pure objective to minimise the training error rather than discover class membership uncertainty, which later on serve as the only index for the sampling criteria.

Based on the above observations, we propose another sampling strategy based on hierarchical clustering, which aims to select samples based on sampling diversity but also avoids the reliance on the inferred class membership probability. The motivation is to select the most diversified subset of data from the unlabelled. The algorithm consists of three steps.

1) Apply hierarchical clustering algorithm to the unknown data;
2) Cut the clustering tree at the level H such that the data set forms H clusters, i.e.

$$\boldsymbol{X}_{unkn} = \bigcup_{n=1}^{H} \mathrm{C}_n$$

3) Randomly choose k data sample from each cluster C_n for $n = 1, \ldots, H$, the chosen data set with $k \times H$ instances is then forwarded to the user to label.

Unlike the uncertainty sampling based method, there is no need for the initial annotation step. Hierarchical Clustering is chosen mainly because it forms a full clustering trajectory with different cluster size [34]; a flexible number of clusters can then be formed to facilitate the selection process. For example, to choose H samples, one simply cuts the tree at the H level, and select $k = 1$ instance from each of the H clusters. In this work, as all data instances are represented as directions, a bottom-up or agglomerative hierarchical clustering algorithm with an average linkage and the cosine distance is used.

5 Experiment and Evaluation

The main goal of the evaluation is to assess how good the proposed approaches are at capturing unknown activities and learning and recognising the activity labels. In particular, we also want to verify the proposed model is a suitable choice for real-world datasets. In the following, we will introduce the evaluation methodology including the datasets and evaluation metrics, and discuss the experiments carried out together with their results.

5.1 Experiment data set

We choose four real-world smart home datasets for evaluation. The first two datasets (House A and House B) are collected by the University of Amsterdam and on a single-resident house that is instrumented with wireless sensor network. The sensors are state-change binary sensors attached to household objects like cupboards and doors [35]. The recorded activities include making breakfast, cooking dinner, leaving the house, sleeping, and taking shower, etc. The second dataset is collected by a testbed at the Washington state university [2]. The dataset include 9 different activities and 32 sensors. Another dataset under consideration is the PlaceLab Couple dataset [36], which contains over 200 object sensors and was gathered over a period of 15 days with a total of 7 activities being recorded throughout the study. A summary of the key information of the four datasets are listed in Table 1. The four datasets are of various dimensions, class sizes, and instances, which implies different noise levels and classification difficulty. We believe the

[2] http://ailab.wsu.edu/casas/datasets/

four datasets are good representatives of the sensor based human activity literature and should give comprehensive evaluation results.

TABLE 1: Summary of the four datasets used

	Sensor Size	Activity Size (K)	Min. Train
House A	16	7	10%
House B	24	8	34%
Wst. Uni	85	9	19%
Place Lab	702	7	34%

5.2 Data preprocessing and feature extraction

We segment sensor events into time slots of a fixed interval (say one minute). For each time slot, we extract features on sensor data and timestamps. A sensor feature vector is represented as $\boldsymbol{x}_s = [x_1, x_2, ..., x_{S'}]$, where S' is the number of sensors being installed, and 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. As the datasets we are working on contain only the binary sensors, we construct $\boldsymbol{x}_s$, where x_i is the frequency of this sensor being activated. That is, $x_i = N_i/N$, 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 a real-valued scalar feature, we apply the standard trigonometric transformation for cyclic data [37], that is

$$\boldsymbol{x}_t = (\cos\theta, \sin\theta), \text{where } \theta = h \times (2\pi/24). \tag{7}$$

This transformation makes sure each time stamp distinguishable but also time distance measure consistent. Note that the transformed time feature is a two dimensional unit vector on a circle, which is ideally modelled by a von Mises distribution. In the end, the combined sensor and time features forms a D-dimensional feature vector with $S' + 1$ sensor/time components, denoted as $\boldsymbol{x}$.

5.3 Evaluation Criteria

In line with the existing literature on human activity recognition [35], [38], we use two criteria 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

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

The reported F-score is by class average over the activities.

5.4 Experiments and results

To comprehensively examine the solution, we evaluate the proposed methods by four experiments revolving the claimed contributions: activity recognition; unknown detection; on-line learning and model updating; and finally the effect of active learning in relieving annotation effort. The derived algorithms and the following experiments are implemented in Matlab [3] and the figures are generated by the Gramm package [39].

5.4.1 Activity recognition

Firstly, we assess the activity classification performance of the solution on a traditional static setting in which all instances' activity labels are assumed known a priori.

Evaluation method To demonstrate the effectiveness of HMCIvMFs, we compare it with four other statistical model based classifiers, namely Mixture of von Mises Fisher (MvMFs) [21], Hierarchical Mixture of von Mises Fishers (HMvMFs) which was first studied and applied in text mining [23], and their two Gaussian counterparts: Mixture of Gaussians (MGs) and Hierarchical Mixture of Gaussians (HMGs). 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. Matlab's Statistics and Machine Learning toolbox [40] is used to evaluate the above algorithms where (hyper-)parameters are either specified based on the properties of the datasets or tuned automatically by the provided software package, and the details are listed in the supplemental file. A ten-fold cross validation is used for this comparison, which implies 90% of the data is used for training. To simulate an alternative situation where labelled data is scarce, we carry out an additional experiment when the training/testing ratio is deliberately set small. In particular, as the class sizes are imbalanced among the datasets, the ratio is set such that each class at least has 3 instances (the number such that a spherical-covariance Gaussian can be fit); the minimum ratios for the four datasets are listed in Table 1. One hundred random experiments were run for the limited training data case.

Results The results for House A and Washington Data are listed in Table 2, 3 respectively. The results for the other two datasets show a very similar pattern. It can be seen that the proposed model HMCIvMFs is the clear winner among the five generative classifiers in both the data abundant and scarce scenarios. The vMF based solutions in general perform better when the data dimension is larger, which shows its advantage over Gaussian for high-dimensional data modelling. The better performance of HMCIvMFs over the other two vMF based variates demonstrate the hierarchical mixture and conditional independence model assumptions are effective for the activity recognition task.

Comparing with discriminative classifiers, the overall stronger performance of the discriminative classifiers echoes some existing belief and research findings on the two schools of techniques [41]. Nevertheless, HMCIvMFs is always among the best cohort, although there is no overall winner especially for the data abundant case (ten-fold cross

[3] The implementation of the derived algorithms can be found at: http://leo.host.cs.st-andrews.ac.uk.

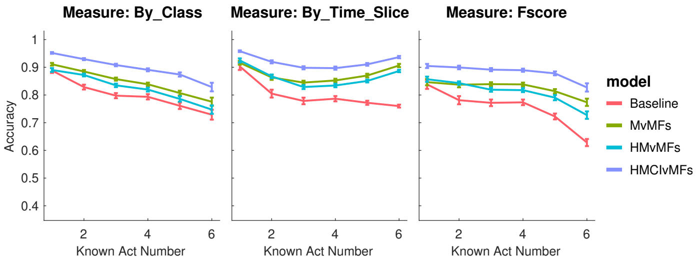

Fig. 1: Unknown detection accuracy of four methods on House A data;

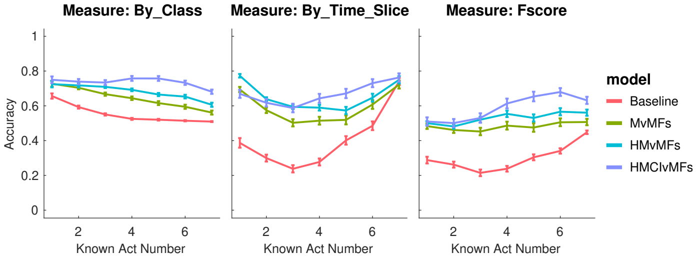

Fig. 2: Unknown detection accuracy of four methods on House B data;

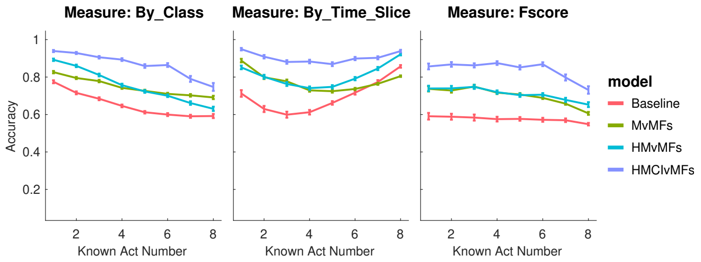

Fig. 3: Unknown detection accuracy of four methods on Washington data

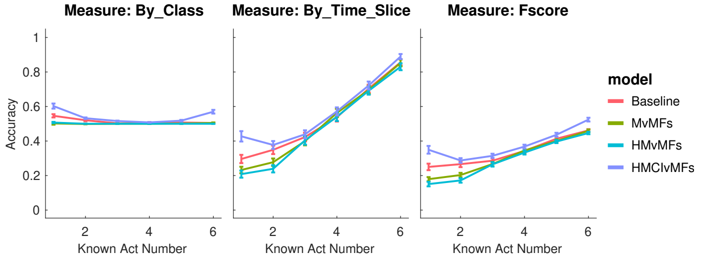

Fig. 4: Unknown detection accuracy of four methods on PlaceLab data

validation). While for the data scarce scenario, HMCIvMFs has better or comparable performance against those state of art discriminative classifiers, which demonstrates its value in real world applications.

5.4.2 *Unknown detection accuracy*

We then measure how well the solution can distinguish unknown activities. This experiment essentially evaluates the unknown detection strategy put forward in section 4.3.

Evaluation method To simulate the unknown activity scenario, we randomly pick a subset of k of the total activities as known and treat the rest $K - k$ as unknown. Only a proportion (50%) of the k known data is then used for training; and the rest data set formed by the unknown data and the rest of the known data is used for known/unknown detection assessment.

As it is not a standard binary classification problem, where both positive and negative data are used for training, we cannot compare the proposed solution to other state of art classifiers. We instead compare the proposed solution with three other similar methods: a baseline solution and two other variates of the statistical model based solution (HMvMFs and MvMFs). The baseline solution is an estimation based unknown detector presented in [6], in which the first and second statistical moments of the cosine distances is used to distinguish known and unknown instances.

Results The results of one hundred random experiments for each experiment setting on the four data sets are listed in Figure 1, 2, 3 and 4 respectively, where the means and their 95% confidence intervals are listed. The proposed model performs in general better than the other three methods across the different datasets and various known activity number settings. This further demonstrates the hierarchical and conditional independence model assumptions are effective. It is also evident that the vMF statistical model based solutions in general outperform the estimation based baseline solution except the PlaceLab data set. This is probably due to the noisy nature of the data set. It can also be observed that all methods uniformly exhibit a V shape trend for the by time slice measure, i.e. the accuracies first go down then pick up; this is because when the number of known is in the middle, there are more ways to partition the data which leads to large variances but also making the detection harder.

5.4.3 *Online recognition accuracy*

In this section, we examine the solution as a complete learning process. In particular, on top of the initial model learning and unknown detection, the effectiveness of learning and inferring those unknown activities, detailed in 4.4, is examined to see how the learning algorithm evolves.

Evaluation method To see how the solution performs at the end after model update with respect to the detected unknown instances, a proportion (20%) of the data is held out for final testing. The rest of the data is used for the training and model update purpose. Similar to the previous unknown detection experiment, a randomly selected k subset of the total activities are assumed known. A subset (50%) of the corresponding known data is used to train an initial model, as presented in Section 4.2. The rest of the known data together with those $K - k$ unknown activities' data are formed together to simulate the emerging data at the later stage. The update data will go through unknown detection first then a subset, with a predefined selection rate, of those filtered unknown are presented to the user for annotation and model learning. After the model update as presented in 4.4, the updated model is assessed at the end to classify the unseen test data. Note that the model update can happen more than once during the whole application process. In reality, the update can be set at certain frequency say on a daily or hourly basis. A key unique feature of this experiment is that not all the training data is presented at

TABLE 2: House A activity recognition result comparing with state of art

	By time slice limited data	By class limited data	F-score limited data	By time slice C.V.	By class C.V.	F-score C.V.
NNet	0.846 (0.043)	0.813 (0.049)	0.805 (0.049)	0.912 (0.031)	0.874 (0.051)	0.874 (0.043)
SVM	0.891 (0.024)	0.808 (0.03)	0.795 (0.031)	0.92 (0.019)	0.847 (0.013)	0.851 (0.02)
KNN	0.825 (0.037)	0.775 (0.046)	0.769 (0.043)	0.912 (0.03)	0.88 (0.051)	0.884 (0.054)
Random Forest	0.891 (0.02)	0.842 (0.032)	0.841 (0.033)	**0.926 (0.03)**	0.899 (0.06)	**0.895 (0.036)**
MG	0.84 (0.027)	0.806 (0.039)	0.808 (0.034)	0.887 (0.044)	0.875 (0.058)	0.857 (0.048)
HMG	0.85 (0.028)	0.814 (0.04)	0.817 (0.034)	0.852 (0.022)	0.83 (0.038)	0.831 (0.03)
MvMFs	0.767 (0.042)	0.749 (0.045)	0.741 (0.041)	0.796 (0.024)	0.823 (0.04)	0.793 (0.031)
HMvMFs	0.791 (0.043)	0.728 (0.045)	0.732 (0.041)	0.895 (0.022)	0.853 (0.047)	0.85 (0.059)
HMCIvMFs	**0.893 (0.019)**	**0.863 (0.029)**	**0.851 (0.029)**	0.912 (0.028)	**0.905 (0.047)**	0.882 (0.041)

TABLE 3: Washington Data activity recognition result comparing with state of art

	By time slice limited data	By class limited data	F-score limited data	By time slice C.V.	By class C.V.	F-score C.V.
NNet	0.882 (0.022)	0.759 (0.044)	0.756 (0.043)	0.905 (0.023	0.8 (0.051)	0.787 (0.038)
SVM	**0.921 (0.009)**	0.779 (0.034)	0.781 (0.029)	0.927 (0.011)	0.803 (0.022)	0.801 (0.014)
KNN	0.895 (0.015)	0.736 (0.035)	0.742 (0.037)	0.922 (0.021)	0.814 (0.061)	0.819 (0.059)
Random Forest	0.92 (0.009)	0.802 (0.025)	0.801 (0.022)	0.926 (0.015)	0.822 (0.035)	0.815 (0.029)
MG	0.776 (0.076)	0.733 (0.037)	0.605 (0.073)	0.65 (0.05)	0.678 (0.065)	0.618 (0.073)
HMG	0.808 (0.065)	0.748 (0.038)	0.652 (0.066)	0.792(0.067)	0.703 (0.032)	0.685 (0.043)
MvMFs	0.735 (0.047)	0.713 (0.036)	0.65 (0.037)	0.754 (0.036)	0.739 (0.049)	0.675 (0.039)
HMvMFs	0.888 (0.018)	0.801 (0.031)	0.775 (0.03)	0.892 (0.015)	0.76 (0.056)	0.735 (0.033)
HMCIvMFs	**0.921 (0.011)**	**0.841 (0.032)**	**0.827 (0.027)**	**0.929 (0.016)**	**0.86 (0.068)**	**0.847 (0.048)**

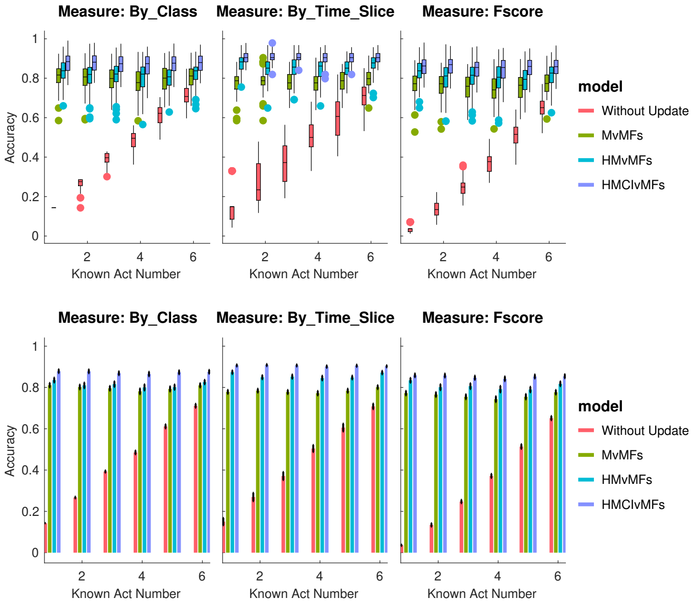

Fig. 5: Comparison between off-line learning without update and online model update on House A data set

the training stage but only a subset of pre-selected known activity data while all the remaining data is used to update the model on an on-going basis.

Results The experiment results over the four data sets are listed in Figure 5, 6, 7 and 8 respectively where both box-plots and bar-plots (the error bars are 95% CI of the mean) are shown. The selection rate here is set as 50% i.e. half of the marked unknown instances are randomly selected for annotation, while the whole filtered unknown data set is used for model update with the semi-supervised algorithm. As expected, the classification performance is poor when the training data is incomplete, i.e. when the known activity number is small; however, coupled with the unknown detection and model update, the algorithm correctly picks up those unknown or emerging instances and incorporates them into the existing model, which can be seen from the improved classification rates across all the unknown settings even when the pre-known activity number is limited. The boosted performance of the final updated model demonstrates the proposed model update procedure 4.4.1 and semi-supervised label learning algorithm detailed in 4.4.2 is effective. It is interesting to observe the same V shape trend that echoes what has been found in the previous unknown detection section. This is because the unknown detection is an integral step of the update procedure; therefore its performance affects the final results.

An ongoing assessment To see how the algorithm works in a continuous way with respect to different update settings, we also report the “ongoing” activity recognition accuracies. Instead of presenting the classification accuracy at the end on a separate test data set, we present the overall accuracy that essentially measures the percentage of the correctly classified instances during the whole training and update process. We vary the update interval, selection rate and known activity number to see how they affect the results. To save space, we only list the results for House A and Washington Data in Figure 9 and 10 where the shown results are the averages over 25 experiments, while the results on House B and Place Lab show similar trend. According to the figure, as expected, the accuracy improves when the initial

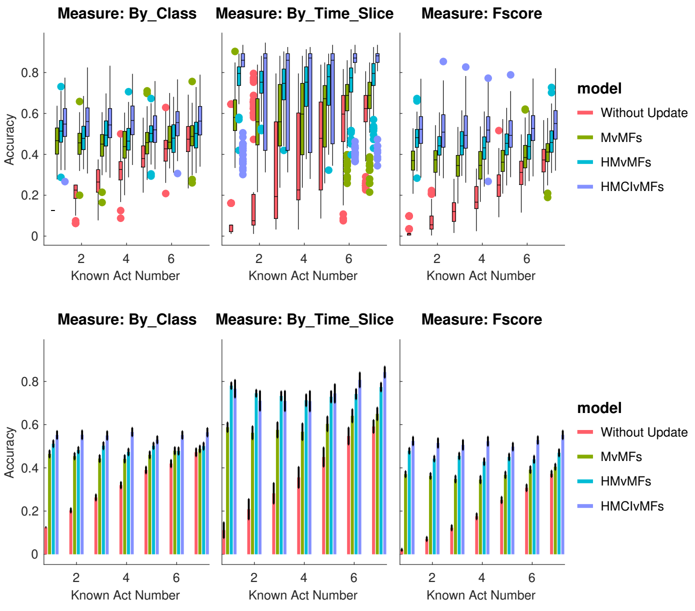

Fig. 6: Comparison between off-line learning without update and online model update on House B data set

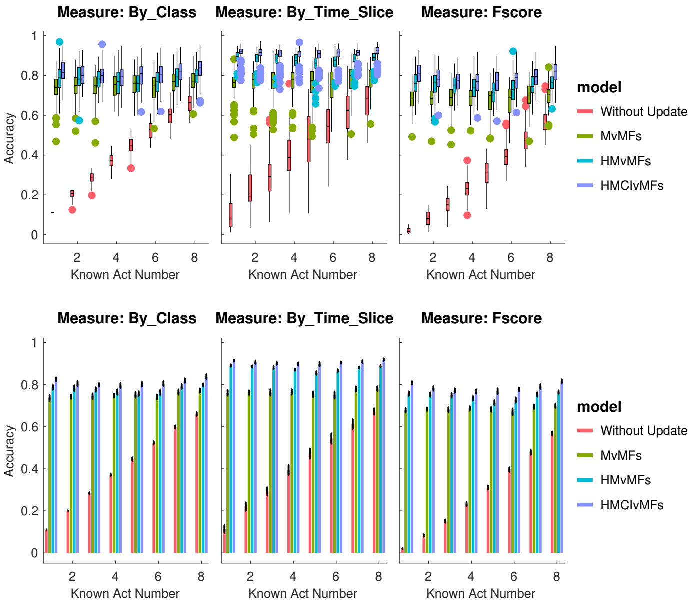

Fig. 7: Comparison between off-line learning without update and online model update on Washington Lab data set

knowledge is more complete. The model update settings, on the other side, affect the overall performance of the solution. When updated more frequently, say at every 50 instances, the solution achieves better overall performance. Whereas the selection rate affects the performance especially when the rate is low. When more unknown data is selected for annotation, the performance is better in general.

5.4.4 Active Learning

As shown in previous experiments, the selection rate affects the model performance. In this section, we examine whether active learning based model update strategies presented in 4.4.3 can help alleviate the annotation effort.

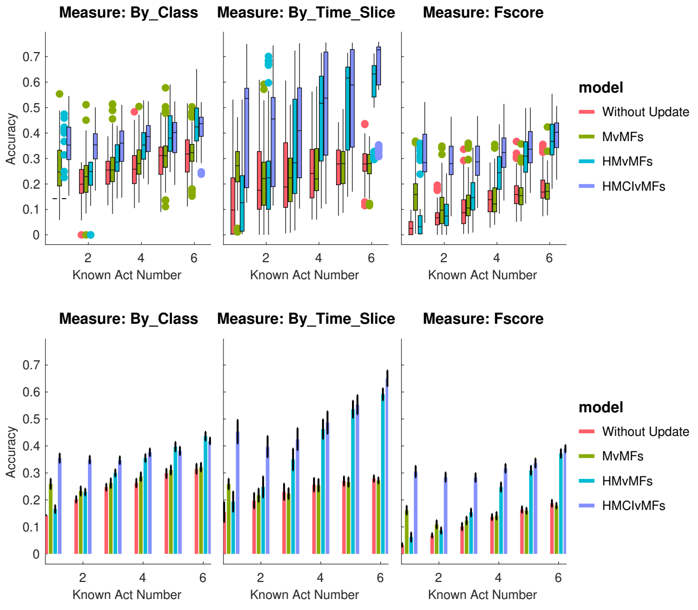

Fig. 8: Comparison between off-line learning without update and online model update on Place Lab data set

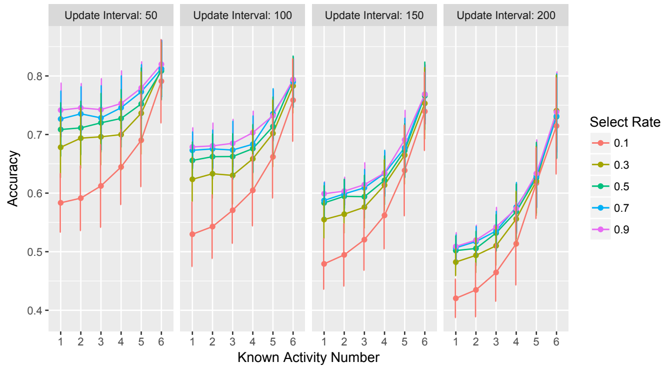

Fig. 9: An ongoing assessment on the model update procedure on House A data set

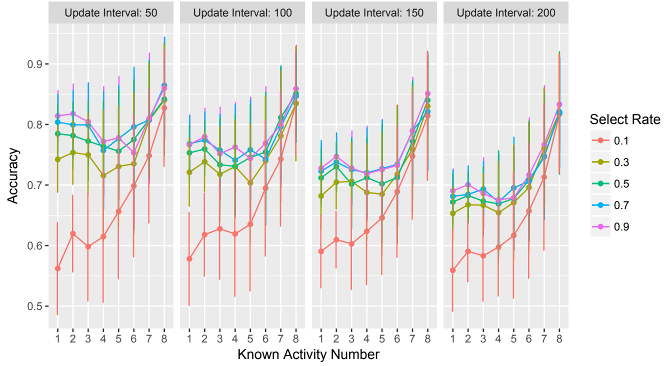

Fig. 10: An ongoing assessment on the model update procedure on Washington data set

Evaluation method Similar to the previous experiment, we compare the final recognition results achieved by applying different active learning strategies in the model update process. To be more specific, we vary the selection rate from 10% up to 1, and see how the strategies affects the final result.

Results To save space, only results of House A and Washington data are listed in Figure 11 and 12, while the other two data sets show similar trend. The shown is the averaged classification accuracy of the updated model over one hundred independent runs for each configuration (the error bars are 95% confidence interval of the means). By inspecting the figures, we can make the following observations. First of all, as suggested by the results, it is clear that not all data is required to be annotated, as the accuracies converge when selection rate is around 0.3-0.4 and stay there even when the selection rates increase. Secondly, there is no overall clear advantage of the three uncertainty sampling strategies (i.e. least confidence, smallest margin, entropy based) over random sampling, where similar results have been confirmed in previous active learning studies [42]. However, when the selection rate is small (< 0.4), active learning strategies seem deliver better results which are more evident for the Washington data. Among all the strategies, the diversity method is the best in delivering stable and consistent performance across the different settings especially when the selection rate is low.

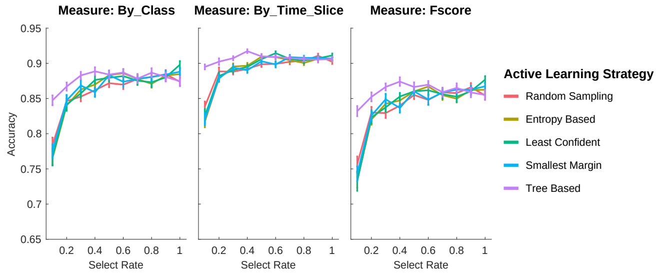

Fig. 11: Comparison between different active learning based model update strategies on House A data set

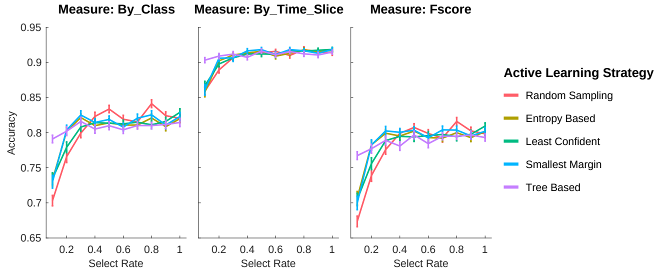

Fig. 12: Comparison between different active learning based model update strategies on Washington data set

6 Conclusion and Future Work

This paper addresses a new research question in activity recognition: discovering and recognising unknown human activities. This is a critical requirement for large-scale and long-term deployment of an activity-aware pervasive system in real-world environments, where it is inappropriate to assume that users will only perform a pre-defined closed set of activities all the time or the patterns of users performing such activities will be fixed. Towards addressing the challenge, this paper explores the solution space of various vMF based statistical models. The proposed model, featuring hierarchical mixture and conditional independence, works well not only in traditional activity recognition setting but also in detecting and learning new activities. The possibility of using active learning strategies is also explored, which shows promising performance in alleviating annotation effort.

In terms of future work, we intend to improve the robustness of the HMCIvMFs algorithm in the face of sensor errors. We plan to deploy both algorithms in real-world applications to assess the effectiveness of detecting and learning unknown activities further, and more importantly conduct the user studies to find the opportune moment to query the users for labelling unknown activities.

Acknowledgments

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, G. Stevenson, and S. Dobson, “KCAR: A knowledge-driven approach for concurrent activity recognition,” *Pervasive and Mobile Computing*, no. 0, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574119214000297

[2] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, “Sensor-based activity recognition,” *Trans. Sys. Man Cyber Part C*, vol. 42, no. 6, pp. 790–808, Nov. 2012. [Online]. Available: https://doi.org/10.1109/TSMCC.2012.2198883

[3] J. Ye, G. Stevenson, and S. Dobson, “Detecting abnormal events on binary sensors in smart home environments,” *Pervasive and Mobile Computing*, vol. 33, pp. 32 – 49, 2016.

[4] Z. S. Abdallah, M. M. Gaber, B. Srinivasan, and S. Krishnaswamy, “Anynovel: detection of novel concepts in evolving data streams,” *Evolving Systems*, pp. 1–21, 2016.

[5] H. Gjoreski and D. Roggen, “Unsupervised online activity discovery using temporal behaviour assumption,” in *Proceedings of ISWC '17*, 2017, pp. 42–49.

[6] J. Ye, L. Fang, and S. Dobson, “Discovery and recognition of unknown activities,” in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*. ACM, 2016, pp. 783–792.

[7] J. Yin, Q. Yang, and J. J. Pan, “Sensor-based abnormal human-activity detection,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 8, pp. 1082–1090, 2008.

[8] D. H. Hu, X.-X. Zhang, J. Yin, V. W. Zheng, and Q. Yang, “Abnormal activity recognition based on hdp-hmm models.” in *IJCAI*, 2009, pp. 1715–1720.

[9] 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.

[10] D. M. Tax and R. P. Duin, “Support vector data description,” *Machine learning*, vol. 54, no. 1, pp. 45–66, 2004.

[11] J. H. Shin, B. Lee, and K. S. Park, “Detection of abnormal living patterns for elderly living alone using support vector data description,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 3, pp. 438–448, 2011.

[12] H.-T. Cheng, M. Griss, P. Davis, J. Li, and D. You, “Towards zero-shot learning for human activity recognition using semantic attribute sequence model,” in *Ubicomp ’13*. ACM, 2013, pp. 355–358.

[13] T. Huynh, M. Fritz, and B. Schiele, “Discovery of activity patterns using topic models,” in *Proceedings of the 10th international conference on Ubiquitous computing*. ACM, 2008, pp. 10–19.

[14] L. Peng, L. Chen, X. Wu, H. Guo, and G. Chen, “Hierarchical complex activity representation and recognition using topic model and classifier level fusion.” *IEEE transactions on bio-medical engineering*, 2016.

[15] K. Farrahi and D. Gatica-Perez, “Discovering routines from large-scale human locations using probabilistic topic models,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 1, p. 3, 2011.

[16] K. Rieping, G. Englebienne, and B. Kröse, “Behavior analysis of elderly using topic models,” *Pervasive and Mobile Computing*, vol. 15, pp. 181 – 199, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574119214001308

[17] 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.

[18] D. J. Cook, N. C. Krishnan, and P. Rashidi, “Activity discovery and activity recognition: A new partnership,” *IEEE transactions on cybernetics*, vol. 43, no. 3, pp. 820–828, 2013.

[19] D. J. Cook, A. S. Crandall, B. L. Thomas, and N. C. Krishnan, “Casas: A smart home in a box,” *Computer*, vol. 46, no. 7, 2013.

[20] V. I. Levenshtein, “Binary codes capable of correcting deletions, insertions and reversals,” in *Soviet physics doklady*, vol. 10, 1966, p. 707.

[21] 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

[22] 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.

[23] S. Gopal and Y. Yang, “Von Mises-Fisher clustering models,” in *International Conference on Machine Learning*, 2014, pp. 154–162.

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

[25] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[26] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[27] K. Hornik and B. Grün, “movMF: An R package for fitting mixtures of von Mises-Fisher distributions,” *Journal of Statistical Software*, vol. 58, no. 10, pp. 1–31, 2014.

[28] T. Hastie, R. Tibshirani, and J. Friedman, “The elements of statistical learning,” *Springer Series in Statistics*, 2009. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-84858-7

[29] G. Malsiner-Walli, S. Frühwirth-Schnatter, and B. Grün, “Identifying mixtures of mixtures using bayesian estimation,” *Journal of Computational and Graphical Statistics*, vol. 26, no. 2, pp. 285–295, 2017.

[30] 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

[31] B. Settles, “Active learning literature survey,” *University of Wisconsin, Madison*, vol. 52, no. 55-66, p. 11, 2010.

[32] T. Scheffer, C. Decomain, and S. Wrobel, “Active hidden markov models for information extraction,” in *International Symposium on Intelligent Data Analysis*. Springer, 2001, pp. 309–318.

[33] C. E. Shannon, “A mathematical theory of communication,” *Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[34] S. C. Johnson, “Hierarchical clustering schemes,” *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.

[35] 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.

[36] B. Logan, J. Healey, M. Philipose, E. M. Tapia, and S. Intille, “A long-term evaluation of sensing modalities for activity recognition,” in *Proceedings of the 9th International Conference on Ubiquitous Computing*, ser. UbiComp ’07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 483–500. [Online]. Available: http://dl.acm.org/citation.cfm?id=1771592.1771620

[37] N. J. Cox *et al.*, “Speaking stata: in praise of trigonometric predictors,” *Stata Journal*, vol. 6, no. 4, pp. 561–579, 2006.

[38] 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

[39] P. Morel, “Gramm: grammar of graphics plotting in matlab,” vol. 3, p. 568, 03 2018.

[40] “Matlab statistics and machine learning toolbox,” 2018a, the MathWorks, Natick, MA, USA. [Online]. Available: https://uk.mathworks.com/products/statistics.html

[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.

[42] 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.

Lei Fang is a research fellow in the School of Computer Science at the University of St Andrews. He works on wireless sensor networks, adaptive pervasive systems, statistical modelling, uncertainty reasoning, data mining and machine learning. He holds a BSc (1st Hons) from the University of Liverpool and a PhD from the University of St Andrews.

Juan Ye is a lecturer in the School of Computer Science at the University of St Andrews. Her research interests centre around adaptive pervasive systems, specialising in sensor-based human activity recognition, sensor fusion, context awareness, ontologies, and uncertainty reasoning. Ye has a PhD in computer science from University College Dublin.

Simon Dobson is Professor of Computer Science in the School of Computer Science at the University of St Andrews. He works on complex and sensor systems, especially on sensor analytics and the modelling of complex processes. His research has generated over 150 internationally peer-reviewed publications, driven by leadership roles in research grants worth over EUR30M, most recently as part of a £ 5M EPSRC-funded programme grant in the Science of Sensor Systems Software. He has served, amongst other activities, as programme and general chairs for the IEEE International Conference on Autonomic Computing; as an associate editor of ACM Transactions on Autonomous and Adaptive Systems; as a member of UKCRC, the expert committee on UK computing research; and on the programme committees of a wide range of leading international conferences and specialised workshops. He holds a BSc from the University of Newcastle upon Tyne and DPhil from the University of York, both in computer science, is a Chartered Fellow of the British Computer Society, a Chartered Engineer and Senior Member of the IEEE and ACM.