Sensor-based Activity Recognition via Learning from Distributions

Hangwei Qian[†], **Sinno Jialin Pan**[‡], **Chunyan Miao**[‡]
[†] Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly, Interdisciplinary Graduate School, Nanyang Technological University, Singapore
[‡] School of Computer Science and Engineering, Nanyang Technological University, Singapore
qian0045@e.ntu.edu.sg, {sinnopan, ascymiao}@ntu.edu.sg

Abstract

Sensor-based activity recognition aims to predict users' activities from multi-dimensional streams of various sensor readings received from ubiquitous sensors. To use machine learning techniques for sensor-based activity recognition, previous approaches focused on composing a feature vector to represent sensor-reading streams received within a period of various lengths. With the constructed feature vectors, e.g., using predefined orders of moments in statistics, and their corresponding labels of activities, standard classification algorithms can be applied to train a predictive model, which will be used to make predictions online. However, we argue that in this way some important information, e.g., statistical information captured by higher-order moments, may be discarded when constructing features. Therefore, in this paper, we propose a new method, denoted by SMM_{AR}, based on learning from distributions for sensor-based activity recognition. Specifically, we consider sensor readings received within a period as a sample, which can be represented by a feature vector of infinite dimensions in a Reproducing Kernel Hilbert Space (RKHS) using kernel embedding techniques. We then train a classifier in the RKHS. To scale-up the proposed method, we further offer an accelerated version by utilizing an explicit feature map instead of using a kernel function. We conduct experiments on four benchmark datasets to verify the effectiveness and scalability of our proposed method.

Introduction

Recently, activity recognition has become one of the most crucial techniques in many real-world applications, such as healthcare, smart homes, security (Lara and Labrador 2013; Bulling *et al.* 2014; Frank *et al.* 2010; Ravi *et al.* 2005). The goal of activity recognition is to classify streams of sensor readings received from different types of sensors with the use of artificial intelligence (Bulling *et al.* 2014; Mannini and Sabatini 2010). In general, approaches to activity recognition can be classified into two categories: sensor-based and vision-based (cameras can also be considered as a special type of sensors) (Chen *et al.* 2012). In this work, we focus on wireless sensor-based activity recognition.

To build a recognition model from raw sensor readings to high-level activities, it mainly consists of two steps. The first step is to segment continuous streaming sensor readings automatically or manually (Yin *et al.* 2005; Janidarmian *et al.* 2017). Each segment contains sensor readings received from a set of sensors in a specific period of various lengths, and is supposed to correspond to one activity category. After that, the second step is to learn a predictive model to map each segment to its corresponding activity label. This is referred to as a multivariate time series classification problem. In this work, we assume segments of the streaming sensor data is prepared beforehand (yet the number of frames[1] of each segment can be different), and focus on solving the time series classification problem for activity recognition. In this context, as each segment is multi-dimensional and of various lengths, a key research issue is how to construct a feature vector to represent each segment because conventional classification algorithms are vector based (Lockhart and Weiss 2014), i.e., an input, either training or test instance, to a classification algorithm needs to be a feature vector of *fixed* dimensionality.

A simple solution to address the aforementioned research issue is to consider each individual frame of a segment as an instance, i.e., a vector of readings received from a fixed set of sensors at a particular time stamp, and assign each frame a label as the activity category of the segment. In this way, conventional classification algorithms can be performed in the frame-level instead of the segment-level to train a classifier. For instance, suppose only one sensor is used, whose frequency is set to be 1Hz, and a segment, whose activity label is "walking upstairs", lasts 5 seconds, which means that 5 frames are recorded. Frame-level approaches assign the activity label "walking upstairs" to each frame of the segment, and consider each framework as an individual instance. An alterative solution is to aggregate all the frames within a segment to generate a single feature vector. For example, an average vector of all the frames in a segment can be used to represent the segment. Consider the "walking upstairs" example. One can use the average vector of the 5 frames to represent the whole segment. In the past, one of the most widely used feature extraction approaches is to calculate statistical metrics, e.g., mean, variance, etc.,

Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[1]The term "frame" is often used in vision-based cases. In this paper, a frame is a vector of sensor readings from multiple sensors at a particular timestamp. A segment contains multiple frames.

from the raw sensor data of a segment (Plötz *et al.* 2011; Lockhart and Weiss 2014).

However, both of the aforementioned solutions fail to retain all the important information underlying a segment of sensor readings while constructing a feature vector. In the first solution, each frame is considered as an individual instance, and cannot fully represent the whole activity. In the second solution, one needs to predefine what statistical metrics, e.g., what orders of moments, are used, which is difficult to determine in practice. For example, if the mean vector is used to represent a segment corresponding to "walking upstairs", then it may be similar to that of another activity like "walking downstairs". Note that most classification algorithms are distance or similarity based. If the feature representation fails to distinguish instances from different classes, it is difficult to learn a precise classifier. In this case, more statistical moments, such as variance or even higher-order moments, are required to construct features. However, how to decide what orders of moments to construct features that are able to effectively distinguish different activities is challenging. Intuitively, if each segment can be represented by *infinite* orders of moments, then the feature representation should be rich enough to distinguish instances between different classes. In this work, we offer a solution based on this motivation.

We first consider each segment as a data sample that follows an unknown probability distribution, and aim to extract features of each segment to capture sufficient statistical information. We then propose a novel method for time series classification with an application to activity recognition via kernel embedding. Specifically, with the kernel embedding technique (Smola *et al.* 2007; Schölkopf and Smola 2002), each segment or sample is mapped to an element in a Reproducing Kernel Hilbert Space (RKHS). A RKHS is a high-dimensional or even infinite-dimensional feature space, which is able to capture any order of moments of the probability distribution from which the sample is drawn. Therefore, each element in the RKHS can be considered as a feature vector of sufficient statistics for representing the corresponding time-series segment. Finally, with the new feature vectors in a RKHS, we cast the multivariate time series classification problem as a Support Measure Machines (SMM) formulation (Muandet *et al.* 2012; Muandet 2015), which is a new method proposed for learning problems on distributions.

However, similar to other kernel-based methods, our proposed kernel-embedding-based approach for activity recognition suffers from a scalability issue due to highly computational cost on calculation of a kernel matrix. There have been several approaches proposed to alleviate the computational cost of kernel methods, such as low-rank approximation of the Gram matrix (Bach and Jordan 2005), explicit finite-dimensional features for additive kernels (Maji *et al.* 2013), Nyström methods (Williams and Seeger 2000), and Random Fourier Features (RFF) (Rahimi and Recht 2007; Sriperumbudur and Szabó 2015). In this work, we adopt RFF to propose an accelerated version to deal with large-scale datasets.

The rest of this paper is organized as follows. We first review related work on feature extraction for activity recognition, and some preliminaries of our proposed method. After that we present our proposed method in detail, and report extensive experimental results on four benchmark datasets. Finally we conclude this work and point out some potential directions in the future.

Related Work and Preliminary

As mentioned in the previous section, feature extraction from each sensor readings segment of variate-length to generate a representative feature vector of fixed-length is crucial for sensor-based activity recognition. There are two types of feature extraction approaches in general: statistical and structural (Lara and Labrador 2013). Statistical approaches concatenate (hand-picked) statistical metrics, e.g., moments, to construct feature vectors. Structural approaches take into account the interrelationship among data. The ECDF approach (Hammerla *et al.* 2013; Plötz *et al.* 2011) leverages distributions' quantile function to preserve the overall shape of the distribution as well as the spatial positions; Lin *et al.* (2007) proposed SAX method to discrete data into symbolic strings to represent equal probability mass. Our proposed method is a unified framework that naturally embeds feature extraction and classification. The feature extraction component of our method extracts all orders of moments to form a concatenated feature vector in the RKHS, thus falls into the statistical category.

Kernel Embedding of Distributions

Given a sample $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ drawn from a probability distribution $\mathbb{P}$, where each instance $\mathbf{x}_i$ is of d dimensions. The technique of kernel embedding (Smola *et al.* 2007) for representing an arbitrary distribution is to introduce a mean map operation $\mu(\cdot)$ to map instances to a RKHS, $\mathcal{H}$, and to compute their mean in the RKHS as follows,

$$\boldsymbol{\mu}_{\mathbb{P}} := \mu(\mathbb{P}) = \mathbb{E}_{\mathbf{x}\sim\mathbb{P}}[\phi(\mathbf{x})] = \mathbb{E}_{\mathbf{x}\sim\mathbb{P}}[k(\mathbf{x}, \cdot)], \qquad (1)$$

where $\phi : \mathbb{R}^d \to \mathcal{H}$ is a feature map, and $k(\cdot, \cdot)$ is the kernel function induced by $\phi(\cdot)$. If the condition $\mathbb{E}_{\mathbf{x}\sim\mathbb{P}}(k(\mathbf{x}, \mathbf{x})) < \infty$ is satisfied, then $\boldsymbol{\mu}_{\mathbb{P}}$ is also an element in $\mathcal{H}$. It has been proven that if the kernel $k(\cdot, \cdot)$ is characteristic, then the mapping $\mu : \mathcal{P} \to \mathcal{H}$ is injective (Sriperumbudur *et al.* 2009). The injectivity indicates an arbitrary probability distribution $\mathbb{P}$ is uniquely represented by an element in a RKHS through the mean map. As each distribution can be mapped to $\mathcal{H}$, the operations defined in $\mathcal{H}$, such as inner product and distance measure, are capable of estimating similarity or distance between distributions.

In practice, an underlying probability distribution of a sample is unknown. One can use an unbiased empirical estimation to approximate the mean map as follows,

$$\hat{\boldsymbol{\mu}}_{\mathbb{P}} = \frac{1}{n}\sum_{i=1}^{n} \phi(\mathbf{x}_i) = \frac{1}{n}\sum_{i=1}^{n} k(\mathbf{x}_i, \cdot). \qquad (2)$$

Though in theory, the dimension of $\hat{\boldsymbol{\mu}}_{\mathbb{P}}$ is potentially infinite, by using the kernel trick, the inner product of two probability

distributions in a RKHS can be computed efficiently through a kernel function associated to the RKHS,

$$\langle \hat{\boldsymbol{\mu}}_{\mathbb{P}_x}, \hat{\boldsymbol{\mu}}_{\mathbb{P}_z} \rangle = \tilde{k}(\hat{\boldsymbol{\mu}}_{\mathbb{P}_x}, \hat{\boldsymbol{\mu}}_{\mathbb{P}_z}) = \frac{1}{n_x n_z} \sum_{i=1}^{n_x} \sum_{j=1}^{n_z} k(\mathbf{x}_i, \mathbf{z}_j), \quad (3)$$

where $\tilde{k}(\cdot,\cdot)$ is a linear kernel defined in the RKHS, n_x and n_z are the sizes of the samples $\mathbf{X}$ and $\mathbf{Z}$ drawn from $\mathbb{P}_x$ and $\mathbb{P}_z$, respectively. In general, $\tilde{k}(\cdot,\cdot)$ can be a nonlinear kernel defined as follows,

$$\tilde{k}(\hat{\boldsymbol{\mu}}_{\mathbb{P}_x}, \hat{\boldsymbol{\mu}}_{\mathbb{P}_z}) = \langle \psi(\hat{\boldsymbol{\mu}}_{\mathbb{P}_x}), \psi(\hat{\boldsymbol{\mu}}_{\mathbb{P}_z}) \rangle, \quad (4)$$

where $\psi(\cdot)$ is the associated feature mapping of the nonlinear kernel $\tilde{k}(\cdot,\cdot)$.

Random Fourier Features

Though the kernel trick helps to avoid computation on inner product between high-dimensional (or even infinite-dimensional) vectors, the resultant kernel matrix is still of expensively computational cost, especially when training data is large-scale. Random Fourier Features (Rahimi and Recht 2007) provide explicit relatively low-dimensional feature maps for shift-invariant kernels $k(\mathbf{x},\mathbf{x}') = k(\mathbf{x}-\mathbf{x}')$ based on the following theorem:

Theorem 1 (Bochner's Theorem (Rudin 2017))**.** *A continuous, shift-invariant kernel k is positive definite if and only if there is a finite non-negative measure $\mathbb{P}(\omega)$ on $\mathbb{R}^d$, such that $k(\mathbf{x}-\mathbf{x}') = \int_{\mathbb{R}^d} e^{i\omega^\top(\mathbf{x}-\mathbf{x}')} d\mathbb{P}(\omega) = \int_{\mathbb{R}^d \times [0,2\pi]} 2cos(\omega^\top \mathbf{x} + b) cos(\omega^\top \mathbf{x}' + b) d(\mathbb{P}(\omega) \times \mathbb{P}(b)) = \int_{\mathbb{R}^d} 2(cos(\omega^\top \mathbf{x}) cos(\omega^\top \mathbf{x}') + sin(\omega^\top \mathbf{x}) sin(\omega^\top \mathbf{x}')) d\mathbb{P}(\omega)$, where $\mathbb{P}(b)$ is a uniform distribution on $[0, 2\pi]$.*

The randomized feature map $\mathbf{z} : \mathbb{R}^d \to \mathbb{R}^D$ linearizes the kernel:

$$k(\mathbf{x},\mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle \approx \mathbf{z}(\mathbf{x})^\top \mathbf{z}(\mathbf{x}'), \quad (5)$$

where the inner product of explicit feature maps can uniformly approximate the kernel values without the kernel trick. The random Fourier features are generated by:

$$z_w(\mathbf{x}) = \sqrt{2} cos(w^\top \mathbf{x} + b) \quad (6)$$

where $w \sim p(w)$, which is $k(\cdot,\cdot)$'s Fourier transform distribution on $\mathbb{R}^D$, and b is sampled uniformly from $[0, 2\pi]$. Then $k(\mathbf{x},\mathbf{x}') = \mathbb{E}(z_w(\mathbf{x})^\top z_w(\mathbf{x}'))$ for all $\mathbf{x}$ and $\mathbf{x}'$. Such a relatively low-dimensional feature map enables the kernel machine to be efficiently solved by fast linear solvers, therefore enables kernel methods to handle large-scale datasets (Sriperumbudur and Szabó 2015).

The Proposed Methodology

Problem Statement

In our problem setting, we assume that segments have been prepared on streams of sensor readings in advance. Suppose given n segments, $\{\mathbf{X}_i\}_{i=1}^n$, for training, where $\mathbf{X}_i = [\mathbf{x}_{i1} \ ... \ \mathbf{x}_{in_i}] \in \mathbb{R}^{d \times n_i}$. Here, each column $\mathbf{x}_{ij} \in \mathbb{R}^{d \times 1}$ is a vector of signals received from d sensors at a time stamp, which is referred to as a frame in the segment, and n_i is the length of the i-th segment. Note that for different segment, the values of n_i can be different. Moreover, for training, each segment $\mathbf{X}_i$ is associated with a label $y_i \in Y$, where $Y = \{1, ..., L\}$ is a set of predefined activity categories. Our goal is to train a classifier f to map $\{\mathbf{X}_i\}$'s to $\{y_i\}$'s. For testing, given m segments $\{\mathbf{X}_i^*\}_{i=1}^m$ without corresponding labels, we use the trained classifier to make predictions.

Motivation and High-Level Idea

For most standard classification methods, the input is a feature vector of fixed dimensionality, and the output is a label. However, in our problem setting, the input $\mathbf{X}_i$ is a matrix. Moreover, for different segments i and j, the sizes of the matrices $\mathbf{X}_i$ and $\mathbf{X}_j$ can be different (have the same number of rows, but different number of columns). Therefore, standard classification methods cannot be directly applied. As discussed, a commonly used solution is to decompose the matrix $\mathbf{X}_i$ to n_i vectors or frames $\{\mathbf{x}_{ij}\}$'s, each of which is of d dimensions, and assign the same label y_i to each vector. In this way, for each segment, one can construct n_i input-output pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n_i}$. By combining such input-output pairs from all the segments, one can apply standard classification methods to train a classifier f. For testing, given a segment $\mathbf{X}_k^*$, we can first use the classifier to predict the labels of each feature vector $\mathbf{x}_{kj}^*$ in the segment, and use the majority class of $f(\mathbf{x}_{kj}^*)$'s as the predicted label for $\mathbf{X}_k^*$. A major drawback of this approach is that a single frame of a segment fails to represent an entire activity that lasts for a period of time.

Another approach is to aggregate the n_i frames of a segment $\mathbf{X}_i$ to generate a feature vector of fixed dimensionality to represent the segment. For example, one can use the mean vector $\bar{\mathbf{x}}_i = \sum_{j=1}^{n_i} \mathbf{x}_{ij} \in \mathbb{R}^{d \times 1}$ to represent a segment $\mathbf{X}_i$. This approach can capture some global information of a segment, but in practice, one needs to manually generate a very high-dimensional vector to fully capture useful information of each segment. For example, one may need to generate a set of vectors of different orders of moments for a segment, and then concatenate them to construct a unified feature vector to capture rich statistic information of the segment, which is computationally expensive.

Different from previous approaches, we consider each segment $\mathbf{X}_i$ as a sample of n_i instances drawn from an unknown probability $\mathbb{P}_i$, and all $\{\mathbb{P}_i\}_{i=1}^n \subseteq \mathcal{P}$, where $\mathcal{P}$ is the space of probability distributions. By borrowing the idea from kernel embedding of distributions, we can map all samples to a RKHS through a characteristic kernel, and then use a potentially infinite-dimensional feature vector to represent each sample, and thus each segment. As the kernel embedding with characteristic kernel is able to capture any order of moments of the sample, the feature vector is supposed to capture all statistical moments information of the segment. With the new feature representations for each segment in the RKHS, we can train a classifier with their corresponding labels in the RKHS for activity recognition.

Activity Recognition via SMM$_{AR}$

In this section, we present our method for activity recognition in detail. First, each segment or sample $\mathbf{X}_i$ is mapped to a RKHS with a kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ via an implicit feature map $\phi(\cdot)$, and represented by an element $\boldsymbol{\mu}_i$ in the RKHS via the mean map operation:

$$\boldsymbol{\mu}_i = \frac{1}{n_i} \sum_{p=1}^{n_i} \phi(\mathbf{x}_{ip}). \tag{7}$$

As a result, we have n pairs of input-output in the RKHS $\{(\boldsymbol{\mu}_1, y_1), ..., (\boldsymbol{\mu}_n, y_n)\}$. Then our goal is to learn a classifier $f : \mathcal{H} \to \tilde{\mathcal{H}}$ such that $f(\boldsymbol{\mu}_i) = y_i$ for $i = 1, ..., n$. Here $\tilde{\mathcal{H}} = \mathcal{H}$ if a linear kernel on $\{\boldsymbol{\mu}_i\}$'s is used, i.e., $\tilde{k}(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) = \langle \boldsymbol{\mu}_i, \boldsymbol{\mu}_j \rangle$. Otherwise, $\tilde{\mathcal{H}}$ is another RKHS if nonlinear kernel is used on $\{\boldsymbol{\mu}_i\}$'s, i.e., $\tilde{k}(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) = \langle \psi(\boldsymbol{\mu}_i), \psi(\boldsymbol{\mu}_j) \rangle$, where $\psi(\cdot)$ is a nonlinear feature map that induces the kernel $\tilde{k}(\cdot, \cdot)$.

By using the empirical risk minimization framework (Vapnik 1998), we aim to learn $f(\cdot)$ by solving the following optimization problem,

$$\min_f \frac{1}{n} \sum_{i=1}^{n} \ell(f(\boldsymbol{\mu}_i), y_i) + \lambda \|f\|_{\tilde{\mathcal{H}}}, \tag{8}$$

where $\ell(\cdot)$ is a data-dependent loss function, $\lambda > 0$ is the tradeoff parameter to control the impact of the regularization term $\|f\|_{\tilde{\mathcal{H}}}$ and the complexity of the solution, and $\tilde{\mathcal{H}}$ is a RKHS associated with the kernel $\tilde{k}(\cdot, \cdot)$. As proven in the representer theorem in (Muandet *et al.* 2012) that the functional $f(\cdot)$ can be represented by

$$f = \sum_{i=1}^{n} \alpha_i \psi(\boldsymbol{\mu}_i), \tag{9}$$

where $\alpha_i \in \mathbb{R}$. If a linear kernel is used for $\tilde{k}(\cdot, \cdot)$ on $\mathcal{P}$, then $\tilde{\mathcal{H}} = \mathcal{H}$, and (9) can be reduced as

$$f = \sum_{i=1}^{n} \alpha_i \boldsymbol{\mu}_i, \text{ where } \alpha_i \in \mathbb{R}. \tag{10}$$

By specifying (9) or (10) using the Support Vector Machines (SVMs) formulation[2], we reach the following optimization problem, which is known as Support Measure Machines (SMMs) (Muandet *et al.* 2012),

$$\begin{aligned} \min_f \quad & \frac{1}{2}\|f\|_{\tilde{\mathcal{H}}}^2 + C \sum_{i=1}^{n} \xi_i, \\ s.t. \quad & y_i f(\boldsymbol{\mu}_i) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \\ & 1 \leq i \leq n, \end{aligned} \tag{11}$$

where $\tilde{\mathcal{H}}$ is a RKHS associated with the kernel $\tilde{k}(\cdot, \cdot)$ on $\mathcal{P}$, $\{\xi_i\}_{i=1}^n$ are slack variables to absorb tolerable errors, and $C > 0$ is a tradeoff parameter. When the form of the kernels, $k(\cdot, \cdot)$ and $\tilde{k}(\cdot, \cdot)$, are specified[3], many optimization techniques developed for standard linear or nonlinear SVMs can be applied to solve the optimization problem of SMMs.

After the classifier $f(\cdot)$ is learned, given a test segment $\mathbf{X}_k^*$, one can first represent it using the mean map operation

$$\boldsymbol{\mu}_k^* = \frac{1}{n_k} \sum_{p=1}^{n_k} \phi(\mathbf{x}_{kp}^*),$$

and then use $f(\cdot)$ to make a prediction $f(\boldsymbol{\mu}_k^*)$. In the sequel, we denote this kernel-embedding-based method for activity recognition by SMM$_{AR}$.

R-SMM$_{AR}$ for Large-Scale Activity Recognition

Note that the technique of kernel embedding of distributions used in SMM$_{AR}$ makes a feature vector of each segment be able to capture sufficient statistics of the segment. This is very useful for calculating similarity or distance metric between segments. However, it needs to compute two kernels, one is for kernel embedding of the frames within each segment, and the other is for estimating similarity between segments. This makes SMM$_{AR}$ computationally expensive when the number of segments is large and/or the number of frames within each segment is large. To scale up SMM$_{AR}$, in this section, we present an accelerated version using Random Fourier Features to construct an explicit feature map instead of using the kernel trick.

To be specific, based on (7) and (5), the empirical kernel mean map on a segment $\mathbf{X}_i$ with explicit Random Fourier Features can be written by

$$\boldsymbol{\mu}_i = \frac{1}{n_i} \sum_{p=1}^{n_i} \mathbf{z}(\mathbf{x}_{ip}).$$

where $\boldsymbol{\mu}_i \in \mathbb{R}^D$. We aim to learn a classifier $f(\cdot)$ in terms of parameters $\mathbf{w}$. If $f(\cdot)$ is linear with respect to $\{\boldsymbol{\mu}_i\}$'s, then the form of $f(\cdot)$ can be parameterized as

$$f(\boldsymbol{\mu}_i) = \mathbf{w}^\top \boldsymbol{\mu}_i. \tag{12}$$

If $f(\cdot)$ is a nonlinear classifier, then it can be written as

$$f(\boldsymbol{\mu}_i) = \mathbf{w}^\top \tilde{\mathbf{z}}(\boldsymbol{\mu}_i), \tag{13}$$

where $\tilde{\mathbf{z}} : \mathbb{R}^D \to \mathbb{R}^{\tilde{D}}$ is another mapping of Random Fourier Features. (12) is a special case of (13) when $\tilde{\mathbf{z}}$ is an identity mapping. The resultant optimization problem is reformulated accordingly as follows,

$$\min_{\mathbf{w} \in \mathbb{R}^{\tilde{D}}} \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}^\top \tilde{\mathbf{z}}(\boldsymbol{\mu}_i), y_i) + \lambda \|\mathbf{w}\|_2^2. \tag{14}$$

As $\tilde{\mathbf{z}}(\cdot)$ is an explicit feature map, standard linear SVMs solvers can be applied to solve (14), which is much more efficient than solving (11). Accordingly, in the sequel, we denote this accelerated version of SMM$_{AR}$ with Random Fourier Features by R-SMM$_{AR}$.

[2]Note that one can also specify (9) or (10) using other loss functions, which result in different particular approaches.

[3]Recall that the kernel $k(\cdot, \cdot)$ is defined on $\{\mathbf{X}_i\}$'s to perform a mean map operation for generating $\{\boldsymbol{\mu}_i\}$'s, and the kernel $\tilde{k}(\cdot, \cdot)$ is defined on $\{\boldsymbol{\mu}_i\}$'s for final classification.

Experiments

In this section, we conduct comprehensive experiments on four real-world activity recognition datasets to evaluate the effectiveness and scalability of our proposed SMM$_{AR}$ and its accelerated version R-SMM$_{AR}$.

Datasets

Four benchmark datasets are used in our experiments. The overall statistics of the datasets are listed in Table 1.

Datasets	# Seg.	# En.	# Fea.	# C.	f	# Sub.
Skoda	1,447	68.8	60	10	14	1
WISDM	389	705.8	6	6	20	36
HCI	264	602.6	48	5	96	1
PS	1,614	4.0	9	6	50	4

Table 1: Statistics of the four datasets. Note that in the table, "Seg." denotes segments, "En." denotes average number of frames per segment, "Fea." denotes feature dimensions, "C." denotes classes, "f" denotes frequency in Hz (sampling rates of sensors may be various, but we assume the frequency of all sensors in a dataset is the same after preprocessing), and "Sub." denotes subjects.

Skoda (Stiefmeier *et al.* 2007) contains 10 gestures performed during car maintenance scenarios. 20 sensors are placed on the left and right arms of the subject. The features are accelerations of 3 spatial directions of each sensor. Each gesture is repeated about 70 times.

WISDM is collected using accelerometers built into phones (Kwapisz *et al.* 2010). A phone was put in each subject's front pants leg pockets. Six regular activities were performed, i.e., walking, jogging, ascending stairs, descending stairs, sitting and standing.

HCI focuses on variations caused by displacement of sensors (Förster *et al.* 2009). The gestures are arm movements with the hand describing different shapes, e.g., a pointing-up triangle, an upside-down triangle, and a circle. Eight sensors are attached to the right lower arm of each subject. Each gesture is recorded for over 50 repetitions, and each repetition for 5 to 8 seconds.

PS is collected by four smartphones on four body positions: (Shoaib *et al.* 2013). The smartphones are embedded with accelerometers, magnetometers and gyroscopes. Four participants were asked to conduct six activities for several minutes: walking, running, sitting, standing, walking upstairs and downstairs.

Evaluation Metric

We adopt the F_1 score as our evaluation metric. As the activity recognition datasets are imbalanced and of multiple classes, we adopt both micro-F_1 score (miF) and weighted macro-F_1 score (maF) to evaluation the performance of different methods. Note that the *Null* class is included during training and testing, and is always considered as a "negative" class when computing miF and maF. More specifically, miF is defined as follows,

$$\text{miF} = \frac{2 \times \text{precision}_{all} \times \text{recall}_{all}}{\text{precision}_{all} + \text{recall}_{all}},$$

where precision_{all} and recall_{all} are computed from the pooled contingency table of all the positive classes as follows,

$$\begin{aligned} \text{precision}_{all} &= \frac{\sum_i \text{TP}_i}{\sum_i \text{TP}_i + \sum_i \text{FP}_i}, \\ \text{recall}_{all} &= \frac{\sum_i \text{TP}_i}{\sum_i \text{TP}_i + \sum_i \text{FN}_i}, \end{aligned}$$

where i denotes the i-th class of a set of predefined activity categories (i.e., positive classes), and TP_i, FP_i, and FN_i denote true positive, false positive, and false negative with respect to i-th positive class, respectively. Different from miF, maF is defined as follows,

$$\text{maF} = \sum_i w_i \frac{2 \times \text{precision}_i \times \text{recall}_i}{\text{precision}_i + \text{recall}_i},$$

where w_i is the proportion of the i-th positive class.

Experimental Setup

In our experiments, each dataset is randomly split into training and testing sets using a ratio of $70\% : 30\%$. Missing values are replaced by the mean values of the certain class in the training data. PCA is conducted as preprocessing with 90% variance kept. All the results are reported by taking average values together with the standard deviation over 6 repeated experiments. We use SVMs as the base classifier, and LIBSVM (Chang and Lin 2011) for implementation. For overall comparisons between our proposed methods and baseline methods, we use the RBF kernel $k(x, x') = \exp(-\gamma\|x - x'\|^2)$. Note that in SMM$_{AR}$, we use RBF kernels for both kernel embedding within each segment and classifier learning over different segments. We will further investigate different choices of kernels in SMM$_{AR}$. We tune the kernel parameter γ as well as the tradeoff parameter C in LibSVM, and choose optimal parameter settings based on 5-fold cross-validation on the training set. We compare SMM$_{AR}$ with the following baseline methods.

Segment-based methods This type of methods aim to aggregate sensor-reading segments of variable-lengths into feature vectors of a fixed-length. In order to compare feature extraction methods, to minimize the impact of classifiers, SVM is chosen as the unique classifier for different feature extraction methods.

- **Moment-x**. All the frames in a segment is aggregated by extracting different orders of moments to concatenate a single feature vector to be fed to SVMs. We use Moment-x to denote up to x orders of moments (inclusive) are extracted to generate a feature vector.
- **ECDF-d**. ECDF-d extracts d descriptors per sensor per axis. The range is set to $d \in \{5, 15, 30, 45\}$ following the settings in (Hammerla *et al.* 2013).

Methods	Skoda		WISDM		HCI		PS	
	miF	maF	miF	maF	miF	maF	miF	maF
SMM$_{AR}$	**99.61±.24**	**99.60±.25**	55.87±2.66	56.09±3.03	**100±0**	**100±0**	**96.74±1.20**	**96.72±1.22**
Moment-1	92.46±1.97	92.39±2.01	38.30±4.10	44.63±12.22	91.35±2.28	91.32±2.33	93.90±.94	93.85±.93
Moment-2	92.27±1.47	92.14±1.49	52.55±1.46	57.21±7.22	96.47±.79	96.47±.77	95.95±.86	95.94±.86
Moment-5	94.49±1.66	94.45±1.70	57.31±5.91	**62.52±9.81**	97.76±.79	97.77±.78	93.31±.99	93.42±.93
Moment-10	95.24±.63	95.23±.64	**57.79±3.97**	62.44±8.02	98.72±.79	98.72±.79	91.93±1.44	92.00±1.36
ECDF-5	92.96±1.57	92.95±1.52	52.77±2.73	56.22±7.33	**100±0**	**100±0**	95.63±1.07	95.63±1.06
ECDF-15	93.62±1.34	93.60±1.36	54.01±3.09	57.47±7.65	**100±0**	**100±0**	93.97±.96	94.04±.97
ECDF-30	93.25±1.11	93.21±1.15	55.33±4.50	58.26±7.13	**100±0**	**100±0**	90.82±.53	91.05±.57
ECDF-45	92.20±1.07	92.20±1.13	53.46±2.84	57.77±7.02	**100±0**	**100±0**	87.15±1.32	87.23±1.59
SAX-3	94.54±1.28	94.48±1.21	32.90±1.47	23.62±1.81	21.15±0	7.39±0	50.28±2.40	41.30±3.89
SAX-6	96.13±1.57	96.10±1.55	35.49±3.11	28.77±2.82	21.15±0	7.39±0	52.95±2.54	46.86±.68
SAX-9	97.36±1.33	97.31±1.34	32.43±1.16	23.84±1.61	21.15±0	7.39±0	51.70±1.14	43.58±1.52
SAX-10	96.22±.84	96.18±.83	32.57±1.48	26.89±2.39	21.15±0	7.39±0	52.81±1.08	44.60±1.52
miFV	61.40±3.24	53.63±2.50	14.61±2.04	4.72±2.13	21.64±1.58	18.78±2.24	15.32±4.28	7.65±5.83
SVM-f	93.46±1.20	92.65±1.38	27.49±2.71	18.70±2.88	99.52±.53	99.52±.53	95.22±1.10	95.21±1.10
kNN-f	93.17±1.44	92.93±1.45	28.48±2.15	17.96±2.84	99.04±1.22	99.05±1.21	94.73±.65	94.72±.65

Table 2: Overall comparison results on the four datasets (unit: %). The perfect prediction on HCI lies in the fact that the large # En. from Table. 1. It means much more accurate record of each activity. WISDM has the same advantage, but the problem lies in the large # Sub., which greatly enlarges variance of each class, thus affects the prediction.

- **SAX-a**. Following the settings in (Lin *et al.* 2007), we set N to be the number of frames of the segment, n to be the dimension of features (thus no dimension reduction), alphabet_size $a \in \{3, ..., 10\}$.
- **miFV**. miFV (Wei *et al.* 2017) is a state-of-the-art multi-instance learning method. It treats each segment of frames as a bag of instances, and adopts Fisher kernel to transform each bag into a vector. We follow the parameter tuning procedure in (Wei *et al.* 2017) with PCA energy set to 1.0 and the number of centers from 1 to 10.

Frame-based methods This type of methods consider each frame as an individual instance, whose class label is as the same as the corresponding segment's.

- **SVM-f** apply a SVM on frame-level data.
- **KNN-f** apply a kNN classifier on frame-level data, where the value of k is tuned in the range of $\{1, ..., 10\}$.

Overall Experimental Results

The overall comparison results of proposed methods along with all the baseline methods are presented in Table 2. As can be seen from the table, on average, the performance of SMM$_{AR}$/Moment-x/ECDF-d methods are much more stable than that of other methods. For example, SAX-a methods perform very well on Skoda, but perform very poor on all the other datasets. And our proposed SMM$_{AR}$ performs best on three out of four datasets. This illustrates the effectiveness of using kernel embedding technique to generate feature vectors in a RKHS for capturing any order of moments of a segment. Moreover, we can also observe from the table that in general, SVMs trained on feature vectors that contain more moment information perform better. For instance, on average, Moment-10 > Moment-5 > Moment-2 > Moment-1 on the datasets Skoda, WISDM, and HCI. One might notice that miFV performs very poor on all the four datasets. The reason is that it's not robust enough with respect to imbalanced class and the Null class interruption in the activity data. If the activity data is arranged into a balanced manner, the performances of miFV improve about 10%. If the Null class is removed, the performances improve about 30%.

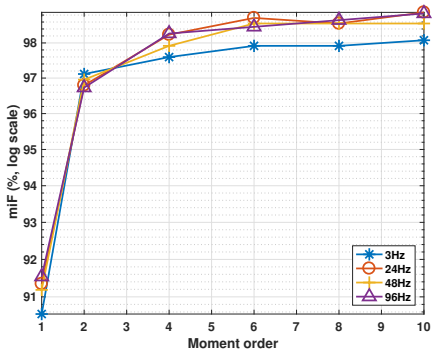

Figure 1: Comparison results of Moment-x in terms of miF on HCI by varying moments and frequencies.

Impact on Orders of Moments

To further investigate impact of different orders of moments to be used for constructing feature vectors on activity recognition, we conduct experiments on HCI as shown in Fig. 1. In the figure, different curve denotes different sampling frequency on sensor readings, which results in different numbers of frames per segment on average. The x-axis indicates up to what orders of moments are used. Though the recognition results are more or less effected by using different sampling frequencies on sensor readings, their increasing trends with more orders of moments are the same. These favourably prove our idea that incorporating more moment information in the feature vectors benefits the activity recog-

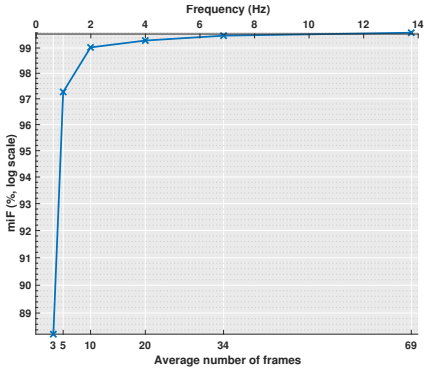

Figure 2: The miF performance on Skoda under different sampling frequencies and different average numbers of frames for each segment. The x-axis on the top and the x-axis are relevant as a lower sampling frequency on sensor readings leads to a smaller number of frames per segment.

nition performance. Hence the proposed method is likely to perform the best since all orders of moments information is utilized in the proposed method.

Impact of Sampling Frequency on Sensor Readings

Maurer *et al.* (2006) found that when increasing the sampling frequency, there is no significant gain in accuracy above 20Hz for activities. Here, we conduct experiments to analyze the impact of sampling frequency on the classification performance of SMM$_{AR}$. Fig. 2 shows the miF performance of SMM$_{AR}$ on Skoda under different sampling rates varying from 0.5Hz to 14Hz, resulting in average numbers of frames per segment varying from 3 to 68. The classification performance increases with larger average number of frames per segment, then becomes stable between 10 to 70 frames/segment. Therefore, our suggestion is that to use SMM$_{AR}$ for activity recognition, each segment needs to contain 10 or more frames, which is reasonable in practice.

Impact on Different Choices of Kernels

In SMM$_{AR}$, there are two types of kernels: $k(\cdot,\cdot)$ for kernel embedding within each segment (3) and $\tilde{k}(\cdot,\cdot)$ for training a nonlinear classifier (4). In this section, we conduct experiments to investigate the impact of different combinations of kernels on the final classification performance of SMM$_{AR}$. The results are shown in Table 3, where linear kernel (LIN), polynomial kernel of degree 3 (POLY3), RBF kernel and sigmoid kernel (SIG) are used. When SMM$_{AR}$ uses the RBF kernel for both $k(\cdot,\cdot)$ and $\tilde{k}(\cdot,\cdot)$, it performs best. Moreover, when the sigmoid kernel is used for kernel embedding, SMM$_{AR}$ performs worst. This may be because sigmoid kernel is not positive semi-definite, thus not characteristic, which may not be able to capture sufficient statistics for each segment (or sample).

Experimental Results on R-SMM$_{AR}$

In our final series of experiments, we test the scalability and effectiveness of our proposed accelerated version R-SMM$_{AR}$. Figure 3 illustrates the trends of performance and runtime with increasing sizes of random feature dimension D, respectively. The experiments are conducted on a Linux computer with Intel(R) Core(TM) i7-4790S 3.20GHz CPU. The runtime in seconds shown in the figure is the total runtime in both training and testing. As can be seen that with the increase of D, the runtime of R-SMM$_{AR}$ increases accordingly, and performance in terms of miF becomes higher. Note that the best performance of SMM$_{AR}$ in terms of miF on Skoda is 99.61%, with runtime of 264 seconds. R-SMM$_{AR}$ is able to achieve a comparable miF score with small standard deviation when $10 \leq D \leq 40$, while requires much less runtime. Therefore, compared with SMM$_{AR}$, R-SMM$_{AR}$ is an efficient and effective approximation approach, which is suitable for large-scale datasets. It saves a large proportion of runtime, and at the mean time, achieves comparable performance.

		$\tilde{k}(\cdot,\cdot)$			
		LIN	POLY3	RBF	SIG
$k(\cdot,\cdot)$	LIN	91.4300	91.3852	91.3632	28.6446
	POLY3	98.1202	98.0728	98.1556	92.0938
	RBF	98.1422	90.8818	**98.8950**	98.3728
	SIG	87.7026	87.0830	90.4140	90.4176

Table 3: Comparison performance in terms of miF of SMM$_{AR}$ on Skoda with different combinations of kernels.

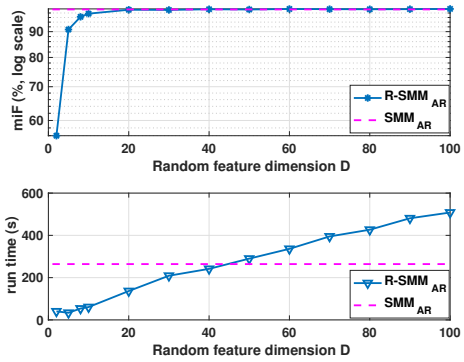

Figure 3: Comparison results between SMM$_{AR}$ and R-SMM$_{AR}$ in terms of runtime and miF score on Skoda.

Conclusion and Future Work

In this paper, we propose a novel solution, named SMM$_{AR}$, to extract all statistical moments of the activity data. This is the very first work to apply the idea of kernel embedding in the context of activity recognition problems. We conduct extensive experiments and demonstrate the effectiveness of SMM$_{AR}$ compared with a number of baseline methods. Moreover, we also present an accelerated version R-SMM$_{AR}$ to solve large-scale problems. In the future, besides statistical information, we plan to extend the proposed method to capture temporal information of each segment for learning feature representation of each segment.

Acknowledgements

This research is supported by the National Research Foundation, Prime Ministers Office, Singapore under its IDM Futures Funding Initiative and the Interdisciplinary Graduate School (IGS), Nanyang Technological University. Sinno J. Pan thanks the support from the NTU Singapore Nanyang Assistant Professorship (NAP) grant M4081532.020.

References

Francis R. Bach and Michael I. Jordan. Predictive low-rank decomposition for kernel methods. In *ICML*, pages 33–40, 2005.

Andreas Bulling, Ulf Blanke, and Bernt Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Comput. Surv.*, 46(3):33:1–33:33, 2014.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol*, 2(3):27:1–27:27, 2011.

Liming Chen, Jesse Hoey, Chris D. Nugent, Diane J. Cook, and Zhiwen Yu. Sensor-based activity recognition. *IEEE Trans. Systems, Man, and Cybernetics, Part C*, 42(6):790–808, 2012.

Kilian Förster, Daniel Roggen, and Gerhard Tröster. Unsupervised classifier self-calibration through repeated context occurences: Is there robustness against sensor displacement to gain? In *ISWC*, pages 77–84, 2009.

Jordan Frank, Shie Mannor, and Doina Precup. Activity and gait recognition with time-delay embeddings. In *AAAI*. AAAI Press, 2010.

Nils Y. Hammerla, Reuben Kirkham, Peter Andras, and Thomas Ploetz. On preserving statistical characteristics of accelerometry data using their empirical cumulative distribution. In *ISWC*, pages 65–68, 2013.

Majid Janidarmian, Atena Roshan Fekr, Katarzyna Radecka, and Zeljko Zilic. A comprehensive analysis on wearable acceleration sensors in human activity recognition. *Sensors*, 17(3):529, 2017.

Jennifer R. Kwapisz, Gary M. Weiss, and Samuel Moore. Activity recognition using cell phone accelerometers. *SIGKDD Explorations*, 12(2):74–82, 2010.

Oscar D. Lara and Miguel A. Labrador. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys and Tutorials*, 15(3):1192–1209, 2013.

Jessica Lin, Eamonn J. Keogh, Li Wei, and Stefano Lonardi. Experiencing SAX: a novel symbolic representation of time series. *Data Min. Knowl. Discov.*, 15(2):107–144, 2007.

Jeffrey W. Lockhart and Gary M. Weiss. Limitations with activity recognition methodology & data sets. In *UbiComp*, pages 747–756, 2014.

Subhransu Maji, Alexander C. Berg, and Jitendra Malik. Efficient classification for additive kernel svms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):66–77, 2013.

Andrea Mannini and Angelo Maria Sabatini. Machine learning methods for classifying human physical activity from on-body accelerometers. *Sensors*, 10(2):1154–1175, 2010.

Uwe Maurer, Asim Smailagic, Daniel P. Siewiorek, and Michael Deisher. Activity recognition and monitoring using multiple sensors on different body positions. In *BSN*, pages 113–116, 2006.

Krikamol Muandet, Kenji Fukumizu, Francesco Dinuzzo, and Bernhard Schölkopf. Learning from distributions via support measure machines. In *NIPS*, pages 10–18, 2012.

K. Muandet. *From Points to Probability Measures: A Statistical Learning on Distributions with Kernel Mean Embedding*. PhD thesis, University of Tübingen, Germany, September 2015.

Thomas Plötz, Nils Y. Hammerla, and Patrick Olivier. Feature learning for activity recognition in ubiquitous computing. In *IJCAI*, pages 1729–1734, 2011.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *NIPS*, pages 1177–1184, 2007.

Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, and Michael L. Littman. Activity recognition from accelerometer data. In *AAAI*, pages 1541–1546. AAAI Press / The MIT Press, 2005.

Walter Rudin. *Fourier analysis on groups*. Courier Dover Publications, 2017.

Bernhard Schölkopf and Alexander Johannes Smola. *Learning with Kernels: support vector machines, regularization, optimization, and beyond*. 2002.

Muhammad Shoaib, Hans Scholten, and Paul J. M. Havinga. Towards physical activity recognition using smartphone sensors. In *UIC/ATC*, pages 80–87, 2013.

Alexander J. Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A hilbert space embedding for distributions. In *ALT*, pages 13–31, 2007.

Bharath K. Sriperumbudur and Zoltán Szabó. Optimal rates for random fourier features. In *NIPS*, pages 1144–1152, 2015.

Bharath K. Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Gert R. G. Lanckriet, and Bernhard Schölkopf. Kernel choice and classifiability for RKHS embeddings of probability distributions. In *NIPS*, pages 1750–1758, 2009.

Thomas Stiefmeier, Daniel Roggen, and Gerhard Tröster. Fusion of string-matched templates for continuous activity recognition. In *ISWC*, pages 41–44, 2007.

Vladimir Vapnik. *Statistical learning theory*. Wiley, 1998.

Xiu-Shen Wei, Jianxin Wu, and Zhi-Hua Zhou. Scalable algorithms for multi-instance learning. *IEEE Trans. Neural Netw. Learning Syst.*, 28(4):975–987, 2017.

Christopher K. I. Williams and Matthias W. Seeger. Using the nyström method to speed up kernel machines. In *NIPS*, pages 682–688, 2000.

Jie Yin, Dou Shen, Qiang Yang, and Ze-Nian Li. Activity recognition through goal-based segmentation. In *AAAI*, pages 28–34, 2005.