Sensor-based activity recognition via learning from distributions

Qian, Hangwei

2019

Qian, H. (2019). Sensor-based activity recognition via learning from distributions. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/137691>

<https://doi.org/10.32657/10356/137691>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 19:02:14 SGT

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

**SENSOR-BASED ACTIVITY
RECOGNITION VIA LEARNING FROM
DISTRIBUTIONS**

HANGWEI QIAN

Interdisciplinary Graduate School

Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly

SENSOR-BASED ACTIVITY RECOGNITION VIA LEARNING FROM DISTRIBUTIONS

HANGWEI QIAN

Interdisciplinary Graduate School

Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2019

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

July. 28, 2019

.....
Date

Hangwei Qian

.....
Hangwei Qian

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

July. 28, 2019

.....
Date

.....
Sinno Jialin Pan

Authorship Attribution Statement

This thesis contains material from 3 papers published in the following peer-reviewed conferences as well as 1 paper submitted to a peer-reviewed journal where I was the first and/or corresponding author.

Chapter 4 is published as Qian, Hangwei, Pan, Sinno Jialin, and Miao, Chunyan. "Sensor-Based Activity Recognition via Learning from Distributions." Thirty-Second AAAI Conference on Artificial Intelligence. 6262-6269 (2018).

The contributions of the co-authors are as follows:

- Prof. Pan and Prof. Miao provided the initial research direction.
- I wrote the manuscript draft. The draft was revised by Prof. Pan.
- I co-designed the experimental study with Prof. Pan, and performed all the laboratory work at the School of Computer Science and Engineering and LILY Lab. I also analyzed the data and experimental results.
- I developed and released the code.

Chapter 5 is published as Qian, Hangwei, Pan, Sinno Jialin, Da, Bingshui and Miao, Chunyan. "A Novel Distribution-Embedded Neural Network for Sensor-Based Activity Recognition." Twenty-Eighth International Joint Conference on Artificial Intelligence. 2019.

The contributions of the co-authors are as follows:

- Prof. Pan and I discussed the initial research direction.
- I wrote the drafts of the manuscript. The manuscript was revised together with Prof. Pan and Mr. Da.
- I designed the experimental study, and performed all the laboratory work at the School of Computer Science and Engineering and LILY Lab.

- I developed the code and conducted experimental study with suggestions provided by Mr. Da. I analyzed the performance of the proposed method compared with baseline methods.
- Prof. Miao provided helpful reading materials.

Chapter 6 is published as Qian, Hangwei, Pan, Sinno Jialin, and Miao, Chunyan. "Distribution-based Semi-Supervised Learning for Activity Recognition." Thirty-Third AAAI Conference on Artificial Intelligence. 2019.

The contributions of the co-authors are as follows:

- Prof. Pan and I discussed the initial research direction.
- I wrote the drafts of the manuscript. The manuscript was revised together with Prof. Pan.
- I developed the code and conducted experimental study with suggestions provided by Prof. Pan. I performed all the laboratory work at the School of Computer Science and Engineering and LILY Lab. I analyzed the performance of the proposed method compared with baseline methods.
- Prof. Miao provided helpful reading materials.

Chapter 7 is published as Qian, Hangwei, Pan, Sinno Jialin, and Miao, Chunyan. "Weakly-Supervised Sensor-based Activity Segmentation and Recognition via Learning from Distributions." Submitted to Artificial Intelligence, 2019.

The contributions of the co-authors are as follows:

- Prof. Pan and I discussed the initial research direction.
- I wrote the drafts of the manuscript. The manuscript was revised together with Prof. Pan and Prof. Miao.
- I formulated the problem in a non-convex optimization problem. Prof. Pan assisted in refining the formulation.
- I developed the code and conducted experimental study with suggestions provided by Prof. Pan. I performed all the laboratory work at the School of

Computer Science and Engineering and LILY Lab. I analyzed the performance of the proposed method compared with baseline methods.

July. 28, 2019

.....

Date

Hangwei Qian

.....

Hangwei Qian

Abstract

Wearable-sensor-based activity recognition aims to predict users' activities from multi-dimensional streams of various sensor readings received from ubiquitous sensors. To utilize 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. However, we argue that the prevalent success of existing methods has two crucial prerequisites: proper feature extraction and sufficient labeled training data. The former is important to differentiate activities, while the latter is crucial to build a precise learning model. These two prerequisites have become bottlenecks to make existing methods more practical. Most existing feature extraction methods are highly dependent on domain knowledge, while labeled data requires intensive human annotation effort. In this thesis, we propose novel methods to tackle the above problems. The first crucial research issue is how to extract proper features from the partitioned segments of multivariate sensor readings. Both feature-engineering-based machine learning models and deep learning models have been explored for wearable-sensor-based human activity recognition. Existing methods have different drawbacks: 1) feature-engineering-based methods are able to extract meaningful features, such as statistical or structural information underlying the segments, but usually require manual designs of features for different applications, which is time consuming, and 2) deep learning models are able to learn temporal and/or spatial features from the sensor data automatically, but fail to capture statistical information.

To solve the problems, we firstly aim to extract statistical information captured by higher-order moments when constructing features. 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 R-SMM_{AR} by utilizing an explicit feature map instead of using a kernel function. Besides, we propose a novel deep learning model to automatically learn meaningful features including statistical features, temporal features and spatial correlation features for activity recognition in a unified framework.

The second research issue is how to alleviate the demand of sufficient training data problem. We propose a novel method, named Distribution-based Semi-Supervised Learning (DSSL for short), to tackle the aforementioned limitations. The proposed method is capable of automatically extracting powerful features with no domain knowledge required, meanwhile, alleviating the heavy annotation effort through semi-supervised learning. Specifically, we treat data stream of sensor readings received in a period as a distribution, and map all training distributions, including labeled and unlabeled, into a RKHS using the kernel mean embedding technique. The RKHS is further altered by exploiting the underlying geometry structure of the unlabeled distributions. Finally, in the altered RKHS, a classifier is trained with the labeled distributions. We also investigate the situation where only the coarse sequence of activity labels are known, while the starting and ending points of activities are unknown. We propose a unified weakly-supervised framework to jointly segment sensor streams and extract statistical features of sensory readings of each segment. We named our proposed algorithm S-SMM_{AR} . Extensive evaluations are conducted on various large-scale datasets to demonstrate the effectiveness of our proposed methods compared with state-of-the-art baselines.

Acknowledgements

Reaching the end of my PhD, I want to express my deepest gratitude to those who have helped me, encouraged me, and guided me during this bittersweet journey.

First and foremost, I am tremendously grateful for my supervisor, professor Sinno Jialin Pan, for his continuous guidance, support and inspiring discussions, and for providing me the freedom to learn and explore a variety of topics throughout my PhD. He is undoubtedly a profound scientist from whom I have learned critical ways of thinking. Thank you to my co-supervisors professor Chunyan Miao, Lihui Chen and my mentor Zhiqi Shen for their support, guidance and fruitful conversations. I am also grateful for the teaching assistant opportunities provided by Zhiqi Shen and Kevin Anthony Jones.

I am very happy to have had the opportunity to join a friendly and vibrant research group: Wenya Wang, Haiyan Yin, Yu Chen, Sulin Liu, Yaodong Yu, Zhengkun Yi, Yunxiang Liu, Jianjun Zhao, Long-Kai Huang, Qiang Zhou, Jianda Chen, Shangyu Chen, Tianze Luo, Zichen Chen, Disheng Dong, Jie Zhang, Jingliang Li, Chen Shao, etc. Those memories of group meetings and discussions, as well as group gatherings, are precious to me.

I am grateful to LILY Research Center and IGS for the financial support throughout my entire PhD study. It is great working with versatile colleagues: Hao Zhang, Yi Dong, Yanhai Xiong, Yong Liu, Chi Zhang, Qingyu Guo, Haipeng Chen, Peng Chen, Yong Liu, Han Yu, Jun Lin, Lei Meng, Qiong Wu, Huiguo Zhang, Benny Tan, Xu Guo, Peixiang Zhong, Chaoyue He, Zhiwei Zeng, Ashish Kumar, Chang Liu, Frank Yunqing Guan, Siyu Jiang, Shan Gao, Zhengjin Guo, Yang Qiu, Siyuan Liu, Xinjia Yu, Yuxi Guo, Liang Zhang, Robin Chan Chung Leung, Bo Huang, Di Wang, Rong Wang, Simon Fauvel, Jessica Hon-Chan, Xuejiao Zhao, Wei Wang, and many others.

Besides, I want to thank all my other friends, including but not limited to Jiebo Chen, Wen Peng, Weizhen Cai, Lei Zhang, Wenyu Zhang, Yang Cao, Feifei Chen, Min Zhou,

Yuehe Zhu, Bingbing Zhuang, Chenyin Liu, Dong Liu, Jiawei Liu, Yichen Zhang, Xiao Liu, Ziyu Liu, Dan Lu, Miaomiao Ma, Xiaoqian Mu, Peng Ni, Kun Ouyang, Haiyun Peng, Kai Qian, Ruidan He, Runtian Ren, Biao Sun, Saifei Sun, Wenchang Tang, Dongxia Wang, Jing Wang, Tianyi Wang, Xiaohong Wang, Xinrun Wang, Zhenkun Wang, Zhenyi Wang, Jin Xia, Cong Xie, Xiaofei Xu, Xin Xu, Haodan Yang, Yang Yang, Dongsen Ye, Changshen You, Li Yuan, Yuan Yuan, Yijie Zeng, Yongquan Zeng, Yiteng Zhai, Huaxin Chen, Qian Chen, Liang Zou, Zhuoxuan Jiang, Yichao Jin, Qiyu Kang, Youzhi Zhang, Chao Zhao, Hao Li, Haoliang Li, Jianshu Li, Jing Li, Zhaomin Chen, Shanshan Feng, Xin Zheng, Han Hu, Jing Tang, Liang Feng, Yaqing Hou, Yijing Li, Zhuo Chen, Peng Chen, Xi Cui, Daniel Han, Jiali Du, Mengchen Zhao, Lei Feng, Shixin Mao, Liuhao Ge, Jiuxiang Gu, Qing Guo, Xinting Hu, Jing Huang, Weiwei Huang, Zhu Sun, Xinghua Qu, Xiaowei Lou, for the joyful times throughout my life.

Finally, special thanks to my beloved family and boyfriend, for all the years of your unconditional love and support.

Contents

Abstract	iii
Acknowledgements	v
Contents	vii
List of Figures	xi
List of Tables	xiii
1 Introduction	1
1.1 Background	1
1.2 Challenges	3
1.2.1 Data Representation	3
1.2.2 Feature Extraction	4
1.2.3 Data and Label Availability	6
1.2.4 Segmentation	7
1.3 Thesis Contribution	9
1.4 Thesis Organization	10
2 Literature Review	13
2.1 Feature Extraction for Activity Recognition	13
2.1.1 Feature-Engineering-Based Feature Extraction	13
2.1.2 Deep-Learning-Based Feature Extraction	14
2.2 Learning with Partial Labels	16
2.3 Time Series Segmentation	17
3 Preliminaries	19
3.1 Kernel Methods in Machine Learning	19
3.2 Kernel Mean Embedding of Distributions	22
3.3 Approximating the Kernel Mean Embedding	25
3.4 Learning with Kernels	26
3.5 The Expectation Loss SVM (e-SVM) Method	27
4 Sensor-based Activity Recognition via Learning from Distributions	29
4.1 Overview	29

4.2	The Proposed Methodology	31
4.2.1	Problem Statement	31
4.2.2	Motivation and High-Level Idea	31
4.2.3	Activity Recognition via SMM_{AR}	32
4.2.4	R- SMM_{AR} for Large-Scale Activity Recognition	34
4.3	Experiments	35
4.3.1	Datasets	35
4.3.2	Evaluation Metric	37
4.3.3	Experimental Setup	37
4.3.3.1	Segment-based methods	40
4.3.3.2	Frame-based methods	40
4.3.4	Overall Experimental Results	41
4.3.5	Impact on Orders of Moments	41
4.3.6	Impact of Sampling Frequency on Sensor Readings	42
4.3.7	Impact on Different Choices of Kernels	43
4.3.8	Experimental Results on R- SMM_{AR}	44
4.4	Summary	44
5	A Novel Distribution-Embedded Neural Network for Sensor-Based Activity Recognition	47
5.1	Overview	47
5.2	The Proposed DDNN Model	49
5.2.1	The Overall Model	49
5.2.2	Statistical Module	50
5.2.3	Spatial Module	53
5.2.4	Temporal Module	54
5.3	Experiments	54
5.3.1	Datasets	54
5.3.2	Experimental Setup	55
5.3.3	Baselines	56
5.3.4	Experimental Results and Analysis	57
5.3.5	Impact of Spatial and Statistical Module	58
5.3.6	Robustness of the Proposed DDNN	58
5.3.7	Parameter's Sensitivity	59
5.4	Summary	60
6	Distribution-based Semi-Supervised Learning for Activity Recognition	61
6.1	Overview	61
6.2	The Proposed Methodology	63
6.2.1	Problem Statement	63
6.2.2	Distribution-based Semi-Supervised Learning	63
6.2.2.1	1) Construction of the Data-dependent Kernel $\check{k}$	64
6.2.2.2	2) Validity of $\check{\mathcal{H}}$	65
6.2.2.3	3) Loss Function Calculation	65

6.3	Detailed Proofs	67
6.3.1	Proof of Theorem 6.1	67
6.3.2	Proof of Proposition 1	68
6.3.3	Proof of Proposition 2	69
6.3.4	Proof of Theorem 6.2	70
6.4	Experiments	71
6.4.1	Datasets	71
6.4.2	Experimental Setup	72
6.4.3	Baselines	73
6.4.4	Experimental Results	74
6.4.4.1	Overall Experimental Results	74
6.4.4.2	Impact of Ratio of Labeled Data	76
6.4.4.3	Impact of Ratio of Unlabeled data	76
6.4.4.4	Impact of Parameter r	77
6.4.4.5	Impact on Random Fourier Feature (RFF) Dimension D	78
6.5	Summary	78
7	Weakly-Supervised Sensor-based Activity Segmentation and Recognition	81
7.1	Overview	81
7.2	The Proposed Methodology	83
7.2.1	Problem Statement	83
7.2.2	Problem Formulation in Weakly-Supervised Setting	83
7.2.3	Alternating Optimization for Joint Segmentation and Classifi- cation	85
7.2.3.1	Learning the classifier f with fixed $\mathbf{I}$ and $\mathbf{C}$	86
7.2.3.2	Update $\mathbf{I}$ and $\mathbf{C}$ with fixed f	88
7.2.4	R-SMM _{AR} for Large-Scale Activity Recognition	91
7.3	Experiments	92
7.3.1	Datasets	93
7.3.2	Evaluation Metric	94
7.3.3	Experiments for Segmentation	95
7.3.3.1	Experimental Setup	95
7.3.3.2	Baselines	95
7.3.3.3	Experimental results	96
7.3.4	Experiments for Joint Segmentation and Feature Extraction	97
7.3.4.1	Experimental Setup	97
7.3.4.2	Baselines	97
7.3.4.3	Experimental results	98
7.3.5	Experiments for Classification with Perfect Segmentation	98
7.4	Summary	99
8	Conclusions and Future Work	101
8.1	Conclusions	101

8.2 Future Work	102

Bibliography	105

List of Figures

1.1	Illustration of the hierarchical structure of the thesis.	11
2.1	Architecture Illustration of the CNN_Yang method.	15
2.2	Architecture Illustration of the state-of-the art baseline method Deep-ConvLSTM.	16
4.1	Comparison results of Moment- x in terms of miF on HCI dataset by varying moments and frequencies.	42
4.2	The miF performance on Skoda dataset 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.	43
4.3	Comparison results between SMM_{AR} and $R-SMM_{AR}$ in terms of run-time and miF score on Skoda dataset.	45
5.1	Illustration of the proposed DDNN architecture. The input to the network consists of a data sequence $\mathbf{X}_i = [\mathbf{x}_{i1} \dots \mathbf{x}_{iL}] = [\mathbf{x}_i^1 \dots \mathbf{x}_i^d]^T \in \mathbb{R}^{d \times L}$ extracted from d sensors and partitioned by sliding window approach with length L . From left to right, there are three modules for extracting spatial, temporal and statistical features respectively. Note that the input data format for these modules are different. Spatial correlations among sensors whose signals are represented as row vectors $\{(\mathbf{x}_i^r)^T\}_{r=1}^d$ are learned by LSTMs. Temporal dependencies are extracted from column vectors $\{\mathbf{x}_i^j\}_{j=1}^L$ by both LSTMs and CNNs (we will explain later why CNNs extract temporal dependencies instead of spatial correlations). Statistical module take the matrix form data $\mathbf{X}_i$ as inputs of autoencoder. All the learned features are then concatenated into a single feature vector, which is input to the fully-connected layers.	51
5.2	Illustration of performance difference with different weights put on the loss function ℓ_{MMD}	59
6.1	Impact of varying ratios of labeled data in semi-supervised learning. . .	76
6.2	Impact of varying ratios of unlabeled data in semi-supervised learning. .	77
6.3	Impact of r to the performance of proposed DSSL method.	77
6.4	Impact of D to the performance on WISDM in semi-supervised learning. .	78

List of Tables

4.1	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.	36
4.2	Overall comparison results on the four datasets (unit: %). The perfect prediction on HCI lies in the fact that the large # En. from Table. 4.1. 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.	39
4.3	Comparison performance in terms of miF of SMM_{AR} on Skoda with different combinations of kernels.	44
5.1	The overall information of the four datasets. Note that “# train”, “# val.” and “# test” refer to total number of training, validation and test samples, respectively. “#sw” denotes the sliding window length used in the experiments. UCIHAR is preprocessed and segmented beforehand by the data provider, which does not contain validation set.	55
5.2	Overall comparison results on the four datasets (unit: %). Note that the results of baselines with * are directly copied from [Morales and Roggen, 2016].	58
6.1	Notations of different kernels used in Chapter 6.	65
6.2	Statistics of datasets used in experiments of Chapter 6.	72
6.3	Experimental results of proposed semi-supervised methods as well as baselines on three activity datasets (unit: %).	74
6.4	Comparison results on drug activity prediction and image annotation tasks (unit: %).	75
7.1	Statistics of the four datasets for joint segmentation and classification. Note that in the table, “Seg.” denotes segments, “En.” denotes average number of frames per segment, “Fea.” denotes feature dimensions, “C.” denotes classes, “freq” 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), “Sub.” denotes subjects, and “ $\frac{\#Seg.}{\#C.}$ ” denotes the average number of segments that each class of activity has.	93

7.2	Overall comparison results of segmentation performance on the four datasets (unit: %). NaN indicates that the produced results are infeasible.	97
7.3	Overall comparison results on joint segmentation and feature extraction on four datasets (unit:%).	98

Chapter 1

Introduction

1.1 Background

Human activity recognition has spurred a great deal of interest with a wide spectrum of real-world applications, such as smart homes, security, personalized health monitoring and assisted living [Avci et al., 2010, Bulling et al., 2014, Cook et al., 2013, Frank et al., 2010, Janidarmian et al., 2017, Lara and Labrador, 2013, Ramamurthy and Roy, 2018, Shoaib et al., 2015, Wang et al., 2017]. The first works on human activity recognition date back to twenty years ago [Foerster et al., 1999]. It has recently received growing attention attributing to the intensive thrusts from the latest technology development and application demands. Over the past decade, sensor technologies, especially low-cost, high-capacity and miniaturized sensors have made substantial progress [Gao et al., 2016, Varatharajan et al., 2018, Yang and Sahabi, 2016, Zhu et al., 2015]. This allows people to interact with the sensor devices as part of the daily living. Particularly, the recognition of human activities has become an active task, especially for medical and security applications [Patel et al., 2012, Qi et al., 2018, Wang et al., 2019]. For instance, patients with dementia and other medical pathologies could be monitored to detect abnormal activities and thereby prevent undesirable consequences. Despite HAR being an active field for more than a decade, there are still key research issues that, if addressed, would benefit human daily life.

The recognition of human activities has generally been approached in two categories based on different types of involved sensors, namely external-sensor-based and wearable-sensor-based [Lara and Labrador, 2013]. External sensors are typically attached to objects in a smart home environment or fixed in points of interest. Human activities then can be inferred through interactions of the user with the sensors. Video cameras and radio frequency identifier (RFID) tags are commonly used as external sensors [Poppe, 2010, Simonyan and Zisserman, 2014]. There are sensors tracking the changes of environment as well, such as temperature sensors, WiFi, radar and sound sensors [Pan et al., 2007, Yang et al., 2008].

Wearable sensors, in contrast, are attached to the different body parts of the user. Wearable sensors usually contain accelerometers, magnetometers, and gyroscopes, which can often be found on smart phones, smart watches, helmets, etc. These sensors are worn by the participants, and the acceleration and angular velocity keep track of the body movements of participants.

One of the limitations of external sensors is that, nothing can be done if the user is not interacting with external sensors, or out of the scope of external sensors. Another concern is the privacy issues, especially for video cameras, where all the behaviours of participants are recorded and are easily recognized by others. Therefore, in this thesis, we focus on wearable-sensor-based activity recognition scenarios since the wearable sensors alleviate the environment constraints and the non-visionary signals are free from privacy issues [Yang et al., 2015].

To build a recognition model from raw sensor readings to high-level activities, it mainly consists of three steps. The first step is to segment continuous streaming sensor readings automatically or manually [Janidarmian et al., 2017, Yin et al., 2005]. 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. Usually fixed-size sliding window method is applied to segment raw signals into equal-length segments in previous works. After that, the second step is to conduct feature extraction on each segment. Finally, the extracted features are then fed into a classifier to recognize different activities [Hammerla et al., 2016]. This is referred to as a multivariate

time series classification problem. In the thesis, we conduct various aspects of research towards the above three steps, as claimed in the following.

1.2 Challenges

There are many issues that motivate the development of new techniques to activity recognition. For example, the data collection procedure, the selection of attributes to be measured, the construction of a portable and unobtrusive data acquisition system and the design of a flexible system to support new users. In this thesis, we consider the problem from the perspective of machine learning, and we focus on the following challenges along with the goal of improving the classification performance of activities: how to extract proper and sufficient features from raw data, how to deal with few labeled data and weakly labeled data, and how to properly segment the raw signals instead of brute-force sliding window method.

1.2.1 Data Representation

The data representation is a fundamental yet important issue for the human activity recognition problem. The raw data collected from wearable sensors can be treated in two ways, i.e., frame-level and segment-level. Here we use the term “frame” to denote a vector of sensor readings from multiple sensors at a particular timestamp. The raw data is composed of streaming frames whose frequency is determined by sensors’ sampling rates. A frame of data represents signals gathered in a specific timestamp. A segment contains multiple frames of data. Each activity can last for various numbers of frames since the duration of different activities can be different. Even different repetitions of a specific activity can last for various durations.

Most of classic algorithms only make use of information from individual data points, which are points in a vector space, and are drawn independent and identically (i.i.d.) from some unknown distribution. Often the grouping properties in a segment are neglected. In the thesis, we claim that representing the segment data as distributions over

such a vector space may be more preferable. We believe that probability distributions, as opposed to data points, contain more information about aggregate behaviour among the data. Probability distributions naturally model noisy observations and uncertain observations among data. The noise and uncertainty of data is inevitable due to the data collection, sensors errors, as well as data preprocessing. The variations of data among participants are inevitable as well, since different participants have their own styles of conducting activities.

1.2.2 Feature Extraction

Depending on the frame-level and segment-level data representations, there are different ways of feature extraction respectively. A simple solution for frame-level data representation 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. For frame-level data, each frame of raw data is paired with a label for the corresponding activity. Alternatively, for segment-level data, each segment of data is paired with a label for the entire containing frames. A corresponding 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. Among the existing literature, one of the most widely used feature extraction approaches is to manually design domain-specific features, and to calculate some basic statistical metrics, e.g., mean, variance, minimum, maximum, median, etc., from the raw sensor data of a segment [Lockhart and Weiss, 2014, Plötz et al., 2011].

However, both 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 thus cannot fully represent the entire 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 Chapter 4, we offer a solution based on this motivation.

Manual feature engineering is possible to be avoided based on the growing trend of representation learning with deep neural networks, which has demonstrated great performance in activity recognition [Morales and Roggen, 2016, Wang et al., 2017]. The raw data is segmented by fixed-size sliding window methods before being fed into deep learning models. Each layer of deep learning models except for the last layer can be considered as feature extractors at different levels. In computer vision, lower layers in deep models are considered as low-level feature extractors, while higher layers can extract more abstract and high-level features. Convolutional neural networks (CNNs) are the most widely used frameworks in this field [Ignatov, 2018, Yang et al., 2015, Zeng et al., 2014]. Besides, temporal dependencies in time-series data are proven to be beneficial for activity recognition as well. Recurrent neural networks (RNNs) and Long Short-Term Memory (LSTMs) are used for extracting temporal features along time scale [Morales and Roggen, 2016]. Besides, deep feed-forward networks (DNNs), and other networks can also be applied as feature extractors. It also works well in practice to stack several types of neural networks together as a combinational feature

extractor. One of the drawbacks of applying existing deep neural networks on the human activity recognition problem is that the networks are initially designed for images input, yet the data from wearable sensors enjoy different properties compared with images. In Chapter 5, we investigate the problem and propose a novel framework for the task of activity recognition.

1.2.3 Data and Label Availability

Supervised learning methods have been the mainstream to activity recognition [Bishop, 2006, Michie et al., 1994]. The prevalent success of existing methods, however, has a crucial prerequisite: sufficient labeled training data. To be specific, each training example has a label indicating its ground-truth label. Though supervised learning has been widely applied in the activity recognition, it is noteworthy that it is costly to collect the strong supervision information such as fully ground-truth labels. The labels of training data require intensive human annotation effort. What's worse, human annotation of activities are time-consuming, costly, and error-prone. Considering all these factors, it is desirable to use as few labeled data as possible during the training stage. However, limited labeled training data is insufficient to train a good classifier due to the cold start problem of supervised learning [Zhu, 2005].

To alleviate the human annotation effort, there are several potential solutions. The first solution is to use a few labeled training data, as well as a large amount of unlabeled data in a semi-supervised learning setting. Semi-supervised learning approaches are appealing in practice since they require only a small fraction of labeled training data with a large amount of easily obtained unlabeled data [Chapelle et al., 2010, Zhu, 2005]. Compared with supervised learning, semi-supervised learning is much less investigated in the scenario of human activity recognition [Lara and Labrador, 2013]. We propose a novel semi-supervised method to tackle the aforementioned limitations in Chapter 6.

Another solution is weakly-supervised learning [Zhou, 2017], where the human annotation on the training data does not have to be accurate and specific on all frames. Typically, there are three types of weak supervision. The first type is incomplete supervision, where a subset of training data are unlabeled. For example, semi-supervised

learning [Zhu, 2005] attempts to exploit unlabeled data in addition to a few annotated data. Active learning [Johnson and Johnson, 2008] assumes that there is a human expert to be queried to get ground-truth labels for some selected unlabeled data. The second type is inexact supervision, where only coarse-grained labels are given. For instance, multi-instance learning only contains labels for sets instead of each data point [Stikic et al., 2011, Zhou et al., 2009]. The last type is inaccurate supervision, where the given labels are not always ground-truth. All these weakly-supervised settings help to alleviate the labeling demands of training data. In chapter 7, we propose a novel weakly-supervised learning framework for activity recognition, where only the sequence of coarse labels are available, whereas the starting and ending positions of each activity are unknown.

1.2.4 Segmentation

The sensors' data is collected continuously while a participant performs different activities in free-living situations. Thereby, the duration of each activity can be various, and there are transition intervals between two adjacent activities. The goal of segmentation is to partition the time series data into continuous segments of variable lengths with changepoints or breakpoints in between. This is a crucial preprocessing step for sensor-based activity recognition, since a good segmentation is beneficial to learning an accurate activity classifier. However, segmentation of sensory streams of activity data is much less investigated compared with other time series data, such as financial or biological data. Most of the existing literature related to activity data focus on applying various machine learning techniques to improve the recognition accuracy of activities with the data segmentation step not optimized.

To partition continuous streaming activity data, existing approaches typically divide the entire sequence of sensor events into sliding windows with static or dynamic size(s). The difference between two adjacent windows is computed to be compared with some threshold to decide whether a breakpoint is found or not. However, how to identify the optimal window size remains an open problem [Baños et al., 2014]. Fixed-size sliding

window methods include non-overlapping and overlapping variants. One major drawback is that the duration of different activities are very likely to vary in the real-world settings. Thus, dynamic sliding window approaches are proposed to enable varying window sizes to segment the data by utilizing extra information, such as meta information, temporal information and multi-features [Ni et al., 2016, Shahi et al., 2017].

One alternative approach is to detect activity transitions or boundaries. Detecting activity breakpoints based on characteristics of observed sensor data can be formulated as changepoint detection problem. To successfully detect the breakpoints, it is often assumed that the data adheres to some degree of homogeneity. For instance, one of the most general parametric models assumes that the time series data consists of piecewise constant distributions. This formulation encompasses a large number of existing models on many scenarios including financial, medical and biological applications [Chen and Gupta, 2011, Maidstone et al., 2017]. Linear model assumptions also have been extensively utilized to model time series. Polynomial functions are complex variants of the parametric models [Fuchs et al., 2010]. For parametric models, the changepoints correspond to changes in the parameter(s). A major drawback of parametric models is the heavy reliance on the assumption that the data fits the predefined model. Hence, extra domain knowledge is usually required. What is worse is that it requires additional operations to test the feasibility of the selected parametric model. For activity data, it is actually improper to feed the activity data into parametric models due to the complexity of activity data. On the one side, multiple dimensions of activity data lead to more complex parametric models. On the other side, the variations among activities are non-negligible. Repetitions of the same activity may be quite different since distinct participants can have various activity patterns. Different from the above parametric models, nonparametric models need no prior knowledge on underlying distribution, thus can be used in a much wider variety of settings. In these models, data is mapped onto a higher-dimensional space and changepoints are detected by comparing the homogeneity of each subsequence. One drawback is that nonparametric methods are conducted in the unsupervised setting, which increases the computational complexity. In Chapter 7, we model the joint segmentation and classification problem as a non-convex problem and further propose a novel segmentation approach in weakly-supervised setting, which

is both efficient and effective.

1.3 Thesis Contribution

Overall, this thesis introduces research works on human activity recognition problem. The major research contributions of this dissertation are four-fold, listed as follows:

- 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, denoted by $R-SMM_{AR}$ by utilizing an explicit feature map instead of a kernel function. As far as we know, our work is the first attempt to explore the kernel mean embedding on the task of activity recognition.
- We further propose a Distribution-Embedded Neural Network (DDNN), which is a unified end-to-end trainable deep learning model. Different from previous deep learning models which are difficult to explain the extracted features, DDNN is able to learn three different types of powerful features for activity recognition in an automated fashion.
- To tackle the heavy annotation effort for labeling training data, we propose a novel method, named Distribution-based Semi-Supervised Learning (DSSL). The proposed method is capable of automatically extracting powerful features with no domain knowledge required, meanwhile, alleviating the heavy annotation effort through semi-supervised learning. Specifically, we treat data stream of sensor readings received in a period as a distribution, and map all training distributions, including labeled and unlabeled, into a RKHS using the kernel mean embedding technique. The RKHS is further altered by exploiting the underlying geometry structure of the unlabeled distributions. Finally, in the altered RKHS, a classifier is trained with the labeled distributions.

- We model the weakly-supervised segmentation problem of activity data as a non-convex optimization problem, and propose a novel iterative kernel-based method to solve it. The segmentation method together with a novel feature extraction method are integrated into a unified framework that enables jointly learning of segmentation, feature extraction and classification for sensor-based activity recognition.

Extensive evaluations and ablation studies are conducted for each of the above proposed methods to compare with the state-of-the-art baselines. The contributions described above have led to the following publications:

- **Accepted:** A conference paper that has been accepted for publication with oral presentation in the 28th International Joint Conference on Artificial Intelligence in 2019 (**IJCAI-19 oral**) entitled “A Novel Distribution-Embedded Neural Network for Sensor-Based Activity Recognition” [[Qian et al., 2019a](#)].
- **Accepted:** A conference paper that has been accepted for publication with oral presentation in the 33rd AAAI Conference on Artificial Intelligence in 2019 (**AAAI-19 oral**) entitled “Distribution-based Semi-Supervised Learning for Activity Recognition” [[Qian et al., 2019b](#)].
- **Accepted:** A conference paper that has been accepted for publication with oral presentation in the 32nd AAAI Conference on Artificial Intelligence in 2018 (**AAAI-18 oral**) entitled “Sensor-based Activity Recognition via Learning from Distributions” [[Qian et al., 2018](#)].
- **Submitted:** A journal paper entitled “Weakly-Supervised Sensor-based Activity Segmentation and Recognition via Learning from Distributions” is submitted to Artificial Intelligence, 2019.

1.4 Thesis Organization

Figure 1.1 depicts a high-level outline of the thesis. The detailed structure of this thesis is organized as follows:

- Chapter 2: This chapter provides a comprehensive literature review on the field of wearable-sensor-based activity recognition.
- Chapter 3: This chapter introduces the preliminaries of our research works.
- Chapter 4: This chapter presents our research work SMM_{AR} and an accelerated version $R-SMM_{AR}$.
- Chapter 5: This chapter demonstrates a novel end-to-end neural network framework, i.e., Distribution-Embedded Deep Neural Network (DDNN).
- Chapter 6: This chapter introduces a semi-supervised learning method named Distribution-based Semi-Supervised Learning (DSSL).
- Chapter 7: This chapter introduces a weakly-supervised framework that enables joint learning of segmentation and feature extraction.
- Chapter 8: This chapter concludes the thesis and depicts some future research directions.

FIGURE 1.1: Illustration of the hierarchical structure of the thesis.

Chapter 2

Literature Review

This chapter provides a comprehensive review of existing works for the task of human activity recognition, with focus on aforementioned challenges in the last chapter that are related to our study.

2.1 Feature Extraction for Activity Recognition

It is well-known that good features can help to discriminate different classes of activities, by increasing the expressiveness of each activity. As mentioned in the previous chapter, feature extraction from each variate-length segment of data 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: feature-engineering-based and deep-learning-based. The former covers semantically meaningful features, while the latter contains deep neural networks as automatic feature extractors.

2.1.1 Feature-Engineering-Based Feature Extraction

Feature-engineering-based methods can be categorized into two kinds: statistical and structural [Lara and Labrador, 2013]. Statistical approaches include PCA, LDA, basis transform coding (wavelet transform and Fourier transform) and handcrafted statistical

features of raw signals including orders of moments (mean, variance, skewness, etc), median, etc [Janidarmian et al., 2017].

Besides statistical features, extra meta information among data can be taken into account as extra structural features. For instance, the ECDF approach [Hammerla et al., 2013, Plötz et al., 2011] leverages distributions' quantile function to preserve the overall shape as well as the spatial positions of time series data. Lin et al. [2007b] proposed the SAX method to discrete data into symbolic strings to represent equal probability mass.

The above feature extraction methods more or less require involvement of domain experts, which is time consuming. To this end, we propose to apply kernel methods to extract features. SMM_{AR} method [Qian et al., 2018] automatically extracts all orders of moments as statistical features by using kernel mean embedding technique. Kernel methods have been well studied during the past decades, with the ability to learn nonlinear transformations of input data as implicit features, and of learning nonlinear classifiers as well [Smola et al., 2007]. Recently, Muandet et al. [2017] illustrated the power of feature embedding on image classification, and Qian et al. [2018] investigated the similar technique on wearable-sensor-based activity recognition, with more reasonable and meaningful explanations on the extracted features. Similar technique has also been applied to the generative adversarial networks (GANs) with a different motivation of matching statistical features to enable the network to generate more realistic synthetic samples [Li et al., 2017, 2015].

2.1.2 Deep-Learning-Based Feature Extraction

Deep learning models are becoming prevalent in various applications, especially for many tasks in computer vision [LeCun et al., 2015]. The power of deep learning models lie in multiple layers of neurons where different layers extract different levels of features automatically. Different from existing feature extraction methods, deep learning methods largely relieve the effort on manual feature design and extraction procedure. Deep learning methods are capable of extracting both low-level and high-level features by training an end-to-end neural network. The first deep learning method on activity recognition applies Restricted Boltzmann Machines (RBMs) to compare with manual

features [Plötz et al., 2011]. Deep neural networks (DNNs) usually serve as dense layers of existing deep models [Hammerla et al., 2016], and usually larger number of hidden layers of DNNs enable the model with stronger representation capability. Convolutional neural networks (CNNs) are the most widely used frameworks in this field [Ignatov, 2018, Yang et al., 2015, Zeng et al., 2014]. CNNs enjoy two extra benefits than other models. The first benefit is local dependency where nearby signals are correlated together. The second benefit is scale invariance, which refers to the scale-invariant property for different paces or frequencies [Wang et al., 2017]. Despite the benefits of CNNs, they are originally designed for images, which is different from signals collected from wearable sensor. Yang et al. [2015] customized CNNs along temporal dimension of activity data to extract salient patterns of sensor signals at different time scales, as illustrated in Fig. 2.1. Besides, temporal dependencies in time-series data are proven to be beneficial for activity recognition as well. Recurrent neural networks (RNNs) are widely used in speech recognition and natural language processing by utilizing the temporal correlations between neurons. DeepConvLSTM model [Morales and Roggen, 2016] applies two Long Short-Term Memory (LSTMs) layers on top of the abstract feature representations extracted by four convolutional layers, whose architecture is shown in Fig. 2.2. There are also research works to jointly learn shallow features by traditional methods and deep features by deep models [Ignatov, 2018, Ravi et al., 2017]. There are also attempts of combinations of shallow classifiers with features learned by deep learning models. Hammerla et al. [2016] provided systematic comparisons on the performance of state-of-the-art deep learning methods with DNNs, CNNs and RNNs on activity recognition problems, especially various LSTMs.

FIGURE 2.1: Architecture Illustration of the CNN_Yang method.

FIGURE 2.2: Architecture Illustration of the state-of-the-art baseline method DeepConvLSTM.

2.2 Learning with Partial Labels

Limited labeled training data is insufficient to train a good classifier due to the cold start problem of supervised learning. Semi-supervised learning approaches are appealing in practice since they require only a small fraction of labeled training data with a large amount of easily obtained unlabeled data [Chapelle et al., 2010, Zhu, 2005]. Among existing semi-supervised learning approaches, manifold regularization [Sindhwani et al., 2005] and wrapping kernels using point cloud [Belkin et al., 2006] are two classic methods, which incorporate the manifold structure underlying both unlabeled and labeled data into the learning of Support Vector Machines (SVMs).

In the context of activity recognition, Stikic et al. [2009] proposed a multi-graph based semi-supervised approach named GLSVM, where each graph propagates different information of activities. Different graphs are then combined to improve label propagation in graphs. After that, an SVM classifier is trained by using both the initially labeled training data and the propagated labels. Matsushige et al. [2015] proposed a semi-supervised kernel logistic regression method for activity recognition, denoted by SSKLR, which extends kernel logistic regression into semi-supervised fashion, and solves the problem by the Expectation-Maximization algorithm. Yao et al. [2016] proposed a robust graph-based semi-supervised method named RSAR to tackle the intra-class variability in activities across different subjects. The RSAR method extracts the intrinsic shared subspace structures from activities with the assumption that intrinsic relationships have invariant properties thus are less sensitive with varying subjects. In Nazabal et al. [2016], a new Bayesian model is proposed to tackle the scenario

with limited number of sensors. The dynamic nature of human activities are further modeled as a first-order homogeneous Markov chain. Note that the setting of multi-instance learning (MIL) [Zhou and Xu, 2007] is related to the setting of learning from distributions, where each input example is a bag of instances. Indeed, each bag can be considered as a sample drawn from a distribution. Though existing MIL approaches do not explicitly incorporate the distributions information into learning a model, in some applications, these approaches can be applied to the setting of learning from distributions. Therefore, we consider some MIL approaches as baseline methods as well, especially for the kernel-based MIL method [Gärtner et al., 2002] and the graph-based semi-supervised MIL method [Rahmani and Goldman, 2006].

2.3 Time Series Segmentation

The goal of segmentation is to partition the time series data into continuous segments of variable lengths with changepoints or breakpoints in between. This is a crucial pre-processing step for sensor-based activity recognition, since a good segmentation is beneficial to learning an accurate activity classifier. The most widely used technique to segment the streams of activity data is fixed-size sliding window method, which partitions raw data into segments of fixed size, regardless the actual starting and ending points of each activity. Another way is to compute the difference between two adjacent windows and to compare it with specific threshold to decide whether a breakpoint is found or not.

General Segmentation algorithms on time-series data can be divided into two categories: exact search and approximate search methods. Exact search methods return segments with optimal solutions with regards to a predefined metric. All the breakpoints can be found with an exhaustive search method, at the cost high computational burden. Therefore, exhaustive search method is extremely difficult to scale up. A more advanced approach is dynamic programming (DP), which recursively solves segmentation on sub-sequences. Compared to exhaustive search, it is more computationally efficient, and also guaranteed to find the optimal solution provided the cost function. In order to reduce the computational cost, approximate segmentation methods produce

sub-optimal solutions. The previously mentioned sliding window algorithms are fast alternative methods, which can be operated in a online fashion.

A common idea to alleviate the computational burden is to prune the set of candidate changepoint locations and then run algorithms on the restricted set. The forward-backward dynamic programming algorithm [Guédon, 2013] computes several most probable segment candidates based on the reversibility property of time series data with the assumption that each segment of data tends to be constant. The cp3o method [Zhang et al., 2017] is proposed to utilize dynamic programming with pruned search space. It is achieved by comparing valid segmentation solutions to a specific solution during each iteration, and those indices with worse solutions will be removed from the set of candidate changepoints for the future iterations. The PELT method [Killick et al., 2012] is originally designed for 1-dimensional data with unknown number of segments and unknown segmentation locations. A linear cost function is designed to find the optimal number of segments, and a pruning step within DP is conducted to decrease complexity. PELT limits the set of potential changepoints by removing those indices of data which cannot reduce the cost function performed at each iteration. It does not affect the exactness of the segmentation under certain conditions. The cDPA algorithm [Hocking et al., 2015] and later its improved version GPDPA method [Hocking et al., 2017] are designed exclusively for peak finding problem using the Poisson likelihood for non-negative count data, and the DP is accelerated by adding an up-down constraint based on the property of peak signals. pDPA method [Rigaiil, 2010, 2015] also aims to improve the computational efficiency of an exact method through pruning, but in a different way. It prunes the set of candidates based on a functional representation of cost functions which introduces one additional scalar parameter, and then prune the functions which are not optimal. This method is optimal under certain conditions, however, it is only suitable for 1-dim data with only a few changepoints, and is restricted by the assumption that the model only has a single parameter within each segment. Two extension methods FPOP and SNIP [Maidstone et al., 2017] are proposed by combining pruning methods PELT and pDPA for 1-dim data. These methods are able to recover the optimal solutions. However, they are under a restrictive set of assumptions which greatly hinder the performance in real-world applications.

Chapter 3

Preliminaries

In the following we denote by $\mathbf{x}$ and $\mathbf{z}$ the random variables, with $\mathcal{X}$ and $\mathcal{Z}$ being the domain of variables respectively. and let $\mathbb{P}_x$ and $\mathbb{Q}_z$ be a probability measure on $\mathcal{X}$ and $\mathcal{Z}$. A joint probability measure on $\mathcal{X} \times \mathcal{Z}$ is denoted by $\mathbb{P}_{x,z}$. We assume all the measures are Borel measures, and the domains are compact.

3.1 Kernel Methods in Machine Learning

The core part of kernel methods is the inner product $\langle \mathbf{x}, \mathbf{x}' \rangle$, which can be viewed as a similarity measure between $\mathbf{x}$ and $\mathbf{x}'$. Any learning algorithm that can be expressed as inner product terms can benefit from kernel methods. Besides the linear function class, kernels can be induced by nonlinear similarity measures, i.e.,

$$\Phi : \mathcal{X} \rightarrow \mathcal{F} \tag{3.1a}$$

$$\mathbf{x} \rightarrow \phi(\mathbf{x}) \tag{3.1b}$$

where data can be mapped into a high-dimensional feature space $\mathcal{F}$ and subsequently evaluate the inner product in the space by:

$$k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle. \tag{3.2}$$

The ϕ is a feature map and k is a kernel function, respectively. One can construct the desired feature mapping ϕ for specific tasks. Alternatively, one can refrain from constructing explicit $\phi(\mathbf{x})$ if the representation can be represented in the form of inner product. This is called the kernel trick, which refrains from expensive computations of kernel methods.

Definition 1. [Aronszajn, 1950] A function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a reproducing kernel if it is symmetric, i.e., $k(\mathbf{x}, \mathbf{z}) = k(\mathbf{z}, \mathbf{x})$, and positive definite:

$$\sum_{i,j=1}^n c_i c_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0 \quad (3.3)$$

for any $n \in \mathbb{N}$ and choice of $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathcal{X}$ and $c_1, \dots, c_n \in \mathbb{R}$.

$\mathcal{H}$ is a Hilbert space of functions $\mathcal{X} \rightarrow \mathbb{R}$ with dot product $\langle \cdot, \cdot \rangle$. Formally,

Definition 3.1. A Hilbert space is a real (or complex) inner product space that is also a complete metric space w.r.t. the distance function induced by the inner product.

Moreover, $\mathcal{H}$ is a reproducing kernel Hilbert space (RKHS) of functions on $\mathbf{X}$ with kernel k if it satisfies the reproducing property:

$$\langle f(\cdot), k(\mathbf{x}, \cdot) \rangle = f(\mathbf{x}) \quad (3.4a)$$

$$\langle k(\mathbf{x}, \cdot), k(\mathbf{x}', \cdot) \rangle = k(\mathbf{x}, \mathbf{x}'). \quad (3.4b)$$

The formal definition of a RKHS is shown in the following:

Definition 3.2. A Hilbert space $\mathcal{H}$ is a RKHS if the evaluation functionals are bounded, i.e., if for all $\mathbf{x} \in \mathcal{X}$ there exists some $C > 0$ such that

$$|f(\mathbf{x})| \leq C \|f\|_{\mathcal{H}}, \forall f \in \mathcal{H}. \quad (3.5)$$

Intuitively, functions in the RKHS are smooth in the sense of 3.5. The smoothness property ensures that the solution in the RKHS will be well-behaved, i.e., small distance between two functionals $\|f - g\|_{\mathcal{H}}$ implies that $f(\mathbf{x})$ and $g(\mathbf{x})$ are close to each other.

This shows we can view the linear map from a function f to its value at $\mathbf{x}$ as an inner product. The evaluation functional is given by $k(\mathbf{x}, \cdot)$. An alternative view is the feature map from x to $\phi(\mathbf{x})$ such that $k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$. Commonly used kernels include the Gaussian and Laplacian kernels

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|_2^2}{2\sigma^2}\right), k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|_2^2}{\sigma}\right), \quad (3.6)$$

where $\sigma > 0$ is a bandwidth parameter. Another characterization of reproducing kernel k is the Mercer's theorem.

Theorem 3.3 (Mercer's theorem). [*Mercer and Forsyth, 1909*] Suppose k is a continuous positive definite kernel on a compact set $\mathcal{X}$, and the integral operator $\mathbf{T}_k : L_2(\mathcal{X}) \rightarrow L_2(\mathcal{X})$ defined by

$$(\mathbf{T}_k f)(\cdot) = \int_{\mathcal{X}} k(\mathbf{x}, \cdot) f(\mathbf{x}) d\mathbf{x} \quad (3.7)$$

is positive definite, i.e., for all $f \in L_2(\mathcal{X})$,

$$\int_{\mathcal{X}} k(\mathbf{u}, \mathbf{v}) f(\mathbf{u}) f(\mathbf{v}) d\mathbf{u} d\mathbf{v} \geq 0. \quad (3.8)$$

Then there is an orthonormal basis $\{\psi_i\}$ of $L_2(\mathcal{X})$ consisting of eigenfunctions of $\mathbf{T}_k$ such that the corresponding sequence of eigenvalues $\{\lambda_i\}$ are non-negative. The eigenfunctions corresponding to non-zero eigenvalues are continuous on $\mathcal{X}$ and $k(\mathbf{u}, \mathbf{v})$ has the representation

$$k(\mathbf{u}, \mathbf{v}) = \sum_{i=1}^{\infty} \lambda_i \psi_i(\mathbf{u}) \psi_i(\mathbf{v}) \quad (3.9)$$

where the convergence is absolute and uniform.

Throughout this thesis, we consider the positive definite kernels.

3.2 Kernel Mean Embedding of Distributions

The idea of extracting infinite number of statistical features may sound counter-intuitive, thus here we briefly introduce the kernel mean embedding technique [Muandet et al., 2017, Qian et al., 2018, Smola et al., 2007].

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,

$$\boldsymbol{\mu}_{\mathbb{P}} := \mu(\mathbb{P}) = \mathbb{E}_{\mathbf{x} \sim \mathbb{P}}[\phi(\mathbf{x})] = \mathbb{E}_{\mathbf{x} \sim \mathbb{P}}[k(\mathbf{x}, \cdot)], \quad (3.10)$$

where $\phi : \mathbb{R}^d \rightarrow \mathcal{H}$ 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 $\mathcal{H}$.

The kernel mean representation is fully characterized by the above transformation. As a result, we do not need to deal with distributions explicitly as many operations on distributions can be transformed into operations on $\boldsymbol{\mu}_{\mathbb{P}}$.

Theorem 3.4. [Smola et al., 2007] *If the kernel k is universal, then the mean map $\mu : \mathbb{P} \rightarrow \boldsymbol{\mu}_{\mathbb{P}}$ is injective.*

The injectivity in the above theorem indicates that 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.

A certain class of kernel functions known as characteristic kernels ensures that the kernel mean representation captures all necessary information about the distribution. In other words, the map is injective which implies that $\|\boldsymbol{\mu}_{\mathbb{P}} - \boldsymbol{\mu}_{\mathbb{Q}}\|_{\mathcal{H}} = 0$ if and only if $\mathbb{P} = \mathbb{Q}$. As a result, we can define metrics over the space of probability distributions.

The above inner product between two distributions can play the role of similarity measure, since the larger the inner product, the more similar the two distributions are.

In addition to the inner product of distributions, an alternative way to measure the similarity is to calculate the distance between the two distributions in the RKHS, i.e.,

$$D(\mathbb{P}_x, \mathbb{P}_z) = \|\boldsymbol{\mu}_{\mathbb{P}_x} - \boldsymbol{\mu}_{\mathbb{P}_z}\|, \quad (3.11)$$

with the guarantee of the above theorems.

The maximum mean discrepancy (MMD) considers functions in the unit ball of RKHS $\mathcal{F} : \{f \mid \|f\|_{\mathcal{H}} \leq 1\}$. MMD is defined as

$$\begin{aligned} MMD[\mathcal{H}, \mathbb{P}, \mathbb{Q}] &= \sup_{\|f\|_{\mathcal{H}} \leq 1} \left\{ \int f(\mathbf{x}) d\mathbb{P}(\mathbf{x}) - \int f(\mathbf{z}) d\mathbb{Q}(\mathbf{z}) \right\} \\ &= \sup_{\|f\|_{\mathcal{H}} \leq 1} \left\{ \langle f, \int k(\mathbf{x}, \cdot) d\mathbb{P}(\mathbf{x}) \rangle - \langle f, \int k(\mathbf{z}, \cdot) d\mathbb{Q}(\mathbf{z}) \rangle \right\} \\ &= \sup_{\|f\|_{\mathcal{H}} \leq 1} \left\{ \langle f, \boldsymbol{\mu}_{\mathbb{P}} - \boldsymbol{\mu}_{\mathbb{Q}} \rangle \right\} \\ &= \|\boldsymbol{\mu}_{\mathbb{P}} - \boldsymbol{\mu}_{\mathbb{Q}}\|_{\mathcal{H}}^2. \end{aligned} \quad (3.12)$$

Equivalently, we can express the MMD in terms of the associated k as

$$MMD[\mathcal{H}, \mathbb{P}, \mathbb{Q}] = \mathbf{E}_{\mathbf{x}, \mathbf{x}'}[k(\mathbf{x}, \mathbf{x}')] - 2\mathbf{E}_{\mathbf{x}, \mathbf{z}}[k(\mathbf{x}, \mathbf{z})] + \mathbf{E}_{\mathbf{z}, \mathbf{z}'}[k(\mathbf{z}, \mathbf{z}')], \quad (3.13)$$

where $\mathbf{x}'$ and $\mathbf{z}'$ are independent copy of $\mathbf{x}$ and $\mathbf{z}$ respectively. It follows that $MMD[\mathcal{H}, \mathbb{P}, \mathbb{Q}] = 0$ if and only if $\mathbb{P} = \mathbb{Q}$. Readers may refer to [Gretton et al., 2012] for details. Kernel mean embedding of distributions enables us to compute distances between distributions without the need for intermediate density estimation. The MMD technique has been applied extensively in many applications, including but not limited to independence tests, causal discovery, covariate shift and domain adaptation, etc. Here, we briefly list several popular applications.

Two-sample test [Gretton et al., 2012] aims to test whether the given two distributions $\mathbb{P}_x, \mathbb{P}_z$ are identical or not. In particular, we test the null hypothesis $\|\boldsymbol{\mu}_{\mathbb{P}} - \boldsymbol{\mu}_{\mathbb{Q}}\|_{\mathcal{H}}^2 = 0$ against the alternative hypothesis $\|\boldsymbol{\mu}_{\mathbb{P}} - \boldsymbol{\mu}_{\mathbb{Q}}\|_{\mathcal{H}}^2 \neq 0$. Distances between samples from two distributions are calculated by MMD.

Independence Measures [Smola et al., 2007] aim to test whether two random variables $\mathbf{x}$ and $\mathbf{z}$ are independent. The measurement measures the distance between the joint probability $\mathbb{P}_{\mathbf{x},\mathbf{z}}$ and the product of two marginal probabilities $\mathbb{P}_{\mathbf{x}} \times \mathbb{P}_{\mathbf{z}}$. Firstly, the mappings of distributions can be defined by

$$\begin{aligned}\mu_{\mathbb{P}_{\mathbf{x}\mathbf{z}}} &= \mathbf{E}_{\mathbf{x},\mathbf{z}}[v((\mathbf{x}, \mathbf{z}), \cdot)] \\ \mu_{\mathbb{P}_{\mathbf{x}}} \times \mu_{\mathbb{P}_{\mathbf{z}}} &= \mathbf{E}_{\mathbf{x}}\mathbf{E}_{\mathbf{z}}[v((\mathbf{x}, \mathbf{z}), \cdot)],\end{aligned}\tag{3.14}$$

where we denote by $\mathcal{V}$ as a RKHS over $\mathcal{X} \times \mathcal{Z}$ associated with kernel $v((\mathbf{x}, \mathbf{z}), (\mathbf{x}', \mathbf{z}'))$. If $\mathbf{x}$ and $\mathbf{z}$ are independent, the equality $\mu_{\mathbb{P}_{\mathbf{x}\mathbf{z}}} = \mu_{\mathbb{P}_{\mathbf{x}}} \times \mu_{\mathbb{P}_{\mathbf{z}}}$ will hold. To this end, we can define the distance of the two terms $\Delta = \|\mu_{\mathbb{P}_{\mathbf{x}\mathbf{z}}} - \mu_{\mathbb{P}_{\mathbf{x}}} \times \mu_{\mathbb{P}_{\mathbf{z}}}\|$ as a measure of dependence.

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,

$$\hat{\mu}_{\mathbb{P}} = \frac{1}{n} \sum_{i=1}^n \phi(\mathbf{x}_i) = \frac{1}{n} \sum_{i=1}^n k(\mathbf{x}_i, \cdot).\tag{3.15}$$

The above empirical mean estimation is a good proxy for the true expectation mean, which is supported by the below theorem:

Theorem 3.5. [Smola et al., 2007] Assume that $\|f\|_{\infty} \leq R$ for all $f \in \mathcal{H}$ with $\|f\|_{\mathcal{H}} \leq 1$. Then with probability at least $1 - \delta$, $\|\mu_{\mathbb{P}} - \hat{\mu}_{\mathbb{P}}\| \leq 2R_m(\mathcal{H}, \mathbb{P}) + R(-m^{-1} \log(\delta))$.

where $R_m(\mathcal{H}, \mathbb{P})$ denotes the Rademacher average associated with the distribution $\mathbb{P}$ and $\mathcal{H}$ [Altun and Smola, 2006, Bartlett and Mendelson, 2002].

Though in theory, the dimension of $\hat{\mu}_{\mathbb{P}}$ 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,

$$\langle \hat{\mu}_{\mathbb{P}_{\mathbf{x}}}, \hat{\mu}_{\mathbb{P}_{\mathbf{z}}} \rangle = \tilde{k}(\hat{\mu}_{\mathbb{P}_{\mathbf{x}}}, \hat{\mu}_{\mathbb{P}_{\mathbf{z}}}) = \frac{1}{n_x n_z} \sum_{i=1}^{n_x} \sum_{j=1}^{n_z} k(\mathbf{x}_i, \mathbf{z}_j),\tag{3.16}$$

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 $\mathbb{P}_{\mathbf{x}}$ and $\mathbb{P}_{\mathbf{z}}$, respectively. In general, $\tilde{k}(\cdot, \cdot)$ can be a

nonlinear kernel defined as follows,

$$\tilde{k}(\hat{\boldsymbol{\mu}}_{\mathbb{P}_x}, \hat{\boldsymbol{\mu}}_{\mathbb{P}_z}) = \langle \psi(\hat{\boldsymbol{\mu}}_{\mathbb{P}_x}), \psi(\hat{\boldsymbol{\mu}}_{\mathbb{P}_z}) \rangle, \quad (3.17)$$

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

3.3 Approximating the Kernel Mean Embedding

In real-world applications, the computational cost of kernel methods may be a critical issue, especially when dealing with large-scale data. Traditional kernel-based methods become computationally prohibitive as the volume of data explodes. The use of kernel mean embedding suffers from the issue due to two aspects. Firstly, the kernel mean estimator involves the weighted sum of the sample data. Secondly, the feature map $\phi(\cdot)$ of many kernel functions such as the Gaussian kernel lives in an infinite dimensional space. Often, the construction of Gram matrix $\mathbf{K}$ where $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ is required. Thereby, most kernel-based learning algorithms scale at least quadratically with the sample size, which makes them prohibitive for large-scale problems.

The existing approaches include two categories. The first category tries to find a smaller subset of samples which approximate well to the original samples. For instance, a sparse linear combination of samples can approximate the kernel mean [Cortes and Scott, 2014]. Sparsity-inducing norm can also be imposed on coefficients of kernel mean [Muandet et al., 2014]. The second category is to find a finite approximation of the feature mapping directly. Rahimi and Recht [2007] proposed two random feature construction schemes. The first type is random Fourier features, where data points are projected onto random vectors drawn from Fourier transform of the kernel and then passed through proper non-linearities. We will introduce this scheme in the following. The second type is called random binning, where the input space is partitioned by a random regular grid into bins and data points are mapped to indicator vectors of bins. Other approaches such as low-rank approximation are also applicable.

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 3.6 (Bochner’s Theorem [Bochner, 1933, 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 $k(\mathbf{x} - \mathbf{x}') = \int_{\mathbb{R}^d} e^{i\omega^\top(\mathbf{x}-\mathbf{x}')} d\mathbb{P}(\omega) = \int_{\mathbb{R}^d \times [0, 2\pi]} 2\cos(\omega^\top \mathbf{x} + b)\cos(\omega^\top \mathbf{x}' + b) d(\mathbb{P}(\omega) \times \mathbb{P}(b)) = \int_{\mathbb{R}^d} 2(\cos(\omega^\top \mathbf{x})\cos(\omega^\top \mathbf{x}') + \sin(\omega^\top \mathbf{x})\sin(\omega^\top \mathbf{x}')) d\mathbb{P}(\omega)$, where $\mathbb{P}(b)$ is a uniform distribution on $[0, 2\pi]$.*

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

$$k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle \approx \mathbf{z}(\mathbf{x})^\top \mathbf{z}(\mathbf{x}'), \quad (3.18)$$

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:

$$z_w(\mathbf{x}) = \sqrt{2}\cos(w^\top \mathbf{x} + b) \quad (3.19)$$

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].

3.4 Learning with Kernels

In supervised learning with distributions, we are given a set of labeled data $\{\mathbf{X}_i, y_i\}_{i=1}^n$, where $\mathbf{X}_i = \{\mathbf{x}_{ij}\}_{j=1}^{n_i}$ and n_i ’s may vary across different $\mathbf{x}_i$. The goal is to learn a classifier f to map $\{\mathbf{X}_i\}$ ’s to $\{y_i\}$ ’s. In SMMs [Muandet et al., 2012], each $\mathbf{X}_i$ is mapped to a functional in a RKHS $\mathcal{H}$ via kernel mean embedding [Berlinet and Thomas-Agnan, 2011] as $\mu_{\mathbb{P}_i} = \mathbb{E}_{\mathbf{x}_{ij} \sim \mathbb{P}_i}[k(\mathbf{x}_{ij}, \cdot)]$, where $k(\cdot, \cdot)$ is a characteristic kernel associated with

the RKHS $\mathcal{H}$. It has been proven that if the kernel is characteristic, then an arbitrary probability distribution $\mathbb{P}_i$ is uniquely represented by an element $\boldsymbol{\mu}_{\mathbb{P}_i}$ in the RKHS, which implicitly captures all orders of statistical moments of $\mathbf{X}_i$.

The inner product, i.e., a linear kernel, of two distributions, which measures their similarity, can be defined as $\langle \boldsymbol{\mu}_{\mathbb{P}_i}, \boldsymbol{\mu}_{\mathbb{P}_j} \rangle = \frac{1}{n_i n_j} \sum_{a=1}^{n_i} \sum_{b=1}^{n_j} k(\mathbf{x}_{ia}, \mathbf{x}_{jb})$. One can also define a nonlinear kernel of $\boldsymbol{\mu}_{\mathbb{P}_i}$ and $\boldsymbol{\mu}_{\mathbb{P}_j}$ to capture their nonlinear relationships via

$$\tilde{k}(\boldsymbol{\mu}_{\mathbb{P}_i}, \boldsymbol{\mu}_{\mathbb{P}_j})_{\tilde{\mathcal{H}}} = \langle \psi(\boldsymbol{\mu}_{\mathbb{P}_i}), \psi(\boldsymbol{\mu}_{\mathbb{P}_j}) \rangle, \quad (3.20)$$

where $\tilde{k}(\cdot, \cdot)$ is the nonlinear kernel induced by the nonlinear feature map $\psi(\cdot)$, and $\tilde{\mathcal{H}}$ is the corresponding RKHS.

To train a classifier from $\{\mathbf{X}_i\}$'s to $\{y_i\}$'s, SMMs define the optimization problem by learning $f \in \tilde{\mathcal{H}}$ that minimizes the following regularized risk functional

$$\frac{1}{n} \sum_{i=1}^n \ell(\boldsymbol{\mu}_{\mathbb{P}_i}, y_i, f) + \Omega(\|f\|_{\tilde{\mathcal{H}}}), \quad (3.21)$$

where $\ell(\cdot)$ is the loss function and $\Omega(\cdot)$ is the regularization term. Note that $\tilde{\mathcal{H}} = \mathcal{H}$ if $\tilde{k}$ is linear.

3.5 The Expectation Loss SVM (e-SVM) Method

We sought to find similar tasks with lack of exact labels for training samples. The e-SVM method was proposed to address the object detection task under weak supervision, where only bounding box annotations for images are available [Zhu et al., 2014]. On the one hand, the problem was formulated to be a binary classification problem, where positive labels in an image indicate the location of the target object, while negative labels indicate background. A linear function was adopted as the prediction function in terms of the parameters $\mathbf{w}$. On the other hand, due to the coarse annotations, only an approximate value u_i (computed by KL divergence) can indicate the probability of the i -th segment proposal belonging to the target object. The e-SVM algorithm treats u_i as

a latent variable, and model the objective function to be as follows:

$$\mathcal{L}(\mathbf{w}, \mathbf{u}) = \frac{1}{2} \lambda_{\mathbf{w}} \mathbf{w}^T \mathbf{w} + \frac{1}{N} \sum_{i=1}^N \{l_i^+ g(u_i) + l_i^-(1 - g(u_i))\} + \lambda_R R(\mathbf{u}), \quad (3.22)$$

where $l_i^+ = \max(0, 1 - \mathbf{w}^T \mathbf{x}_i)$ and $l_i^- = \max(0, 1 + \mathbf{w}^T \mathbf{x}_i)$. Note that the first term and the loss functions l_i^+ and l_i^- come from the standard SVM formulation, and the third term is a regularization term on u . The second term considers both possible labels $\{+1, -1\}$ assignment for segment proposals. To minimize the objective function, $\mathbf{u}$ and $\mathbf{w}$ were fixed alternatively to solve two convex optimization problems until it converges. Our proposed method is inspired by e-SVM in the way of solving complex optimization problems. However, our method and e-SVM differ in many ways, which will be explained in detail later.

Chapter 4

Sensor-based Activity Recognition via Learning from Distributions¹

4.1 Overview

Feature extraction in sensor-based activity recognition 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. To this end, in this chapter, 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

---

¹Partial results of the presented work have been published in [Qian et al., 2018]. Code is available at <https://github.com/Hangwei12358/R-SMM>.

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.

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 [Schölkopf and Smola, 2002, Smola et al., 2007], 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, 2015, Muandet et al., 2012], 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.

4.2 The Proposed Methodology

4.2.1 Problem Statement

In this chapter, 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 $\mathbf{X}_i = [\mathbf{x}_{i1} \dots \mathbf{x}_{in_i}] \in \mathbb{R}^{d \times n_i}$. 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 $Y = \{1, \dots, L\}$ 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.

4.2.2 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 $\{(\mathbf{x}_{ij}, y_i)\}_{j=1}^{n_i}$. 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.

4.2.3 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:

$$\boldsymbol{\mu}_i = \frac{1}{n_i} \sum_{p=1}^{n_i} \phi(\mathbf{x}_{ip}). \quad (4.1)$$

As a result, we have n pairs of input-output in the RKHS $\{(\boldsymbol{\mu}_1, y_1), \dots, (\boldsymbol{\mu}_n, y_n)\}$. Then our goal is to learn a classifier $f: \mathcal{H} \rightarrow \tilde{\mathcal{H}}$ such that $f(\boldsymbol{\mu}_i) = y_i$ for $i = 1, \dots, n$. 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,

$$\min_f \frac{1}{n} \sum_{i=1}^n \ell(f(\boldsymbol{\mu}_i), y_i) + \lambda \|f\|_{\tilde{\mathcal{H}}}, \quad (4.2)$$

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

$$f = \sum_{i=1}^n \alpha_i \psi(\boldsymbol{\mu}_i), \quad (4.3)$$

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 (4.3) can be reduced as

$$f = \sum_{i=1}^n \alpha_i \boldsymbol{\mu}_i, \text{ where } \alpha_i \in \mathbb{R}. \quad (4.4)$$

By specifying (4.3) or (4.4) using the Support Vector Machines (SVMs) formulation¹, we reach the following optimization problem, which is known as Support Measure Machines (SMMs) [Muandet et al., 2012],

$$\begin{aligned} \min_f \quad & \frac{1}{2} \|f\|_{\tilde{\mathcal{H}}}^2 + C \sum_{i=1}^n \xi_i, \\ \text{s.t.} \quad & y_i f(\boldsymbol{\mu}_i) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \\ & 1 \leq i \leq n, \end{aligned} \quad (4.5)$$

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

---

¹Note that one can also specify (7.6) or (7.7) using other loss functions, which result in different particular approaches.

kernels, $k(\cdot, \cdot)$ and $\tilde{k}(\cdot, \cdot)$, are specified¹, 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} .

4.2.4 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 (4.1) and (3.18), the empirical kernel mean map on a segment $\mathbf{X}_i$ with explicit Random Fourier Features can be written by

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

---

¹Recall 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.

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

$$f(\boldsymbol{\mu}_i) = \mathbf{w}^\top \boldsymbol{\mu}_i. \quad (4.6)$$

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

$$f(\boldsymbol{\mu}_i) = \mathbf{w}^\top \tilde{\mathbf{z}}(\boldsymbol{\mu}_i), \quad (4.7)$$

where $\tilde{\mathbf{z}} : \mathbb{R}^D \rightarrow \mathbb{R}^{\tilde{D}}$ is another mapping of Random Fourier Features. (4.6) is a special case of (4.7) when $\tilde{\mathbf{z}}$ is an identity mapping. The resultant optimization problem is reformulated accordingly as follows,

$$\min_{\mathbf{w} \in \mathbb{R}^{\tilde{D}}} \frac{1}{n} \sum_{i=1}^n \ell(\mathbf{w}^\top \tilde{\mathbf{z}}(\boldsymbol{\mu}_i), y_i) + \lambda \|\mathbf{w}\|_2^2. \quad (4.8)$$

As $\tilde{\mathbf{z}}(\cdot)$ is an explicit feature map, standard linear SVMs solvers can be applied to solve (4.8), which is much more efficient than solving (4.5). Accordingly, in the sequel, we denote this accelerated version of SMM_{AR} with Random Fourier Features by R-SMM_{AR} .

4.3 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} .

4.3.1 Datasets

Four benchmark datasets are used in our experiments. The overall statistics of the datasets are listed in Table 4.1.

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 4.1: 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 scenarios¹. Null class data representing none of the above target activities exists as well. 20 sensors are placed on the left and right arms of the participant. The features are accelerations of 3 spatial directions of each sensor. Each gesture is repeated about 70 times.

WISDM contains data collected through controlled laboratory conditions where accelerometers built into phones are used as sensors [Kwapisz et al., 2010]. A phone was put in each participant’s front pants leg pockets, with six regular activities performed².

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 seconds³.

PS is collected by four smart phones on four body positions: [Shoaib et al., 2013]. The smart phones 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 downstairs⁴.

¹The gestures include {“write on notepad”, “open hood”, “close hood”, “check gaps on the front door”, “open left front door”, “close left front door”, “close both left door”, “check trunk gaps”, “open and close trunk”, “check steering wheel”}. The dataset is available at <http://har-dataset.org/doku.php?id=wiki:dataset>.

²The activities are {“walking”, “jogging”, “ascending stairs”, “descending stairs”, “sitting and standing”}. The dataset is available at <http://www.cis.fordham.edu/wisdm/dataset.php#actitracker>.

³The dataset is available at <http://har-dataset.org/doku.php?id=wiki:dataset>.

⁴The dataset is available at <https://www.utwente.nl/en/eemcs/ps/research/dataset/>.

4.3.2 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.

4.3.3 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.

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 4.2: Overall comparison results on the four datasets (unit: %). The perfect prediction on HCI lies in the fact that the large # En. from Table. 4.1. 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.

4.3.3.1 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].
- **SAX- a .** Following the settings in [Lin et al., 2007b], 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 $a \in \{3, \dots, 10\}$.
- **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.

4.3.3.2 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 $\{1, \dots, 10\}$.

4.3.4 Overall Experimental Results

The overall comparison results of proposed methods along with all the baseline methods are presented in Table 4.2. 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%.

4.3.5 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. 4.1. 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 recognition performance. Hence the proposed method is likely to perform the best since all orders of moments information is utilized in the proposed method.

FIGURE 4.1: Comparison results of Moment- x in terms of miF on HCI dataset by varying moments and frequencies.

4.3.6 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. 4.2 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.

FIGURE 4.2: The miF performance on Skoda dataset 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.

4.3.7 Impact on Different Choices of Kernels

In SMM_{AR} , there are two types of kernels: $k(\cdot, \cdot)$ for kernel embedding within each segment (3.16) and $\tilde{k}(\cdot, \cdot)$ for training a nonlinear classifier (3.17). 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 4.3, 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).

	$\tilde{k}(\cdot, \cdot)$			
	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 4.3: Comparison performance in terms of miF of SMM_{AR} on Skoda with different combinations of kernels.

4.3.8 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 4.3 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.

4.4 Summary

In this chapter, we introduce 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 evaluations 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.

FIGURE 4.3: Comparison results between SMM_{AR} and R-SMM_{AR} in terms of runtime and miF score on Skoda dataset.

Chapter 5

A Novel Distribution-Embedded Neural Network for Sensor-Based Activity Recognition¹

5.1 Overview

As stated in the previous chapters, one crucial research issue on human activity recognition is how to extract proper features from the partitioned segments of multivariate sensor readings. Previously we have introduced feature-engineer-based and deep-learning-based feature extraction approaches. The approaches of the former category aim to extract various aspects of information underlying each sensor-reading segment, such as statistical information [Janidarmian et al., 2017], meta information, e.g., overall shape and spatial information [Hammerla et al., 2013, Lara and Labrador, 2013, Lin et al., 2007b]. The approaches of the latter category aim to design deep neural networks to extract temporal and/or spatial features from the segments of sensor readings automatically [Wang et al., 2017]. Different types of neural networks have been proposed to

---

¹Partial results of the presented work have been accepted in [Qian et al., 2019a]. Code is available at https://github.com/Hangwei12358/IJCAI2019_DDNN.

extract different kinds of information [Hammerla et al., 2016]. For instance, deep feed-forward networks (DNNs) are used to extract higher-level features without taking temporal or spatial information into consideration. Convolutional neural networks (CNNs) are used to extract locally translation invariant features with respect to the precise location or precise time of occurrence of certain pattern within a data segment [Ignatov, 2018, Yang et al., 2015, Zeng et al., 2014]. Recurrent neural networks (RNNs) are suitable for exploiting the temporal dependencies within the activity sequence. The state-of-the-art for sensor-based activity recognition are basically combinations of these three types of base models [Morales and Roggen, 2016].

Nevertheless, existing methods have different drawbacks. Feature-engineering-based methods are able to extract meaningful features, such as statistical or structural information underlying the segments, but usually require domain knowledge to manually design proper features for specific applications, which is labor-intensive and time consuming. To overcome the limitations of feature-engineering-based approaches, in the last chapter, we proposed the SMM_{AR} method [Qian et al., 2018] to automatically extract all orders of moments as statistical features by using kernel embedding technique of distributions. However, SMM_{AR} fails to extract temporal and spatial information from the segments of sensor readings, which is important for recognizing activities. Deep learning models, however, are able to learn temporal and/or spatial features from the sensor data automatically, but fail to capture statistical information, such as different orders of statistical moments, which has proven to be useful for activity recognition [Qian et al., 2018, 2019b].

In this chapter, we propose a novel deep learning model, i.e., Distribution-Embedded Deep Neural Network (DDNN) to automatically learn meaningful features including statistical features, temporal features and spatial correlation features for activity recognition in a unified framework. The main novelty of our network lies in that we encode the idea of kernel embedding of distributions into a deep architecture, such that besides temporal and spatial information, all orders of statistical moments can be extracted as features to represent each segment of sensor readings, and further used for activity classification in an end-to-end training manner. Compared with the SMM_{AR} method [Qian et al., 2018], which also makes use of the kernel embedding technique to

extract statistical features for sensor data, our proposed DDNN is capable of learning more powerful features beyond statistical features. In addition, SMM_{AR} assumes that all activities are segmented beforehand, while DDNN relaxes the perfect-segmentation assumption by simply using sliding windows, which makes DDNN more practical for real-world scenarios. Moreover, SMM_{AR} uses a single kernel to embed distributions, which may be sensitive to the parameter settings of the kernel, while DDNN uses a deep neural network to approximate the feature map of the kernel, which is more flexible as the parameters of the deep neural network are learned from the data. Extensive evaluations are conducted on four datasets to demonstrate the effectiveness of our proposed method compared with state-of-the-art baselines.

To summarize, our contributions in this chapter are two-fold:

- Our proposed DDNN is a unified end-to-end trainable deep learning model, which is able to learn different types of powerful features for activity recognition in an automated fashion.
- Extensive evaluations are conducted on several benchmark datasets to demonstrate the superior performance of our proposed DDNN.

5.2 The Proposed DDNN Model

5.2.1 The Overall Model

Activity recognition is challenging as it is affected by many factors, i.e., dynamic spatial-temporal correlations and varying patterns of activities conducted by multiple participants. Based on the above motivation, we design an end-to-end trainable neural network structure for human activity recognition problem. Our proposed model has three main modules to learn feature representations for human activity recognition:

- Statistical module f_1 : this module aims to learn all orders of moments statistics as features in an automated fashion.

- Spatial module f_2 : this module aims to learn correlations among sensors placements.
- Temporal module f_3 : this module aims to learn temporal sequence dependencies along the time scale.

By stacking the above learned features together and forming a unified architecture, we can build a trainable model for activity recognition. The overall illustration of the proposed model is shown in Figure 5.1.

In our problem setting of activity recognition, the streams of multivariate sensor readings are partitioned by fixed-size sliding window with length L . We randomly split activities into training set $\{(\mathbf{X}_i, y_i)\}_{i=1}^n$, validation set $\{(\mathbf{X}_j, y_j)\}_{j=1}^m$ and test set $\{\mathbf{X}_t\}_{t=1}^p$, where each activity $\mathbf{X}_i = [\mathbf{x}_{i1} \dots \mathbf{x}_{iL}] = [\mathbf{x}_i^1 \dots \mathbf{x}_i^d]^T \in \mathbb{R}^{d \times L}$, and $y_i \in \{1 \dots n_c\}$ with n_c denoting the number of predefined activity categories. Here each column $\mathbf{x}_{ij} \in \mathbb{R}^{d \times 1}$ is a vector of signals received from d sensors at j -th timestamp, and each row $(\mathbf{x}_i^r)^T \in \mathbb{R}^{1 \times L}$ represents the signals recorded by r -th sensor within the current sliding window.

Note that in this work, we simply concatenate these three modules' learned features $[f_1(\mathbf{X}_i), f_2(\mathbf{X}_i), f_3(\mathbf{X}_i)]$ before feeding into fully-connected layers. However, it is possible to explore more complex and interleaved ways to connect these modules depending on different scenarios. For instance, one possible choice is $f_1([f_2(\mathbf{X}), f_3(\mathbf{X}_i)])$, with which statistical features are learned on top of the features extracted by other two modules. This is actually a generalized way of learning features, i.e., $[f_2(\mathbf{X}), f_3(\mathbf{X})]$ is considered as a special type of data transformation of raw data $\mathbf{X}_i$, while $f_1(\mathbf{X}_i)$ learns features directly from raw data. It is also possible to build a deeper model with these three modules as atom building blocks.

5.2.2 Statistical Module

Inspired by SMM_{AR} [Qian et al., 2018], we aim to learn statistical features automatically by a deep learning model. One disadvantage of SMM_{AR} is that the learned features are limited by a *fixed* Gaussian kernel $k(x, x') = \exp(-\gamma\|x - x'\|^2)$ with *fixed* γ , hence

FIGURE 5.1: Illustration of the proposed DDNN architecture. The input to the network consists of a data sequence $\mathbf{X}_i = [\mathbf{x}_{i1} \dots \mathbf{x}_{iL}] = [\mathbf{x}_i^1 \dots \mathbf{x}_i^d]^T \in \mathbb{R}^{d \times L}$ extracted from d sensors and partitioned by sliding window approach with length L .

From left to right, there are three modules for extracting spatial, temporal and statistical features respectively. Note that the input data format for these modules are different. Spatial correlations among sensors whose signals are represented as row vectors $\{(\mathbf{x}_i^r)^T\}_{r=1}^d$ are learned by LSTMs. Temporal dependencies are extracted from column vectors $\{\mathbf{x}_i^j\}_{j=1}^L$ by both LSTMs and CNNs (we will explain later why CNNs extract temporal dependencies instead of spatial correlations). Statistical module take the matrix form data $\mathbf{X}_i$ as inputs of autoencoder. All the learned features are then concatenated into a single feature vector, which is input to the fully-connected layers.

parameter tuning of proper bandwidth for kernel is required in advance. Here we aim to learn statistical features from multiple kernels without manual parameter tuning. This statistical module can be seamlessly combined with other modules to form a unified deep learning architecture which can be trained and optimized.

First, we aim to design a neural network f_1 to learn the statistical feature mapping $\phi_{f_1}(\cdot)$ automatically, i.e.,

$$f_1(\mathbf{X}_i) = \phi_{f_1}(\mathbf{X}_i). \quad (5.1)$$

However, the desired ϕ_{f_1} takes the matrix as input, while ϕ_k works for vectorial input. To address this issue, we take the average of feature mapping within each sliding window as

$$\phi_{f_1}(\mathbf{X}_i) = \frac{1}{L} \sum_{j=1}^L \phi_k(\mathbf{x}_{ij}). \quad (5.2)$$

Second, we expect f_1 is able to learn the best kernel automatically from different possible characteristic kernels $k \in \mathcal{K}$.

$$f_1^*(\mathbf{X}_i) = \max_{f_1} \phi_{f_1}(\mathbf{X}_i) = \max_{k \in \mathcal{K}} \frac{1}{L} \sum_{j=1}^L \phi_k(\mathbf{x}_{ij}). \quad (5.3)$$

Note that the learned features $f_1^*(\mathbf{X}_i)$ are in vectorial form. As mentioned in Chapter 3, the prerequisite of expressive feature extraction is the characteristic property of kernels, i.e., the feature mapping $f_1(\cdot)$ should be injective (not necessarily invertible). To make the neural network injective, there should be another function or neural network f_1^{-1} such that $f_1^{-1}(f_1(\mathbf{X}_i)) = \mathbf{X}_i$ for all possible $\mathbf{X}_i$'s. Therefore, as suggested in [Li et al., 2017], we utilize an autoencoder to guarantee the injectivity of the feature mapping.

To be specific, an autoencoder includes an encoder f_e , and a decoder f_d , where the encoder is used to map the input sequence to a fixed-length vector, then the decoder is used to unroll this vector to sequential outputs and try to reconstruct the input data of the encoder. In our scenario, the encoder is the desired f_1 module, and $f_d = f_1^{-1}$. Though both of our proposed model and the model in [Li et al., 2017] utilize an autoencoder to make sure the injectivity of neural networks, the motivations are quite different. We utilize the autoencoder as feature learner for classifying activity classes, while in their model, the autoencoder works for hypothesis testing, i.e., to make generated synthetic samples as indistinguishable from true samples as possible.

The standard loss function of the autoencoder tries to minimize the reconstruction error $\ell_{ae} = \|x - f_d(f_e(\tilde{x}))\|$ between inputs x and outputs $\tilde{x}$, but it is insufficient for statistical feature learning. We further use an extra loss function based on MMD

distance to force the autoencoder to learn good feature representations of inputs:

$$\begin{aligned} \text{MMD}_k(\mathbf{X}_p, \mathbf{X}_q) &= \left\| \frac{1}{n_p} \sum_{i=1}^{n_p} (\phi_k(\mathbf{x}_i)) - \frac{1}{n_q} \sum_{j=1}^{n_q} (\phi_k(\mathbf{x}_j)) \right\|_2 \\ &= \sqrt{\frac{1}{n_p^2} \sum_{i,i'} k(\mathbf{x}_i, \mathbf{x}_{i'}) - \frac{2}{n_p n_q} \sum_{i,j} k(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{n_q^2} \sum_{j,j'} k(\mathbf{x}_j, \mathbf{x}_{j'})} \end{aligned}$$

where n_p and n_q are the numbers of timestamps of two activities $\mathbf{X}_p$ and $\mathbf{X}_q$, respectively. The resultant MMD loss function on the autoencoder is as follows:

$$\ell_{\text{MMD}}(\mathbf{X}_i, f_d(f_e(\mathbf{X}_i))) = \frac{1}{L} \left\| \sum_{j=1}^L f_e(\mathbf{x}_{ij}) - f_e(f_d(f_e(\mathbf{x}_{ij}))) \right\|_2.$$

Note that by taking f_e and f_d to be the identity function, ℓ_{MMD} is reduced to ℓ_{ae} , where the mean vector (1st order moment) difference between inputs and outputs of the autoencoder is calculated. Our choices for f_e and f_d in the proposed deep learning model aim to match higher order moments statistical features. Therefore, this loss function forces the hidden representations of autoencoder to successfully convey sufficient information of desired statistics to the decoder.

5.2.3 Spatial Module

Convolutional layers in CNNs are firstly designed for the image-based problems. The standard CNNs are able to extract spatial-invariant features with a kernel filter running over the images or videos. However, the current so-called CNNs for human activity recognition tasks are actually not truly on spatial dependencies. Usually the wearable sensor data is in 1-dimension, where the so-called spatial CNNs are actually along the temporal aspect [Hammerla et al., 2016, Morales and Roggen, 2016]. There are also attempts to force the multiple 1-dimensional data of different sensor channels into a virtual image, then standard CNNs can be applied [Yang et al., 2015].

Our viewpoint of spatial correlations are different from the previous work. We try to capture the spatial correlations between sensors attached to the human body. From our point of view, the signals of a certain sensor are inevitably affected by the attached

locations on the human body or joints. Imagine a participant is walking, with sensors attached to his upper arm, lower arm and legs. It is common that when the right leg of the participant is on the front, the right arms are waved to the opposite direction at the same time. Also the movements of upper arm and lower arm are constrained by the joints of the human body. Therefore we aim to model such kinds of spatial correlations of the sensors, which is usually ignored in the literature. As illustrated in Figure 5.1, the input data $\mathbf{X}_i$ in the sliding window is treated as d row vectors $[\mathbf{x}_i^1 \dots \mathbf{x}_i^d]^T$, each of which associated to a single sensor. A LSTM is connected with each sensor data, and hence the dependencies between sensors are learned to form a spatial feature vector.

5.2.4 Temporal Module

In order to exploit the temporal dependencies within each activity, we utilize both CNNs and LSTMs as building blocks of temporal module. As discussed in previous subsection, CNNs with 1-D filters are applied on each channel $\{\mathbf{x}_i^r\}_{r=1}^d$ of sensor data $\mathbf{X}_i$. By applying the filter to go through different regions of the input, it is then able to detect the local salience patterns of the signals. Note that CNNs are applied along the temporal dimension, thus it is able to learn the temporal dependencies. Besides, LSTMs are connected to temporal data $\{\mathbf{x}_{ij}\}_{j=1}^L$ to learn temporal information as well. Specifically, we choose LSTMs instead of RNNs due to the diminishing gradients problem. LSTMs are designed to have more dynamic and flexible memory cells through gating mechanism, which enables LSTMs to learn temporal relationships on longer time scales. The outputs of CNNs and LSTMs are concatenated into a single vector to represent temporal features.

5.3 Experiments

5.3.1 Datasets

We conduct experiments on four sensor-based activity datasets. The overall statistics information of datasets are listed in Table 5.1. The Daphnet Gait dataset (DG) [Bächlin

et al., 2010] corresponds to a medical application and records activities from 10 participants affected with Parkinson’s Disease, aiming to detect freezing of gait incidents¹. The data is segmented by sliding window of 1 second duration and 50% overlap. The Opportunity dataset (OPPOR) [Chavarriaga et al., 2013] comprises 17 mid-level gesture classes conducted in an ambient-sensor home environment together with 19 on-body sensors. These gestures are short in duration and non-repetitive. Null class data exists in the dataset indicating transitions of two adjacent activities². The UCIHAR dataset [Anguita et al., 2012] collects six activities (walking, walking upstairs, walking downstairs, sitting, standing, laying) carried out with a group of 30 volunteers within an age range of 19-48 years³. The PAMAP2 dataset [Reiss and Stricker, 2012]⁴ includes 12 different physical activities (household activities and exercise activities) which are performed by 9 subjects wearing 3 inertial measurement units. These activities are prolonged and repetitive, typical for systems aiming to characterize energy expenditure.

Datasets	# train	# val.	# test	# sw	# Feature	# Class	Frequency	# Subjects
OPPOR	715,785	32,224	121,378	30	113	18	30	4
UCIHAR	941,056	NA	377,216	128	9	6	50	30
DG	312,970	37,122	30,188	32	9	2	100	10
PAMAP2	473,447	90,814	83,366	170	52	12	100	9

TABLE 5.1: The overall information of the four datasets. Note that “# train”, “# val.” and “# test” refer to total number of training, validation and test samples, respectively. “#sw” denotes the sliding window length used in the experiments. UCIHAR is preprocessed and segmented beforehand by the data provider, which does not contain validation set.

5.3.2 Experimental Setup

All these datasets have class imbalance problem, especially OPPOR and DG. Therefore, in our experiments, we set the probability of an activity being chosen in a training epoch to be the inverse of the number of the certain activity. We follow the experimental setup

¹The dataset is available at <https://archive.ics.uci.edu/ml/datasets/Daphnet+Freezing+of+Gait>.

²The dataset is available at <https://archive.ics.uci.edu/ml/datasets/OPPORTUNITY+Activity+Recognition>.

³The dataset is available at <https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones>.

⁴The dataset is available at <http://archive.ics.uci.edu/ml/datasets/pamap2+physical+activity+monitoring>.

in [Hammerla et al., 2016]. Micro-F1 (miF) and weighted macro-F1 (maF) are selected as performance measure. 77 out of 113 features are used for OPPOR, with run 2 from subject 1 as validation set, runs 4 and 5 from subject 2 and 3 as test set and the rest as training set. Sliding windows of 1 second duration with 50% overlap is applied. For PAMAP2, 12 protocol activities are studied, with data downsampled to 33Hz. Sliding window length is 5.12 seconds with 1 second as step size. Runs 1 and 2 for subject 5 are used as validation set, and runs 1 and 2 for subject 6 are used as test set, with the rest being training set. The raw data of DG is downsampled to 32Hz as well. Sliding window duration is 1 second with half overlap. We use subject 9’s first run as validation set, subject 2’s runs 1 and 2 as test set with the rest being training set. The UCIHAR has been preprocessed and segmented by data provider beforehand, where the raw data is randomly partitioned into two sets, where 70% of the volunteers generated training data and 30% the test data. The sensor signals were pre-processed by applying noise filters and then sampled in sliding windows of 2.56 second (128 readings). Data normalization is conducted on all datasets. For our architecture, we utilize 4 linear layers with ReLU attached after each linear layer as encoder and decoder’s architecture. Both LSTMs in spatial and temporal module have l layers of LSTMs with h -dimensional hidden representations, where $l \in \{1, 2, 3\}$ and $h \in \{32, 64, 128, 256, 512, 1024\}$. Four convolutional layers with filter size (1, 5) are utilized in the temporal module, with ReLUs and max pooling layers attached after each convolutional layer. All feature vectors are concatenated into a single vector before feeding into three fully-connected layers. The batch size is set to 64, and the maximum training epoch is 100. Adam optimizer is used for training with learning rate 10^{-3} and weight decay 10^{-3} . All experiments are run on a Tesla V100 GPU.

5.3.3 Baselines

We compare our proposed model with baseline methods as well as state-of-the-art methods. Due to the fact that feature-engineering-based machine learning methods are hard to scale up, in this paper we mainly compare our proposed DDNN model with deep learning based methods.

- DDNN– f_1 : the proposed deep model without the statistical module. This baseline is set to investigate the efficacy of the statistical module.
- DDNN– f_2 : the proposed deep model without the spatial module. This baseline is set to investigate the efficacy of the spatial module.
- CNN_Yang [Yang et al., 2015]: a state-of-the-art CNN-based model with 3 convolutional layers. We follow the architecture in the paper and reproduce the model.
- DeepConvLSTM [Morales and Roggen, 2016]: a state-of-the-art model with 4 convolutional layers and 2 LSTM layers. We also follow the architecture and reproduce the model.
- DNN: 5-layer linear transformation with ReLU activation function.
- CNN: 4-layer CNNs with kernel size (1, 5) with ReLU activation function and max pooling layer attached to the output of each CNN.
- LSTM: 2-layer LSTMs with the dimension of hidden representation in the range {32, 64, 128, 256, 512}.
- LSTM-f, LSTM-S, b-LSTM-S: state-of-the-art LSTMs variants to capture temporal sequences information. Results are directly from [Morales and Roggen, 2016].

5.3.4 Experimental Results and Analysis

The results of the proposed method and baselines on 4 datasets are listed in Table 5.2. The best performance for each evaluation metric is highlighted in bold. Our proposed DDNN has achieved the best performance on all datasets, except for the maF of OPPOR. These results indicate that our proposed model is capable of learning powerful various features for classification of activity recognition with more discriminative power.

Methods	DG		OPPOR		UCIHAR		PAMAP2	
	miF	maF	miF	maF	miF	maF	miF	maF
DDNN	92.59	91.61	83.66	86.01	90.53	90.58	93.23	93.38
DDNN- f_1	91.38	90.67	81.27	84.51	89.96	89.93	87.49	86.84
DDNN- f_2	89.67	88.97	77.96	82.27	88.60	88.58	89.37	89.43
CNN_Yang	87.96	86.65	9.98	2.95	88.12	88.11	70.17	70.46
DeepConvLSTM	87.21	84.28	75.47	78.92	89.05	89.07	84.31	82.73
DNN	88.91	86.47	77.05	80.25	87.65	87.72	80.31	79.82
CNN	89.23	88.85	10.66	3.56	86.66	86.77	89.75	89.72
LSTM	88.34	86.93	63.17	69.92	74.52	74.75	90.38	90.29
LSTM-f*	67.3	-	67.2	90.8	-	-	92.9	-
LSTM-S*	76.0	-	69.8	91.2	-	-	88.2	-
b-LSTM-S*	74.1	-	74.5	92.7	-	-	86.8	-

TABLE 5.2: Overall comparison results on the four datasets (unit: %). Note that the results of baselines with * are directly copied from [Morales and Roggen, 2016].

5.3.5 Impact of Spatial and Statistical Module

Remarkably, the performances of DDNN are consistently better than those of DDNN- f_1 and DDNN- f_2 on all datasets. This favorably validates our motivation that statistical features and spatial features are beneficial to the deep learning models besides the widely used temporal features in existing literature.

5.3.6 Robustness of the Proposed DDNN

One interesting finding is that our proposed model is more robust than other baselines. For instance, LSTM-related methods are obviously inferior on DG and UCIHAR, and CNN-based models are much worse than other baselines in OPPOR. One possible reason may lie in the unified framework of DDNN, where different aspects of features are learned together. It is reasonable that the contributions of different features on the classification performance are task-dependent, i.e., the importance of statistical module f_1 and spatial module f_2 varies in datasets since each dataset has unique characteristics on properties.

5.3.7 Parameter's Sensitivity

Another aspect of robustness is found during parameter tuning, where DDNN is less sensitive to the changes of parameters. For example, when we set the number of LSTM layers to be $\{1, 2, 3\}$, and LSTMs hidden representation dimensions to be $\{32, 64, 128, 256, 512\}$, the performance difference of DDNN is only roughly several percentage, while other models' performance gap is larger. We also investigate the dimensions of hidden representations in the autoencoder of statistical module ranging from $0.5d$ to $10d$ with d indicating the number of dimensions of raw data. Empirically, higher dimensional hidden representations actually hinder the performance of deep model, while the dimensions lower than $4d$ does not affect the performance drastically. We also investigate the weights on the added loss function ℓ_{MMD} for statistical module. We conduct experiments with various weights put on the loss function. As illustrated in Figure 5.2, the performance is steady (ranging from 0.88 to 0.9) within the weight ranging from 10^{-4} to 10^1 , but when the weights are larger than 10^1 , the performance degrades drastically. The reason may be the large weights on the ℓ_{MMD} leads to less contribution of the rest two modules (temporal and spatial), which affects the final performance.

FIGURE 5.2: Illustration of performance difference with different weights put on the loss function ℓ_{MMD} .

5.4 Summary

In this chapter, we propose a novel architecture for wearable-sensor-based activity recognition tasks. Our proposed DDNN model is able to automatically learn three types of features: 1) statistical features, 2) spatial correlations among sensors, and 3) temporal features. Extensive evaluations with analysis are conducted to compare with state-of-the-art methods. Experimental results demonstrate the superior efficacy of the proposed model.

Chapter 6

Distribution-based Semi-Supervised Learning for Activity Recognition¹

6.1 Overview

Though SMM_{AR} is able to systematically extract powerful statistical features, as a supervised learning based method, it requires a plethora of labeled data for training. Note that label annotation on a large-scale dataset on sensor readings is a costly process. Therefore, growing research interests have been focused on exploring the trade-off between label ambiguity and human annotation effort. Some researchers focus on efficient annotation strategies to reduce labeling effort, including offline and online strategies [Stikic et al., 2011], such as experience sampling, self-recall and video recording. Specifically, semi-supervised learning methods utilize a large amount of unlabeled data besides a few labeled data [Zhu, 2005]. This setting is widely applicable in a variety of real-world applications, where unlabeled data is abundant but labeling all instances may not be practical. Besides, the large amount of unlabeled data can shed light on the underlying structure and manifolds of all data, thereby boosting the learning process. Most existing methods construct a graph to propagate labels by utilizing manifold structure [Belkin et al., 2006], and all the data points are treated as nodes in the graph, which

---

¹Partial results of the presented work have been published in [Qian et al., 2019b]. Code is available at https://github.com/Hangwei12358/AAAI2019_DSSL.

are used to approximate density along manifolds. Connected nodes with a path through high density regions are likely to share the same label. However, it is surprising that relatively few approaches have implemented activity recognition in a semi-supervised fashion [Lara and Labrador, 2013]. Due to the above considerations, semi-supervised learning draws our primal research interest.

Therefore, in this chapter, we propose a novel semi-supervised learning method, namely Distribution-based Semi-Supervised Learning (DSSL), to tackle the aforementioned limitations. Intensive efforts on data annotation as well as feature engineering are freed by using the kernel mean embedding technique for distributions. The proposed method is capable of automatically extracting powerful features with no domain knowledge required, meanwhile, alleviating the heavy annotation effort through semi-supervised learning. To elaborate, we treat data stream of sensor readings received in a period as a distribution, and map all training distributions, including labeled and unlabeled, into a reproducing kernel Hilbert space (RKHS) using the kernel mean embedding technique. The RKHS is further altered by exploiting the underlying geometry structure of the unlabeled distributions. Finally, in the altered RKHS, a classifier is trained with the labeled distributions. We conduct extensive evaluations on three public datasets to verify the effectiveness of our method compared with state-of-the-art baselines. Our proposed method, DSSL, is an extension of SMM_{AR} in the semi-supervised learning manner. Compared with SMM_{AR} and other supervised or semi-supervised learning methods for activity recognition, our contributions are 4-fold:

- Compared with other supervised or semi-supervised learning methods, DSSL is able to represent each instance, i.e., data stream of a period, using all the orders of statistical moments implicitly and automatically, which contains rich information to distinguish activities.
- Compared with SMM_{AR} , DSSL relaxes its full supervision assumption, and is able to exploit unlabeled instances to learn an underlying data structure. With the learned structure and a few labeled instances, DSSL is able to learn a precise classifier for activity recognition.

- Most existing works on learning with distributions are supervised. To the best of our knowledge, DSSL is the first attempt on semi-supervised learning with distributions. Moreover, we provide theoretical analysis proving that DSSL is valid for semi-supervised learning in a reproducing kernel Hilbert space (RKHS).
- Extensive evaluations are conducted to demonstrate the superior performance of DSSL over a number of state-of-the-art baselines.

6.2 The Proposed Methodology

6.2.1 Problem Statement

In our project setting of activity recognition, we are given a set of l labeled segments data $\{\mathbf{X}_i, y_i\}_{i=1}^l$, and a set of $u = n - l$ unlabeled segments $\{\mathbf{X}_i\}_{i=l+1}^{i=n}$ as training data obtained by applying segmentation methods on the raw data, where $\mathbf{X}_i = [\mathbf{x}_{i1} \dots \mathbf{x}_{in_i}] \in \mathbb{R}^{d \times n_i}$, $y_i \in \{1, \dots, L\}$, $l \ll u$, and n_i may vary across different segments. The goal is to make use of both labeled and unlabeled segments to learn a classifier from each segment $\mathbf{X}$ to its corresponding label y .

Following [Qian et al., 2018], each segment $\mathbf{X}_i$, including both labeled and unlabeled, is treated as a *sample* of n_i data points drawn from an unknown distribution $\mathbb{P}_i$. Kernel mean embedding is then applied to map each $\mathbf{X}_i$ to an element $\mu_{\mathbb{P}_i}$ in a RKHS. In practice, to make the learning process more efficient, random Fourier features are used to approximate the nonlinear feature map induced by the kernel of the RKHS via $\mu_{\mathbb{P}_i} = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathbf{z}(\mathbf{x}_{ij})$, where $\mu_{\mathbb{P}_i} \in \mathbb{R}^D$. Therefore, our goal becomes to learn a classifier $f : \mu_{\mathbb{P}} \rightarrow y_i$ from $\{\mu_{\mathbb{P}_i}, y_i\}_{i=1}^l$ and $\{\mu_{\mathbb{P}_i}\}_{i=l+1}^{i=n}$.

6.2.2 Distribution-based Semi-Supervised Learning

Borrowing the idea from manifold regularization [Belkin et al., 2006] and the technique on warping data-dependent kernels [Sindhwani et al., 2005], we aim to incorporate the underlying manifold structure of both labeled and unlabeled data into the learning

of a classifier via warping a RKHS. Specifically, we wrap the RKHS $\tilde{\mathcal{H}}$ defined in (3.20) to another RKHS $\check{\mathcal{H}}$ by leveraging unlabeled training segments or distributions to reflect the underlying geometry of $\{\psi(\mu_{\mathbb{P}_i})\}$'s. Notations on different kernels and their corresponding RKHSs used in this paper are summarized in Table 6.1. The new RKHS $\check{\mathcal{H}}$ is associated with the new kernel $\check{k}$, which is data-dependent for semi-supervised learning. We will discuss how to achieve the kernel as well as the resulting new space later. Here, we assume the new kernel $\check{k}$ is constructed, then the revised optimization problem over $\check{\mathcal{H}}$ is formulated as

$$f^* = \arg \min_{f \in \check{\mathcal{H}}} \frac{1}{l} \sum_{i=1}^l \ell(\mu_{\mathbb{P}_i}, y_i, f) + \|f\|_{\check{\mathcal{H}}}^2, \quad (6.1)$$

where $\ell(\cdot)$ is the loss function. Note the objective function looks similar to that in the supervised learning setting in (3.21). However, in (6.1) the RKHS, where the functional to be optimized is $\check{\mathcal{H}}$, which is influenced by both labeled and unlabeled distributions, while the RKHS in (3.21) is $\tilde{\mathcal{H}}$, which is defined by labeled distributions only. The new optimization problem raises a potential problem: f is to be learned in $\check{\mathcal{H}}$, while the input space of $\mu_{\mathbb{P}_i}$ is $\mathcal{H}$. As these RKHSs are not the same, how to calculate the loss function remains a problem. To sum up, in order to solve the optimization problem (6.1), three crucial questions need to be answered:

- How to construct the data-dependent kernel $\check{k}$ by incorporating unlabeled training data?
- Is the new space $\check{\mathcal{H}}$ valid?
- How to calculate the loss function given $\mu_{\mathbb{P}} \in \mathcal{H}$ and $f \in \check{\mathcal{H}}$ are not in the same space?

In the following, we investigate the questions one by one.

6.2.2.1 1) Construction of the Data-dependent Kernel $\check{k}$

Since unlabeled data may shed light on the underlying structure and manifolds of all data, now the problem becomes how to appropriately construct such a valid RKHS $\check{\mathcal{H}}$

TABLE 6.1: Notations of different kernels used in Chapter 6.

Kernel	Space	Descriptions
k	$\mathcal{H}$	kernel mean embedding of distributions
$\tilde{k}$	$\tilde{\mathcal{H}}$	kernel on the embedded distributions
$\check{k}$	$\check{\mathcal{H}}$	data-dependent kernel constructed based on $\tilde{k}$ for semi-supervised learning

from $\tilde{\mathcal{H}}$ to achieve so. We first define $\check{\mathcal{H}}$ to be the space of functionals from $\tilde{\mathcal{H}}$ with the following modified inner product:

$$\langle f, g \rangle_{\check{\mathcal{H}}} \triangleq \langle f, g \rangle_{\tilde{\mathcal{H}}} + \langle Sf, Sg \rangle_{\mathcal{V}}, \quad (6.2)$$

where $\mathcal{V}$ is a linear space and $S : \tilde{\mathcal{H}} \rightarrow \mathcal{V}$ is a bounded linear operator. The first term in (6.2) is the common definition of inner product between two functionals, while the second term with the operator S reflects that unlabeled embedded distributions alter our beliefs in the overall structure. Denote by $\mathbf{f}(\boldsymbol{\mu}) = (f(\boldsymbol{\mu}_{\mathbb{P}_1}), \dots, f(\boldsymbol{\mu}_{\mathbb{P}_n}))$, we have $\langle Sf, Sf \rangle_{\mathcal{V}} = \mathbf{f}(\boldsymbol{\mu})M\mathbf{f}(\boldsymbol{\mu})^\top$ with M being a positive semidefinite matrix.

6.2.2.2 2) Validity of $\check{\mathcal{H}}$

Theorem 6.1. $\check{\mathcal{H}}$ is a valid RKHS.

A space is valid if it is bounded and complete. Detailed proofs for this theorem are shown in Section. 6.3.

6.2.2.3 3) Loss Function Calculation

Based on Theorem 6.1, we have the following propositions.

Proposition 1. $\check{\mathcal{H}} = \tilde{\mathcal{H}}$.

The two spaces are the same if each of the space is the subset of the other space. Although the two spaces are the same, the kernels therein are not identical. However, they are connected due to the involvement of unlabeled distributions. Detailed proofs for this theorem are shown in Section. 6.3.

Proposition 2. $\check{K} = (I + \check{K}M)^{-1}\check{K}$, where $\check{K}$ with $\check{K}_{ij} = \check{k}(\mu_{\mathbb{P}_i}, \mu_{\mathbb{P}_j})$ is the kernel matrix for $\check{\mathcal{H}}$ on $\mu_{\mathbb{P}_i}$'s, and $\check{K}$ is the kernel matrix in the altered space $\check{\mathcal{H}}$.

Note that detailed proofs and derivations of theorems and propositions introduced in this section can be found in next section. The complexity of the above kernel seems to be a potential problem when the data scales up, since it involves matrix multiplication as well as matrix inversion. However, when conducting experiments on large scale activity recognition datasets, the problem actually is not severe in practice. The reason is that the entries of kernels are dependent on the number of distributions, i.e., number of segments, each containing a repetition of activity, instead of the number of total instances, i.e., one entry for each timestamp equivalent to the product of # sample and # instances per sample. Other feasible solutions to further alleviate this problem include matrix factorization, low-rank approximation [Bach and Jordan, 2005], etc. Data selection or feature selection [Nie et al., 2010] can be conducted on training data beforehand to keep a small fraction of key training data. The proposed method can be further developed in an online learning fashion [Hoi et al., 2014], so that the matrix are maintained in a small scale.

Note that the choice of M is crucial regarding how to properly incorporate unlabeled embedded distributions. In this paper, we set M to be $M = rL^2$, where r is a scalar and $L = D - W$ is the Laplacian matrix, which is widely used in semi-supervised learning [Belkin et al., 2006, Sindhwani et al., 2005] to model the geometry structure underlying the data. To be specific, $W_{ij} = \exp\left(-\frac{\|\mu_{\mathbb{P}_i} - \mu_{\mathbb{P}_j}\|^2}{2\sigma^2}\right)$ if $\mu_{\mathbb{P}_i}$ and $\mu_{\mathbb{P}_j}$ are connected in the graph, and D is the diagonal matrix with $D_{ii} = \sum_j W_{ij}$. Based on the following Theorem 6.2 (whose derivations are at the end of the paper), the solution for the optimization problem in (6.1) can be expressed as a linear combination of the functionals $\{\check{k}(\mu_{\mathbb{P}_i}, \cdot)\}_{i=1}^l$ as

$$f^*(\mu_{\mathbb{P}}) = \sum_{i=1}^l \alpha_i \check{k}(\mu_{\mathbb{P}}, \mu_{\mathbb{P}_i}). \quad (6.3)$$

Theorem 6.2 (Representer Theorem for the proposed DSSL method). *Given l labeled distributions $\{(\mathbb{P}_1, y_1), \dots, (\mathbb{P}_l, y_l)\} \in \mathcal{P} \times \mathbb{R}$, a loss function $\ell : (\mathcal{P} \times \mathbb{R}^2)^l \rightarrow \mathbb{R} \cup \{+\infty\}$ and a strictly monotonically increasing real-valued function Ω on $[0, +\infty)$, the*

minimizer of the regularized risk functional

$$\ell(\mathbb{P}_1, y_1, \mathbb{E}_{\mathbb{P}_1}[f], \dots, \mathbb{P}_l, y_l, \mathbb{E}_{\mathbb{P}_l}[f]) + \Omega(\|f\|_{\tilde{\mathcal{H}}}), \quad (6.4)$$

admits an expansion $f = \sum_{i=1}^l \alpha_i \check{k}(\mu_{\mathbb{P}_i}, \cdot)$, where $\alpha_i \in \mathbb{R}$, for $i = 1, \dots, l$.

6.3 Detailed Proofs

6.3.1 Proof of Theorem 6.1

Let's start with $\tilde{\mathcal{H}}$ with the kernel $\tilde{k}$. Since $\tilde{\mathcal{H}}$ is a complete Hilbert space, and evaluation functionals therein are bounded, i.e., $\forall \mu \in \mathcal{H}, f \in \tilde{\mathcal{H}}, \exists C_\mu \in \mathbb{R}$, s.t. $|f(\mu)| \leq C_\mu \|f\|_{\tilde{\mathcal{H}}}$. Moreover, the bounded operator S is bounded by a constant D , i.e., $\|S\| = \sup_{f \in \tilde{\mathcal{H}}} \frac{\|Sf\|_{\mathcal{V}}}{\|f\|_{\tilde{\mathcal{H}}}} \leq D$. The complete $\tilde{\mathcal{H}}$ means every Cauchy sequence in the space converges to an element in $\tilde{\mathcal{H}}$. Let (f_n) be a Cauchy sequence in $\tilde{\mathcal{H}}$ converging to f , then $\forall \epsilon > 0, \exists$ an integer $N(\epsilon)$, s.t.

$$m > N(\epsilon), n > N(\epsilon) \Rightarrow \|f_m - f_n\|_{\tilde{\mathcal{H}}} < \frac{\epsilon}{\sqrt{1 + D^2}}.$$

Now let's turn to $\check{\mathcal{H}}$. We need to prove the completeness of the space first. According to the definition in Eq. (6.2), we obtain that for any Cauchy sequence in $\check{\mathcal{H}}$,

$$\begin{aligned} \|f_m - f_n\|_{\check{\mathcal{H}}}^2 &= \|f_m - f_n\|_{\tilde{\mathcal{H}}}^2 + \|S(f_m - f_n)\|_{\mathcal{V}}^2 \\ &\leq \|f_m - f_n\|_{\tilde{\mathcal{H}}}^2 + D^2 \|f_m - f_n\|_{\tilde{\mathcal{H}}}^2 \\ \Rightarrow \|f_m - f_n\|_{\check{\mathcal{H}}} &\leq \sqrt{1 + D^2} \|f_m - f_n\|_{\tilde{\mathcal{H}}} \\ &< \sqrt{1 + D^2} \times \frac{\epsilon}{\sqrt{1 + D^2}} = \epsilon. \end{aligned}$$

Hence $\check{\mathcal{H}}$ is complete since every Cauchy sequence in $\check{\mathcal{H}}$ converges to an element in $\check{\mathcal{H}}$. Moreover, $\check{\mathcal{H}}$ is bounded based on the property that any Cauchy sequence is bounded [Berlinet and Thomas-Agnan, 2011, Lemma 5]. This completes the proof.

6.3.2 Proof of Proposition 1

Firstly, we decompose $\check{\mathcal{H}}$ to two orthogonal parts as

$$\check{\mathcal{H}} = \text{span}\{\check{k}(\boldsymbol{\mu}_{\mathbb{P}_1}, \cdot), \dots, \check{k}(\boldsymbol{\mu}_{\mathbb{P}_l}, \cdot)\} \oplus \check{\mathcal{H}}^\perp,$$

where $\check{\mathcal{H}}^\perp$ vanishes at all labeled embedded distributions, i.e.,

$$\forall f \in \check{\mathcal{H}}^\perp, i \in \{1, \dots, l\}, f(\boldsymbol{\mu}_{\mathbb{P}_i}) = 0. \quad (6.5)$$

Accordingly $Sf = 0$, which means $\langle f, g \rangle_{\check{\mathcal{H}}} = \langle f, g \rangle_{\tilde{\mathcal{H}}}, \forall f \in \check{\mathcal{H}}^\perp, g \in \check{\mathcal{H}}$. Moreover,

$$\begin{aligned} f(\boldsymbol{\mu}_{\mathbb{P}}) &= \langle f, \check{k}(\boldsymbol{\mu}_{\mathbb{P}}, \cdot) \rangle_{\check{\mathcal{H}}} = \langle f, \tilde{k}(\boldsymbol{\mu}_{\mathbb{P}}, \cdot) \rangle_{\tilde{\mathcal{H}}} \\ &= \langle f, \tilde{k}(\boldsymbol{\mu}_{\mathbb{P}}, \cdot) \rangle_{\tilde{\mathcal{H}}} + \langle Sf, S\tilde{k}(\boldsymbol{\mu}_{\mathbb{P}}, \cdot) \rangle_{\mathcal{V}} \\ &= \langle f, \tilde{k}(\boldsymbol{\mu}_{\mathbb{P}}, \cdot) \rangle_{\tilde{\mathcal{H}}}. \end{aligned}$$

Thus, we have

$$\forall f \in \check{\mathcal{H}}^\perp, \langle f, \tilde{k}(\boldsymbol{\mu}_{\mathbb{P}}, \cdot) - \check{k}(\boldsymbol{\mu}_{\mathbb{P}}, \cdot) \rangle_{\check{\mathcal{H}}} = 0. \quad (6.6)$$

That is $\tilde{k}(\boldsymbol{\mu}_{\mathbb{P}}, \cdot) - \check{k}(\boldsymbol{\mu}_{\mathbb{P}}, \cdot) \in (\check{\mathcal{H}}^\perp)^\perp$. By substituting (6.5) into (6.6), we obtain $\tilde{k}(\boldsymbol{\mu}_{\mathbb{P}_i}, \cdot) \in (\check{\mathcal{H}}^\perp)^\perp, \forall i$, which means

$$\text{span}\{\tilde{k}(\boldsymbol{\mu}_{\mathbb{P}_i}, \cdot)\}_{i=1}^l \subseteq \text{span}\{\check{k}(\boldsymbol{\mu}_{\mathbb{P}_i}, \cdot)\}_{i=1}^l. \quad (6.7)$$

Secondly, we decompose $\tilde{\mathcal{H}}$ as $\tilde{\mathcal{H}} = \text{span}\{\tilde{k}(\boldsymbol{\mu}_{\mathbb{P}_i}, \cdot)\}_{i=1}^l \oplus \tilde{\mathcal{H}}^\perp$. Similarly, we have

$$\langle f, \tilde{k}(\boldsymbol{\mu}_{\mathbb{P}_i}, \cdot) \rangle_{\tilde{\mathcal{H}}} = 0, \forall f \in \tilde{\mathcal{H}}^\perp, \forall i \in \{1, \dots, l\}.$$

As $Sf = 0$, we have $\langle f, g \rangle_{\tilde{\mathcal{H}}} = \langle f, g \rangle_{\check{\mathcal{H}}}$, and

$$\begin{aligned} f(\boldsymbol{\mu}_{\mathbb{P}}) &= \langle f, \tilde{k}(\boldsymbol{\mu}_{\mathbb{P}}, \cdot) \rangle_{\tilde{\mathcal{H}}} = \langle f, \check{k}(\boldsymbol{\mu}_{\mathbb{P}}, \cdot) \rangle_{\check{\mathcal{H}}} \\ &= \langle f, \check{k}(\boldsymbol{\mu}_{\mathbb{P}}, \cdot) \rangle_{\tilde{\mathcal{H}}} + \langle Sf, S\check{k}(\boldsymbol{\mu}_{\mathbb{P}}, \cdot) \rangle_{\mathcal{V}} \\ &= \langle f, \check{k}(\boldsymbol{\mu}_{\mathbb{P}}, \cdot) \rangle_{\tilde{\mathcal{H}}}. \end{aligned}$$

Therefore, we have $\langle f, \check{k}(\boldsymbol{\mu}_{\mathbb{P}}, \cdot) - \tilde{k}(\boldsymbol{\mu}_{\mathbb{P}}, \cdot) \rangle_{\tilde{\mathcal{H}}} = 0$. Since $f \in \tilde{\mathcal{H}}^\perp$, it becomes $\langle f, \check{k}(\boldsymbol{\mu}_{\mathbb{P}}, \cdot) \rangle_{\tilde{\mathcal{H}}} = 0$, i.e., $\check{k}(\boldsymbol{\mu}_{\mathbb{P}}, \cdot) \in (\tilde{\mathcal{H}}^\perp)^\perp$. Therefore, we have

$$\text{span}\{\check{k}(\boldsymbol{\mu}_{\mathbb{P}_i}, \cdot)\}_{i=1}^l \subseteq \text{span}\{\tilde{k}(\boldsymbol{\mu}_{\mathbb{P}_i}, \cdot)\}_{i=1}^l. \quad (6.8)$$

Finally, by considering both (6.7) and (6.8), we conclude that the two spans are the same. This completes the proof.

6.3.3 Proof of Proposition 2

Based on Proposition 1, we have

$$\check{k}(\boldsymbol{\mu}_{\mathbb{P}}, \cdot) = \tilde{k}(\boldsymbol{\mu}_{\mathbb{P}}, \cdot) + \sum_{j=1}^n \beta_j(\boldsymbol{\mu}_{\mathbb{P}}) \tilde{k}(\boldsymbol{\mu}_{\mathbb{P}_j}, \cdot), \quad (6.9)$$

where the coefficients β_j depend on $\boldsymbol{\mu}_{\mathbb{P}}$. If we can obtain the exact formulation for β_j , then we can derive relations between two spaces by explicit forms. To find β_j , we use a system of linear equations generated by evaluating $\tilde{k}(\boldsymbol{\mu}_{\mathbb{P}_i}, \cdot)$ at $\boldsymbol{\mu}_{\mathbb{P}}$:

$$\begin{aligned} & \langle \tilde{k}(\boldsymbol{\mu}_{\mathbb{P}_i}, \cdot), \check{k}(\boldsymbol{\mu}_{\mathbb{P}}, \cdot) \rangle_{\tilde{\mathcal{H}}} \\ &= \langle \tilde{k}(\boldsymbol{\mu}_{\mathbb{P}_i}, \cdot), \tilde{k}(\boldsymbol{\mu}_{\mathbb{P}}, \cdot) + \sum_{j=1}^n \beta_j(\boldsymbol{\mu}_{\mathbb{P}}) \tilde{k}(\boldsymbol{\mu}_{\mathbb{P}_j}, \cdot) \rangle_{\tilde{\mathcal{H}}} \\ &= \langle \tilde{k}(\boldsymbol{\mu}_{\mathbb{P}_i}, \cdot), \tilde{k}(\boldsymbol{\mu}_{\mathbb{P}}, \cdot) \rangle_{\tilde{\mathcal{H}}} + \sum_{j=1}^n \beta_j(\boldsymbol{\mu}_{\mathbb{P}}) \langle \tilde{k}(\boldsymbol{\mu}_{\mathbb{P}_i}, \cdot), \tilde{k}(\boldsymbol{\mu}_{\mathbb{P}_j}, \cdot) \rangle_{\tilde{\mathcal{H}}} + \tilde{\mathbf{k}}_{\boldsymbol{\mu}_{\mathbb{P}_i}}^\top M \mathbf{g}, \end{aligned}$$

where $\tilde{\mathbf{k}}_{\boldsymbol{\mu}_{\mathbb{P}_i}}^\top = \left(\tilde{k}(\boldsymbol{\mu}_{\mathbb{P}_i}, \boldsymbol{\mu}_{\mathbb{P}_1}), \dots, \tilde{k}(\boldsymbol{\mu}_{\mathbb{P}_i}, \boldsymbol{\mu}_{\mathbb{P}_n}) \right)$ and $\mathbf{g}$ consists of components $\mathbf{g}_i = \tilde{k}(\boldsymbol{\mu}_{\mathbb{P}}, \boldsymbol{\mu}_{\mathbb{P}_i}) + \sum_{j=1}^n \beta_j(\boldsymbol{\mu}_{\mathbb{P}}) \tilde{k}(\boldsymbol{\mu}_{\mathbb{P}_j}, \boldsymbol{\mu}_{\mathbb{P}_i})$. Then we have the following linear equation for the coefficients $\beta(\boldsymbol{\mu}_{\mathbb{P}}) = (\beta_1(\boldsymbol{\mu}_{\mathbb{P}}), \dots, \beta_n(\boldsymbol{\mu}_{\mathbb{P}}))^\top$:

$$-M\tilde{\mathbf{k}}_{\boldsymbol{\mu}_{\mathbb{P}}} = (I + M\tilde{K})\beta(\boldsymbol{\mu}_{\mathbb{P}}). \quad (6.10)$$

Based on (6.9) and (6.10), we obtain the following explicit form for $\check{k}(\cdot, \cdot)$:

$$\check{k}(\boldsymbol{\mu}_{\mathbb{P}_i}, \boldsymbol{\mu}_{\mathbb{P}_j}) = \tilde{k}(\boldsymbol{\mu}_{\mathbb{P}_i}, \boldsymbol{\mu}_{\mathbb{P}_j}) - \tilde{\mathbf{k}}_{\boldsymbol{\mu}_{\mathbb{P}_i}}^\top (I + M\tilde{K})^{-1} M \tilde{\mathbf{k}}_{\boldsymbol{\mu}_{\mathbb{P}_j}}.$$

The above equation can be written in the following concise matrix form:

$$\check{K} = \tilde{K} - \tilde{K}(I + M\tilde{K})^{-1}M\tilde{K}. \quad (6.11)$$

It can be shown that by applying the Sherman-Morrison-Woodbury (SMW) identity, (6.11) can be further rewritten as

$$\check{K} = (I - \tilde{K}(I + M\tilde{K})^{-1}M)\tilde{K} = (I + \tilde{K}M)^{-1}\tilde{K}. \quad (6.12)$$

This completes the proof.

6.3.4 Proof of Theorem 6.2

Any functional $f \in \check{\mathcal{H}}$ can be uniquely decomposed into a component f_μ in the space spanned by the kernel mean embedding $f_\mu = \sum_{i=1}^l \alpha_i \check{k}(\mu_{\mathbb{P}_i}, \cdot)$, and a component $f_\perp$ orthogonal to it, i.e., $\langle f_\perp, \check{k}(\mu_{\mathbb{P}_j}, \cdot) \rangle = 0, \forall j \in \{1, \dots, l\}$. Therefore, we have

$$f = f_\mu + f_\perp = \sum_{i=1}^l \alpha_i \check{k}(\mu_{\mathbb{P}_i}, \cdot) + f_\perp.$$

Thus, for all j , we can further induce that

$$\begin{aligned} \mathbb{E}_{\mathbb{P}_j}[f] &= \left\langle \sum_{i=1}^l \alpha_i \check{k}(\mu_{\mathbb{P}_i}, \cdot) + f_\perp, \check{k}(\mu_{\mathbb{P}_j}, \cdot) \right\rangle \\ &= \left\langle \sum_{i=1}^l \alpha_i \check{k}(\mu_{\mathbb{P}_i}, \cdot), \check{k}(\mu_{\mathbb{P}_j}, \cdot) \right\rangle. \end{aligned}$$

This indicates the loss function term in (6.4) does not depend on $f_{\perp}$. Besides, the second term $\Omega(\cdot)$ in (6.4) is strictly monotonically increasing, so we have

$$\begin{aligned}\Omega(\|f\|_{\check{\mathcal{H}}}) &= \Omega\left(\left\|\sum_{i=1}^l \alpha_i \check{k}(\boldsymbol{\mu}_{\mathbb{P}_i}, \cdot) + f_{\perp}\right\|_{\check{\mathcal{H}}}\right) \\ &= \Omega\left(\sqrt{\left\|\sum_{i=1}^l \alpha_i \check{k}(\boldsymbol{\mu}_{\mathbb{P}_i}, \cdot)\right\|_{\check{\mathcal{H}}}^2 + \|f_{\perp}\|_{\check{\mathcal{H}}}^2}\right) \\ &\geq \Omega\left(\left\|\sum_{i=1}^l \alpha_i \check{k}(\boldsymbol{\mu}_{\mathbb{P}_i}, \cdot)\right\|_{\check{\mathcal{H}}}\right),\end{aligned}$$

where the equality holds if and only if $f_{\perp} = 0$. Therefore, the first term in (6.4) is independent of $f_{\perp}$ and the second term reaches its minimum when $f_{\perp} = 0$. Consequently, any minimizer must take the form $f = f_{\mu} = \sum_{i=1}^l \alpha_i \check{k}(\boldsymbol{\mu}_{\mathbb{P}_i}, \cdot)$. This completes the proof.

6.4 Experiments

6.4.1 Datasets

We conduct experiments on three groups of datasets. The statistics are listed in Table 6.2. The first group of datasets is on sensor-based activity recognition. Skoda dataset¹ records 10 gestures in car maintenance scenarios with 20 acceleration sensors being put on the arms of the subject [Stiefmeier et al., 2007]. Each gesture is repeated around 70 times. The transitions between two gestures are labeled as Null class, which are also considered as activities. WISDM dataset² uses accelerometer sensors embedded in the phones to collect six regular activities: jogging, walking, ascending stairs, descending stairs, sitting and standing [Kwapisz et al., 2010]. HCI dataset³ composes of gestures with the hand describing different shapes: a circle, a square, a pointing-up triangle, an upside-down triangle, and an infinity symbol [Förster et al., 2009]. Each

¹The dataset is available at <http://har-dataset.org/doku.php?id=wiki:dataset>.

²The dataset is available at <http://www.cis.fordham.edu/wisdm/dataset.php#actitracker>.

³The dataset is available at <http://har-dataset.org/doku.php?id=wiki:dataset>.

gesture is recorded over 50 repetitions, and about 5 to 8 seconds per repetition. Null class exists as well in HCI dataset. The second group of datasets is about drug detection, and the third group is about image annotation, which are commonly used for MIL approaches¹. In MUSK1 and MUSK2, a bag stands for a molecule and each instance inside is the alternative shape of the molecule. It is regarded as a positive bag if at least one of the alternative shapes could tightly bind to the target area of some target molecules. In Fox, Tiger and Elephant, each image is considered as a bag, containing a set of image regions characterized by color, texture and shape descriptors.

TABLE 6.2: Statistics of datasets used in experiments of Chapter 6.

Datasets	# Sample	# Instances per sample	# Feature	# Class
Skoda	1,447	68.81	60	10
WISDM	389	705	6	6
HCI	264	602	48	5
MUSK1	92	5.17	166	2
MUSK2	102	64.7	166	2
Fox	200	6.6	230	2
Tiger	200	6.1	230	2
Elephant	200	6.96	230	2

6.4.2 Experimental Setup

Following the criteria in [Qian et al., 2018], we adopt both micro- F_1 score (miF) and weighted macro- F_1 score (maF) to evaluate the performance of different methods. All the reported results are the average values together with the standard deviation over 6 random splits for training and testing. Each dataset is randomly split into 3 subsets: labeled training set, unlabeled training set and test set. Each subset is set to contain activities of all classes. We set the ratio to be 0.02:0.1:0.88 and fix $r = 100$. The impact of differentiating r will be discussed later. Different from experimental setups in existing papers that set labeled data’s ratio to be quite large [Matsushige et al., 2015, Stikic et al., 2009], we deliberately set the labeled data’s ratio to be extremely small. Hence, our method requires fewer labels and thus more practical with regards to applicability

¹Datasets are available at <http://www.cs.columbia.edu/~andrews/mil/datasets.html>.

in reality. Evaluations are conducted on the test set. We adopt RBF kernels for all the kernels used in the experiments.

6.4.3 Baselines

We compare the proposed DSSL method with the following state-of-the-art methods.

Sensor-based Activity Recognition Tasks

- State-of-the-art supervised methods with various features:
  - SVMs [Chang and Lin, 2011]: as SVM is a vectorial-based classifier, we use mean, variance, etc to generate a feature vector for each segment.
  - SAX- a [Lin et al., 2007b] treats data as strings, and structural features are extracted. We follow the settings in [Lin et al., 2007b] with no dimension reduction. The parameter alphabet_size range is $a \in \{3, 6, 9\}$.
  - ECDF- d [Hammerla et al., 2013, Plötz et al., 2011] extracts d descriptors from each sensor's each dimension. $d \in \{5, 15, 30, 45\}$.

Note that the overall shape and spatial features besides the mean and variance features are concatenated before applying the SVM classifier.

- State-of-the-art supervised method based on distributions, SMM_{AR} [Qian et al., 2018].
- Classic vectorial-based semi-supervised methods:
  - LapSVM [Belkin et al., 2006] is an extension of SVM with manifold regularization.
  - ∇ TSVM [Chapelle and Zien, 2005] is a Transductive SVM by using gradient descent for training. As this is a transductive approach rather than a truly semi-supervised learning approach, we make the test data available in the training phase of this method.
- State-of-the-art semi-supervised methods specifically designed for activity recognition:

TABLE 6.3: Experimental results of proposed semi-supervised methods as well as baselines on three activity datasets (unit: %).

Methods		Skoda		HCI		WISDM	
		miF	maF	miF	maF	miF	maF
Vectorial-based supervised	SVMs	85.7±1.8	42.5±0.9	69.7±9.6	69.6±9.4	41.5±5.2	39.6±6.8
	SAX_3	39.6±6.3	18.7±2.9	36.0±3.0	34.7±2.5	34.6±1.4	30.6±1.2
	SAX_6	37.2±6.1	18.6±2.8	39.7±7.3	38.4±7.9	34.9±3.0	30.5±5.0
	SAX_9	40.3±6.5	19.9±3.2	39.8±8.7	37.0±9.2	33.6±2.9	28.8±5.8
	ECDF_5	84.2±2.1	41.6±1.0	67.7±10.1	67.6±9.1	42.1±6.3	40.5±7.7
	ECDF_15	79.8±1.5	39.2±0.7	68.4±10.4	68.5±9.6	39.4±3.3	36.2±5.7
	ECDF_30	72.6±1.2	35.4±0.3	68.6±11.1	68.7±10.5	37.7±2.5	32.6±4.9
	ECDF_45	65.7±2.5	31.5±1.3	68.6±11.4	68.6±10.8	36.4±1.4	31.3±3.6
Vectorial-based semi-supervised	LapSVM	89.7±2.1	44.6±1.2	76.1±4.8	76.3±4.7	40.1±3.8	34.5±3.5
	▽TSVM	85.9±2.7	84.8±2.8	75.4±11.5	75.5±11.2	41.3±5.6	39.4±6.9
	SSKLR	25.4±19.3	12.1±2.5	24.2±17.2	18.1±10.1	24.6±17.0	17.3±9.9
	GLSVM	89.7±2.1	44.5±1.2	75.7±5.8	75.7±5.7	40.4±3.8	33.9±4.0
Distribution-based supervised	SMM _{AR}	93.2±0.9	93.1±1.0	82.2±13.4	78.9±18.4	20.5±3.3	11.7±3.9
Distribution-based semi-supervised	DSSL	98.8±0.5	98.8±0.5	99.9±0.2	99.9±0.2	56.5±5.1	55.6±5.0

- SSKLR [Matsushige et al., 2015] is a semi-supervised kernel logistic regression method with Expectation-Maximization algorithm.
- GLSVM [Stikic et al., 2009] is a multi-graph method where each graph captures different aspects of the activities.

Drug Activity Prediction and Image Annotation Tasks

Besides the above methods used in activity recognition tasks, we further compare S3MM with the following methods on the tasks of drug activity prediction and image annotation: 1) kernel-based methods including SIL, STK [Gärtner et al., 2002], MISVM, miSVM [Andrews et al., 2002], MissSVM [Zhou and Xu, 2007], 2) sparse variants of MIL including sMIL, stMIL, sbMIL [Bunescu and Mooney, 2007], and 3) semi-supervised MIL including semi-MIL [Zhou and Ming, 2016] and MISSL [Rahmani and Goldman, 2006].

6.4.4 Experimental Results

6.4.4.1 Overall Experimental Results

The experimental results are presented in Table 6.3. The proposed DSSL consistently performs the best on all datasets. DSSL outperforms all the other methods by 5.6%, 17.7%, and 14.4% respectively on three datasets in terms of miF. This favorably indicates the effectiveness of the proposed DSSL. Note that in Table 6.3, the performances

TABLE 6.4: Comparison results on drug activity prediction and image annotation tasks (unit: %).

Methods	MUSK1		MUSK2		Fox		Tiger		Elephant	
	miF	maF	miF	maF	miF	maF	miF	maF	miF	maF
DSSL	67.3±3	66.5±3	72.2±4	70.7±3	58.4±3	56.6±4	68.6±4	68.5±4	68.1±4	67.5±4
SMM	53.1±5	40.5±9	53.5±10	47.0±11	50.7±2	38.3±8	52.6±6	38.1±12	53.1±5	41.1±10
SVM	59.9±4	55.7±8	58.5±6	58.1±6	51.3±3	48.4±4	58.8±8	57.2±7	49.3±8	48.2±8
LapSVM	62.3±10	60.7±11	67.4±11	64.5±13	57.0±2	55.8±1	62.5±6	61.4±6	57.7±4	56.2±6
∇ TSVM	63.6±3	63.2±3	62.8±4	61.6±4	52.8±3	52.5±3	59.7±4	59.6±4	55.6±5	55.2±5
MissSVM	55.8±4	49.7±8	63.9±2	55.9±6	53.4±3	46.3±7	54.4±4	49.6±6	56.1±5	50.1±10
MISVM	55.1±5	47.8±10	66.3±2	61.9±6	52.1±3	42.6±8	54.1±4	49.1±6	59.3±10	54.4±15
miSVM	53.5±6	46.9±8	47.8±13	38.7±17	55.4±3	51.8±5	54.7±3	45.8±3	57.5±8	53.1±12
sMIL	55.3±7	48.5±14	62.2±0	49.1±3	50.4±1	40.1±7	50.0±0	34.1±1	53.2±4	41.2±9
stMIL	54.7±6	48.8±13	62.2±0	51.7±6	50.1±0	35.3±2	50.0±0	33.5±1	52.1±3	38.6±6
sbMIL	55.1±6	45.9±11	63.9±2	54.4±6	52.8±3	45.0±9	51.7±3	41.9±6	55.0±7	46.3±12
STK	54.1±6	49.6±6	49.1±6	48.6±8	48.7±5	46.9±5	55.8±9	54.6±8	53.7±6	52.7±6
SIL	53.5±6	46.9±8	47.8±13	38.7±17	55.4±3	51.8±5	54.7±3	45.8±3	57.5±8	53.1±12
semi-MIL	53.9±5	47.1±7	54.6±6	51.5±6	50.4±1	43.7±6	52.4±4	43.1±11	50.3±0	38.7±7
MISSL	50.6±0	34.0±0	62.2±0	47.7±0	50.0±0	33.3±0	50.0±0	33.3±0	50.0±0	33.3±0

of the comparison methods on WISDM are much worse than those on the other two datasets. This may be due to the data complexity caused by the large number of subjects in WISDM. On datasets Skoda and HCI, the performance ranking is $DSSL > SMM_{AR} > SVMs \approx ECDF > SAX$, which reveals that 1) distribution-based methods are more capable of distinguishing different activities; 2) feature extraction plays an important role and string-based data representation in SAX is not that proper for activity data compared to ECDF; 3) with the increase of descriptor d , the performance of ECDF is increasing in HCI dataset while decreasing in Skoda and WISDM, meaning ECDF may be task-dependent. However, note that SMM_{AR} performs the worst on WISDM dataset, which illustrates that distribution-based methods are more dependent on the number of labeled data than vectorial-based methods. This indeed reflects the motivation of our proposed method. Nevertheless, DSSL does not suffer from this limitation ascribed to its semi-supervised fashion. For semi-supervised methods, the ranking is $DSSL > LapSVM \approx GLSVM \approx \nabla TSVM > SSKLR$, which demonstrates the prevalence of graph-based methods over logistic regression method for activity data. For the two MIL tasks, our proposed method performs the best as shown in Table 6.4, once again demonstrating S3MM's capability to extract discriminative information from bags for classification. LapSVM and $\nabla TSVM$ perform consistently better than SMM and SVM, revealing the benefits of learning from unlabeled bags.

FIGURE 6.1: Impact of varying ratios of labeled data in semi-supervised learning.

6.4.4.2 Impact of Ratio of Labeled Data

To analyze the impact on the proportion of labeled training data, we conduct experiments on WISDM dataset. We fix the ratio of test data and unlabeled training data to be 20% and 20% respectively, and alter the ratio of labeled training data to be $\{0.02, 0.05, 0.1, 0.3, 0.5, 0.7, 0.9\}$ of the rest 60% data. The results are depicted in Figure 6.1. DSSL performs the best under all the ratios. When more labeled training data becomes available, all methods perform better. Moreover, distributional-based method (SMM_{AR}) has larger performance enhancement than vectorial-based methods, which further verifies the superiority of learning from distributions.

6.4.4.3 Impact of Ratio of Unlabeled data

We investigate the influence of unlabeled data by fixing the ratio of labeled training data and test data to be 1% and 20%, respectively, and modifying unlabeled training data to be $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ of the remaining 79% data. Note that supervised methods (SMM_{AR} , SVMs) and transductive methods (∇TSVM) perform the same under this setting, while the performances of semi-supervised methods keep increasing with more unlabeled training data as shown in Figure 6.2.

FIGURE 6.2: Impact of varying ratios of unlabeled data in semi-supervised learning.

FIGURE 6.3: Impact of r to the performance of proposed DSSL method.

6.4.4.4 Impact of Parameter r

In previous experiments, we fix $r = 100$. Here we conduct sensitivity test on r . As indicated in Fig. 6.3, the performance of DSSL on test data keeps stable when $r \in [10^{-6}, 1]$. When r becomes larger, the performance of DSSL begins to decrease. This observation indicates that r balances the tradeoff between labeled and unlabeled data. Larger r implies stronger emphasis on unlabeled data. More importantly, under all

different r values, DSSL consistently outperforms all other methods. Fig. 6.3 shows the best baseline, i.e., ECDF_5 in WISDM’s case.

6.4.4.5 Impact on Random Fourier Feature (RFF) Dimension D

FIGURE 6.4: Impact of D to the performance on WISDM in semi-supervised learning.

We analyze how R-DSSL accelerates DSSL with D -dimensional explicit statistical features. The experiments are conducted on a Linux server with Intel(R) Xeon(R) E5-2695 2.40GHz CPU. As shown in Fig. 6.4, R-DSSL steadily outperforms the best baseline when $D \geq 2$. Note that R-DSSL performs slightly worse than DSSL due to its approximation nature, however it requires less computational run time when $D < 8$ compared to DSSL.

6.5 Summary

In this chapter, we propose a semi-supervised learning framework named Distribution-based Semi-Supervised Learning (DSSL), for sensor-based activity recognition problems. The proposed DSSL naturally embeds automatic feature extraction and classification in a semi-supervised learning manner. Extensive evaluations are conducted on

three activity datasets to demonstrate the superiority of DSSL compared with a number of state-of-the-art methods.

Chapter 7

Weakly-Supervised Sensor-based Activity Segmentation and Recognition via Learning from Distributions¹

7.1 Overview

Sensor-based activity recognition aims to predict users' activities from multi-dimensional streams of various sensor readings received from ubiquitous sensors. An end-to-end solution to this task consists of two steps: performing segmentation on multivariate streams of sensory readings and learning a classifier for activity recognition. Most previous studies focused on the latter step to manually design features for each segment of sensory readings based on its statistical or structural information by either assuming the segmentation is given in advance or using simple sliding windows techniques. In this chapter, we argue that most existing segmentation methods often fail to segment activities of variable lengths properly. Moreover, some important information, e.g., statistical information captured by higher-order moments, may be discarded when manually constructing features in previous approaches.

Therefore, we propose a unified weakly-supervised framework to jointly segment sensor streams and extract infinite-dimensional statistical features of sensory readings

---

¹Partial results of the presented work have been submitted to Artificial Intelligence Journal, 2019.

of each segment based on kernel embedding of distributions for learning an activity recognition classifier. We named our proposed algorithm $S-SMM_{AR}$. To scale-up the proposed method, we further offer an accelerated version, denoted by $R-SMM_{AR}$, 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.

To summarize, our contributions are 4-fold:

- We model the weakly-supervised segmentation problem of activity data as a non-convex optimization problem, and propose a novel iterative kernel-based method to solve it. The segmentation method together with a novel feature extraction method are integrated into a unified framework $S-SMM_{AR}$ that enables jointly learning of segmentation, feature extraction and classification for sensor-based activity recognition.
- We study the feasibility of existing general time series segmentation methods in the scenario of activity data.
- We propose an accelerated method, denoted by $R-SMM_{AR}$, to scale up the proposed method.
- Extensive evaluations are conducted to demonstrate the efficacy of the proposed method compared with state-of-the-art methods.

Note that in our preliminary work on applying SMM for activity recognition [Qian et al., 2018], we assumed that a perfect partition on each time series of sensory readings is given in advance. Thus, it is not an end-to-end solution. In this paper, we extend it to an end-to-end solution by integrating a segmentation module and a classifier into a unified framework, which is more practical in real-world scenarios.

7.2 The Proposed Methodology

7.2.1 Problem Statement

In our problem setting, we are given a stream of multivariate activity data $\mathbf{X} = \{\mathbf{x}_t\}_{t=1}^N$ where $\mathbf{x}_t \in \mathbb{R}^{d \times 1}$ is a vector of signals received from d sensors at the t -th timestamp, which is referred to as a frame in the segment. Associated with signals are a sequence of K activity labels $\mathbf{y} = \{y_k\}_{k=1}^K$, where $y_k \in \{Y_1, \dots, Y_L\}$ is a set of predefined L activity categories. Each y_k may last for n_k timestamps, and n_k can be different, with the sum of all the n_k 's to be equal to the total duration N . This setting is referred to as a weakly supervised setting, as no ground-truth partition and ground-truth label on each segment is provided in training, while only a sequence of activity labels is available for the whole data stream. Note that this setting is more practically applicable, since human usually can remember effortlessly the sequences of activities conducted in a time period, but the exact starting and ending time require expensive annotation effort.

Our goal is to first find $K - 1$ breakpoints indices $\mathbf{I} = \{I_k | 1 < I_k < N, I_k < I_{k+1}\}_{k=1}^{K-1}$ to segment the stream of activity data into K adjacent segments $\{\mathbf{X}_k\}_{k=1}^K$, where $\mathbf{X}_k = \{\mathbf{x}_{I_{k-1}+1}, \dots, \mathbf{x}_{I_k}\}$ such that each segment $\mathbf{X}_k$ is aligned with an activity y_k of the sequences of activities sequentially. With the K segments, each of which, $\mathbf{X}_i$, is aligned with a label $y_i \in \{Y_1, \dots, Y_L\}$, we aim to train a classifier f to map $\{\mathbf{X}_i\}$'s to $\{y_i\}$'s.

For testing, we suppose the segmentation is done, and we are given m new unseen segments $\{\mathbf{X}_i^*\}_{i=1}^m$, each of which corresponds to an unknown label. We use the trained classifier f to make predictions.

7.2.2 Problem Formulation in Weakly-Supervised Setting

In weakly-supervised setting, given the data stream $\mathbf{X} = \{\mathbf{x}_t\}_{t=1}^N$, the ground-truth labels on each segment as well as breakpoints indices $\mathbf{I}$ are unknown, while only the sequences of activities $\mathbf{y} = \{y_k\}_{k=1}^K$ are available, where K is the total number of activity segments in the stream consisting of L classes of activities. We propose the

following optimization problem to jointly learn the classifier f and the segmentation in terms of $\mathbf{I}$,

$$\begin{aligned} \min_{f, \mathbf{I}, \mathbf{C}} \quad & \frac{1}{K} \sum_{k=1}^K \sum_{j=1}^L \mathbf{C}_{kj} \ell(f(\mathbf{X}_k), Y_j; \mathbf{I}) + \lambda_1 \Omega_1(\|f\|_{\tilde{\mathcal{H}}}) + \lambda_2 \Omega_2(\mathbf{I}), \\ \text{s.t.} \quad & f(\mathbf{X}_k) = y_k, \forall k \in \{1, \dots, K\}, \\ & \sum_{j=1}^L \mathbf{C}_{kj} = 1, \forall k \in \{1, \dots, K\}, \end{aligned} \quad (7.1)$$

where $\ell(\cdot)$ is a data-dependent loss function, $\lambda_1, \lambda_2 > 0$ are the tradeoff parameters to control the impact of the regularization terms $\Omega_1(\cdot)$ and $\Omega_2(\cdot)$. $\tilde{\mathcal{H}}$ is a RKHS associated with the kernel $\tilde{k}(\cdot, \cdot)$, which will be explained later. The first term in the objective is the weighted average loss function on classification, and $\mathbf{C} \in \mathbb{R}^{K \times L}$ is the matrix of confidence scores, with each element $\mathbf{C}_{kj}$ being the confidence score of the k -th segment associated with the j -th activity class. The confidence score matrix leads the classifier f to correctly learn those easy-to-classify segments first. A higher confidence score of a segment means a higher probability of a correct prediction by the classifier. Therefore, the classifier tends to predict the corresponding label correctly, or the weighted loss function is increased by a larger value compared to those with smaller confidence scores. The second term is a regularization term on the learned classifier to prevent overfitting. The the form of $\Omega_1(\cdot)$ is chosen to be a strictly monotonically increasing function, with a special choice being the linear function as used in our previous work [Qian et al., 2018]. The last term is the regularization term on segmentation breakpoints to ensure the segmentation results to be reasonable, and is set to be the average of the MMD distance between segments with the same predicted label:

$$\begin{aligned} \Omega_2(\mathbf{I}) &= \frac{1}{M} \sum_{\substack{1 \leq i < j \leq N \\ f(\mathbf{X}_i) = f(\mathbf{X}_j)}} \text{MMD}(\mathbf{X}_i, \mathbf{X}_j) \\ &= \frac{1}{M} \sum_{\substack{1 \leq i < j \leq N \\ f(\mathbf{X}_i) = f(\mathbf{X}_j)}} \left\| \frac{1}{n_i} \sum_{k=1}^{n_i} (\phi(\mathbf{x}_k^i)) - \frac{1}{n_j} \sum_{k=1}^{n_j} (\phi(\mathbf{x}_k^j)) \right\|_2 \end{aligned} \quad (7.2)$$

$$= \frac{1}{M} \sum_{\substack{1 \leq i < j \leq N \\ f(\mathbf{X}_i) = f(\mathbf{X}_j)}} \left(\frac{1}{n_i^2} \sum_{k_1, k_2} k(x_{k_1}^i, x_{k_2}^j) - \frac{2}{n_i n_j} \sum_{k_1, k_2} k(x_{k_1}^i, x_{k_2}^j) + \frac{1}{n_j^2} \sum_{k_1, k_2} k(x_{k_1}^i, x_{k_2}^j) \right)^{\frac{1}{2}},$$

where $\mathbf{x}_i^k$ denotes the k -th instance in the i -th segment $\mathbf{X}_i$, and n_i is the length of segment $\mathbf{X}_i$. The kernel $k(\cdot, \cdot)$ is induced by the feature map $\phi(\cdot)$.

The first constraint in (7.1) is to enforce that the sequence of predicted labels is aligned with the sequence of the ground-truth activities. The second constraint is to ensure that the summation of the confidences over all possible classes for each segment equals 1.

7.2.3 Alternating Optimization for Joint Segmentation and Classification

Note that the optimization problem (7.1) is a joint learning framework, where the breakpoints influence the formation of segments of activities, while the predicted labels further influence the detection of breakpoints. Therefore, in this section, we propose an alternating optimization algorithm to solve the problem. In the sequel, we denote our proposed joint learning algorithm for activity segmentation and classification by S-SMM_{AR}. The overall algorithm is shown in Algorithm 1.

Algorithm 1: The proposed S-SMM_{AR} algorithm

- Input:** A data sequence $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \in \mathbb{R}^{d \times 1}$, a coarse label sequence $\mathbf{y} = \{y_k\}_{k=1}^K \in \{Y_1, \dots, Y_L\}$, the number of breakpoints $K - 1$
- Output:** the breakpoints indices $\mathbf{I}$ and the classifier f
- 1: Randomly initialize breakpoints indices $\mathbf{I} = \{I_k\}_{k=1}^{K-1}$ and the matrix of confidence scores $\mathbf{C}$
  - 2: **while** not convergent **do**
  - 3:   Fix $\mathbf{I}$ and $\mathbf{C}$, update f with (7.6)
  - 4:   Fix f , update $\mathbf{C}$ as described in the 2nd paragraph in Section 7.2.3.2
  - 5:   Update candidate range of breakpoints with (7.10) and (7.11)
  - 6:   Fix f and $\mathbf{C}$, update $\mathbf{I}$ by solving (7.9)
  - 7: **end while**
  - 8: **return** $\mathbf{I}$, f , and $\mathbf{C}$
-

7.2.3.1 Learning the classifier f with fixed $\mathbf{I}$ and $\mathbf{C}$

With $\mathbf{I}$ and $\mathbf{C}$ fixed, the K segmentations $\mathbf{X}_i$'s from $\mathbf{X}$ are known, and their corresponding class labels are also known by aligning them with the sequence of activities $\mathbf{y}$. Therefore, the optimization problem (7.1) is reduced as the following unconstrained optimization problem,

$$\min_f \frac{1}{K} \sum_{k=1}^K \mathbf{C}_{k\hat{k}} \ell(f(\mathbf{X}_k), y_k) + \lambda_1 \Omega_1(\|f\|_{\tilde{\mathcal{H}}}), \quad (7.3)$$

where $\hat{k}$ is the index of y_k in $\{Y_1, \dots, Y_L\}$.

To construct a classifier, in most standard classification methods, the input is required to be 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, the sizes of the different segments can be different. 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 $\{\mathbf{x}_k^i\}_{k=1}^{n_i}$, and assign the same label y_i to each vector. In this way, for each segment, one can construct n_i input-output pairs $\{(\mathbf{x}_k^i, y_i)\}_{k=1}^{n_i}$. By combining such input-output pairs from all the segments, one can apply standard classification methods to train a classifier f . 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 $\bar{\mathbf{x}}_i = \sum_{k=1}^{n_i} \mathbf{x}_k^i$ 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.

To be specific, firstly, each segment or sample $\mathbf{X}_i$ is mapped to a RKHS with a kernel $k(\mathbf{x}_{k_1}^i, \mathbf{x}_{k_2}^i) = \langle \phi(\mathbf{x}_{k_1}^i), \phi(\mathbf{x}_{k_2}^i) \rangle$ via an implicit feature map $\phi(\cdot)$, and represented by an element $\boldsymbol{\mu}_i$ in the RKHS via the mean map operation:

$$\boldsymbol{\mu}_i = \frac{1}{n_i} \sum_{k=1}^{n_i} \phi(\mathbf{x}_k^i). \quad (7.4)$$

As a result, we have K pairs of input-output in the RKHS $\{(\boldsymbol{\mu}_1, y_1), \dots, (\boldsymbol{\mu}_K, y_K)\}$. Then our goal becomes to learn a classifier f by solving

$$\min_f \frac{1}{K} \sum_{k=1}^K C_{k\hat{k}} \ell(f(\boldsymbol{\mu}_k), y_k) + \lambda_1 \Omega_1(\|f\|_{\tilde{\mathcal{H}}}). \quad (7.5)$$

As shown in our preliminary work [Qian et al., 2018], by using the representer theorem in [Muandet et al., 2012], the solution of the functional $f(\cdot)$ in (7.5) can be represented by

$$f = \sum_{i=1}^K \alpha_i \psi(\boldsymbol{\mu}_i), \quad (7.6)$$

where the weights $C_{k\hat{k}}$'s are incorporated into α_i 's, the feature map $\psi : \mathcal{H} \rightarrow \tilde{\mathcal{H}}$ is used for classification, and $\tilde{\mathcal{H}}$ is another RKHS with a kernel $\tilde{k}(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) = \langle \psi(\boldsymbol{\mu}_i), \psi(\boldsymbol{\mu}_j) \rangle$ defined by $\psi(\cdot)$. If $\tilde{\mathcal{H}} = \mathcal{H}$, then a linear kernel on $\{\boldsymbol{\mu}_i\}$'s is used, i.e., $\tilde{k}(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) = \langle \boldsymbol{\mu}_i, \boldsymbol{\mu}_j \rangle$, and (7.6) can be reduced as

$$f = \sum_{i=1}^K \alpha_i \boldsymbol{\mu}_i, \text{ where } \alpha_i \in \mathbb{R}. \quad (7.7)$$

By specifying (7.6) or (7.7) using the Support Vector Machines (SVMs) formulation¹, we reach the following optimization problem, which is known as Support Measure Machines (SMMs) [Muandet et al., 2012],

$$\begin{aligned} \min_f \quad & \frac{1}{2} \|f\|_{\tilde{\mathcal{H}}}^2 + C \sum_{i=1}^K \xi_i, \\ \text{s.t.} \quad & y_i f(\boldsymbol{\mu}_i) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \\ & 1 \leq i \leq K, \end{aligned} \tag{7.8}$$

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 specified², 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 $\mathbf{X}_p^*$, one can first represent it using the mean map operation

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

and then use $f(\cdot)$ to make a prediction $f(\boldsymbol{\mu}_k^*)$.

7.2.3.2 Update I and C with fixed f

After obtaining the updated classifier f , we now show how to update $\mathbf{I}_{\text{bkps}}$ and $\mathbf{C}$. With f fixed, the optimization problem (7.1) becomes

$$\begin{aligned} \min_{\mathbf{I}, \mathbf{C}} \quad & \frac{1}{K} \sum_{k=1}^K \sum_{j=1}^L \mathbf{C}_{kj} \ell(f(\boldsymbol{\mu}_k), Y_j; \mathbf{I}) + \lambda_2 \Omega_2(\mathbf{I}), \\ \text{s.t.} \quad & f(\mathbf{X}_k) = y_k, \forall k \in \{1, \dots, K\}, \end{aligned} \tag{7.9}$$

¹Note that one can also specify (7.6) or (7.7) using other loss functions, which result in different particular approaches.

²Recall 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.

$$\sum_{j=1}^L \mathbf{C}_{kj} = 1, \forall k \in \{1, \dots, K\},$$

where the regularization term $\Omega_2(\mathbf{I})$ is defined in (7.2).

Regarding updating the matrix $\mathbf{C}$, the confidence score $\mathbf{C}_{kj}$ is expected to measure the confidence of the segment X_k that belongs to the class Y_j . In the supervised setting where the ground-truth labels are available, the confidence score can be easily obtained by calculating the accuracy of predicted labels of each segment. However, as we discussed, the annotation effort on segmentation is highly expensive as the exact start and end time stamps of each activity need to be marked for training. In our proposed weakly-supervised setting, where we only have access to the coarse activities sequence, we aim to make the confidence score of a predicted segment depending on the distance to the decision boundary. Specifically, for classification of L classes activities, a common practice is to learn L classifiers by one-vs-rest mechanism, which transforms the problem into learning multiple binary classifiers. For each binary classifier, the distance of the data point to the decision boundary matters in the way that a larger distance reflects the easier classification of the data point. Therefore, we set the confidence score to be $\frac{1}{1+\exp(Af(\mu)+B)}$ in the binary case, and further normalize the scores in the multi-class case. The confidence score is similar to the Platt's probabilistic output [Lin et al., 2007a], where A and B are decided by the data distribution prior.

Regarding updating $\mathbf{I}$, Dynamic Programming (DP) can be applied to find breakpoints one by one sequentially, but the candidate range of a new breakpoint is from the former breakpoints to the end of a time series, which is computationally expensive. Therefore, in the literature, various algorithms have been proposed to alleviate the computational cost by limiting the searching range of each breakpoint. Specifically, the computational cost is alleviated by pruning the set of candidate breakpoint locations and finding the next breakpoint in the restricted set. However, as discussed previously, existing algorithms are supposed to work under non-trivial assumptions on the data property or the model.

In our proposed algorithm, we also aim to prune the candidate set to reduce the complexity, but we do not have any assumptions on the data or the model. Different from

other pruning methods for DP, our method prunes the candidates set from a probabilistic point of view. Specifically, for each segment $\mathbf{X}_k$ with breakpoints indices $I_{k-1} + 1$ and I_k being the starting and ending locations, respectively, there is a vector of confidence scores, $\mathbf{C}_{k*}$ (the k -th row of $\mathbf{C}$), to represent the probabilities over all the classes for this segmentation. A larger $\mathbf{C}_{kj}$ indicates the higher probability of the segment k that belongs to the class Y_j . Intuitively, for a good segment, there should exist a i such that the corresponding confidence score $\mathbf{C}_{ki}$ is large and all the other confidence scores $\{\mathbf{C}_{kj}, i \neq j\}$'s are small. Thus, we set the confidence score of the segment k to be the maximum of $\{\mathbf{C}_{kj} | 1 \leq j \leq L\}$, i.e., $\max_j \mathbf{C}_{kj}$. Our proposed method prunes the candidate range of a breakpoint with its neighbors' status, i.e., the candidate range of I_k is the range of low confidence score neighbors with different labels $[\mathbf{I}_{left}, \mathbf{I}_{right}]$:

$$\mathbf{I}_{left} = \max(\mathbf{I}_m | m < k, y_m \neq y_k, \text{ and } \max_j(\mathbf{C}_{mj}) < \epsilon), \quad (7.10)$$

and

$$\mathbf{I}_{right} = \min(\mathbf{I}_m | m > k, y_m \neq y_k, \text{ and } \max_j(\mathbf{C}_{mj}) < \epsilon), \quad (7.11)$$

where ϵ is a threshold. In the next iteration, the location indices with lower confidence scores are more likely to be modified. And the breakpoints indices with high confidence scores are kept unchanged. In this way, the complexity of DP is reduced by pruning the candidate sets of breakpoints as well as reducing the number of modified breakpoints.

After specifying the candidate range of breakpoints, the next step is to go through each candidate range to search for an updated breakpoint location for each of the modified breakpoints by minimizing the optimization problem (7.9) with the updated $\mathbf{C}$ fixed. Note that the computational cost of the regularization term in (7.9) can be further reduced by reusing the precomputed kernel values in the previous classifier training step.

Precise segmentation of activity stream data is an essential prerequisite for learning an accurate classifier. Once the segmentation of activity data is corrupted, the extracted features are no longer representative for the corresponding activity class, hence the learning process of classification would be hindered. From another perspective, a

low similarity measure (as shown in (3.17)) between two segments from the same activity indicate two possibilities: 1) the classifier is not trained properly, and/or 2) the data is not segmented correctly. Thus, with the same similarity measure applied in both the segmentation and the prediction phases, the two modules can interact and boost each other in an iterative manner.

Our proposed segmentation algorithm is inspired by the e-SVM method [Zhu et al., 2014]. However, our proposed method is different from e-SVM in the following aspects: 1) an e-SVM can be readily solved by existing standard SVM solvers, while our problem involves integer variables and thus is non-convex, therefore we propose a novel DP-based method to solve the problem; 2) an e-SVM is actually a binary classification problem for each pixel in an image, while ours is a joint segmentation and prediction problem; 3) e-SVM adopts a linear function as prediction function, however we can use either linear or nonlinear kernel functions to learn a more precise classifier.

7.2.4 $\mathbf{R-SMM}_{AR}$ for Large-Scale Activity Recognition

Note that the technique of kernel embedding of distributions used in $\mathbf{S-SMM}_{AR}$ makes a feature vector of each segment be able to capture sufficient statistics of the segment. This is 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 $\mathbf{S-SMM}_{AR}$ computationally expensive when the number of segments is large and/or the number of frames within each segment is large. To scale up $\mathbf{S-SMM}_{AR}$, 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 (7.4) and (3.18), the empirical kernel mean map on a segment $\mathbf{X}_i$ with explicit Random Fourier Features can be written by

$$\boldsymbol{\mu}_i = \frac{1}{n_i} \sum_{k=1}^{n_i} \mathbf{z}(\mathbf{x}_k^i).$$

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

$$f(\mu_i) = \mathbf{w}^\top \mu_i. \quad (7.12)$$

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

$$f(\mu_i) = \mathbf{w}^\top \tilde{\mathbf{z}}(\mu_i), \quad (7.13)$$

where $\tilde{\mathbf{z}} : \mathbb{R}^D \rightarrow \mathbb{R}^{\tilde{D}}$ is another mapping of Random Fourier Features. (7.12) is a special case of (7.13) when $\tilde{\mathbf{z}}$ is an identity mapping. The resultant optimization problem on learning a classifier is reformulated accordingly as follows,

$$\min_{\mathbf{w} \in \mathbb{R}^{\tilde{D}}} \frac{1}{n} \sum_{k=1}^K \mathbf{C}_{k\hat{k}} \ell(\mathbf{w}^\top \tilde{\mathbf{z}}(\mu_k), y_k) + \lambda \|\mathbf{w}\|_2^2. \quad (7.14)$$

As $\tilde{\mathbf{z}}(\cdot)$ is an explicit feature map, standard linear SVMs solvers can be applied to solve (7.14), which is much more efficient than solving (7.8). Accordingly, in the sequel, we denote this accelerated version of S-SMM_{AR} with Random Fourier Features by R-SMM_{AR}.

7.3 Experiments

In this section, we investigate three different experimental settings: 1) different segmentation methods with fixed feature extraction; 2) joint segmentation and classification scenario; 3) feature extraction and classification under the perfect segmentation scenario. We conduct comprehensive experiments on four real-world activity recognition datasets to evaluate the effectiveness and scalability of our proposed S-SMM_{AR} and its accelerated version R-SMM_{AR}.

7.3.1 Datasets

The overall statistics of the four benchmark datasets used in our experiments are listed in Table 7.1.

Datasets	# Seg.	# En.	# Fea.	# C.	freq	# Sub.	$\frac{\#Seg.}{\#C.}$
Skoda	1,447	68.8	60	10	14	1	144.7
WISDM	389	705.8	6	6	20	36	64.8
HCI	264	602.6	48	5	96	1	52.8
PS	1,614	100.0	9	6	50	4	269

TABLE 7.1: Statistics of the four datasets for joint segmentation and classification.

Note that in the table, “Seg.” denotes segments, “En.” denotes average number of frames per segment, “Fea.” denotes feature dimensions, “C.” denotes classes, “freq” 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), “Sub.” denotes subjects, and “ $\frac{\#Seg.}{\#C.}$ ” denotes the average number of segments that each class of activity has.

Skoda [Stiefmeier et al., 2007]¹ 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 standing².

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 seconds³.

PS is collected by four smartphones on four body positions: [Shoaib et al., 2013]. The smartphones are embedded with accelerometers, magnetometers and gyroscopes.

¹The dataset is available at <http://har-dataset.org/doku.php?id=wiki:dataset>.

²The dataset is available at <http://www.cis.fordham.edu/wisdm/dataset.php#actitracker>.

³The dataset is available at <http://har-dataset.org/doku.php?id=wiki:dataset>.

Four participants were asked to conduct six activities for several minutes: walking, running, sitting, standing, walking upstairs and downstairs¹.

7.3.2 Evaluation Metric

For segmentation, we adopt an indicator *Ind* to indicate whether the method can find the exact correct number of breakpoints. We also adopt the rand index [Truong et al., 2018] as measurement. Specifically, the rand index is able to measure the similarity between two segmentation solutions for time series data $\{y_t\}_{t=1}^T$, i.e., the ground truth segmentation $\mathcal{S}$ and an estimated solution $\tilde{\mathcal{S}}$. The rand index (denoted by *RI*) is:

$$RI = \frac{\sum_{i < j} \mathbf{1}(A_{ij} = \tilde{A}_{ij})}{T(T-1)/2},$$

where A is the associated membership matrix for $\mathcal{S}$, and the entry of the matrix $A_{ij} = 1$ if both y_i and y_j are in the same segment, and $A_{ij} = 0$ otherwise. The membership matrix $\tilde{A}$ is constructed similarly based on the estimated solution $\tilde{\mathcal{S}}$.

We adopt the F_1 score as our evaluation metric for classification. 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,

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

---

¹The dataset is available at <https://www.utwente.nl/en/eemcs/ps/research/dataset/>.

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.

7.3.3 Experiments for Segmentation

7.3.3.1 Experimental Setup

In this section, we compare the segmentation performance of our proposed method with several state-of-the-art baselines. The feature extraction is fixed, and our proposed feature extraction method is applied after segmentation. There is no splitting of training and testing phase in this experiment. All the raw data as well as the coarse label sequence of the activities are available, and the goal is to decide the changepoints between each activity.

7.3.3.2 Baselines

We compare our proposed method with the following state-of-the-art methods.

- Binseg [Fryzlewicz et al., 2014]: binary segmentation method, which finds one breakpoint in the dataset first, then splits the data into two subsegments, and the same procedure is applied recursively to subsegments.
- BottomUp [Keogh et al., 2001]: contrary to binary segmentation, bottom-up method starts with many breakpoints and successively removes less important ones.
- Window [Baños et al., 2014]: fixed-size sliding window method with step size to be half the window size.

- KCpE [Harchaoui and Cappé, 2007]: a kernel-based nonparametric segmentation method which segments multi-dimensional data by minimizing intra-segment scatter. Dynamic programming is applied to recursively find breakpoints.
- KCpA [Harchaoui et al., 2008]: a kernel-based test statistic based upon the maximum kernel Fisher discriminant ratio as a measure of homogeneity between segments. Sliding windows are running along the data.
- PELT [Killick et al., 2012]: a pruning DP method with exact optimal solution under certain conditions.
- E-Divisive [Matteson and James, 2014]: a nonparametric technique which combines bisection and divergence measure to form a hierarchical statistical testing.
- e-cp3o and ks-cp3o [Zhang et al., 2017]: dynamic programming with search space pruning. Two popular nonparametric goodness-of-fit metrics are utilized as cost functions, namely E-statistics and the Kolmogorov-Smirnov statistics.
- pDPA [Rigaill, 2010, 2015]: a functional pruning method which can only handle scalar data. Hence in the experiments, we only use the first dimension of the data.

7.3.3.3 Experimental results

The overall comparison results are listed in Table. 7.2. Our proposed method achieves the best segmentation performance on three out of four datasets. All the RI values are quite close, but the classification performance of our proposed method are greater than the best baselines with the margin of 9% and 57% on Skoda and PS dataset respectively. It is interesting to find out that the performance of the proposed method seems to be relevant to the number of $\frac{\#Seg.}{\#C.}$ as listed in Table. 7.1. The larger the average number of segments that each class of activity has, the better the performance of segmentation. This may shed light on the reason of the so much higher classification performance of the proposed method in PS dataset, since the $\frac{\#Seg.}{\#C.}$ value is much greater than that of other datasets. This is reasonable, since the more repetitions of activities in the dataset, the more accurate the class-wise similarity measure in the proposed method.

Methods \ Datasets	Ind	Skoda		WISDM		HCI		PS	
		RI	miF $\pm$ std	RI	miF $\pm$ std	RI	miF $\pm$ std	RI	miF $\pm$ std
S-SMM _{AR}	yes	99.85	55.24$\pm$2.35	98.95	29.88$\pm$3.52	99.39	24.04 $\pm$ 4.39	99.96	86.27$\pm$3.34
Binseg	yes	99.77	33.31 $\pm$ 14.02	98.82	24.10 $\pm$ 5.95	99.28	26.12 $\pm$ 3.87	99.89	28.56 $\pm$ 5.71
BottomUp	yes	99.83	46.43 $\pm$ 6.87	98.80	28.80 $\pm$ 2.86	99.38	21.15 $\pm$ 2.65	99.90	19.84 $\pm$ 4.69
KCpE	yes	99.83	45.35 $\pm$ 22.25	98.81	25.24 $\pm$ 6.78	99.39	24.04 $\pm$ 4.39	99.94	20.37 $\pm$ 6.56
KCPA	no	99.69	32.53 $\pm$ 17.97	98.84	11.80 $\pm$ 5.58	99.54	28.04$\pm$8.65	99.88	25.58 $\pm$ 6.06
PELT	no	99.69	12.13 $\pm$ 9.41	98.85	28.85 $\pm$ 3.63	99.29	26.12 $\pm$ 10.85	99.88	22.01 $\pm$ 3.60
Window	no	99.69	0.77 $\pm$ 1.08	98.79	28.42 $\pm$ 2.02	99.41	13.62 $\pm$ 4.53	99.88	16.15 $\pm$ 4.89
e.divisive	yes	96.99	23.15 $\pm$ 1.25	98.86	17.56 $\pm$ 5.52	95.99	19.23 $\pm$ 0.00	96.25	13.41 $\pm$ 3.61
ks.cp3o	yes	95.26	20.81 $\pm$ 0.76	98.79	13.73 $\pm$ 2.45	65.26	19.23 $\pm$ 0.00	96.25	14.65 $\pm$ 2.02
e.cp3o	yes	96.78	22.77 $\pm$ 2.51	98.75	22.53 $\pm$ 8.54	96.48	20.83 $\pm$ 0.50	96.68	14.38 $\pm$ 2.56
pDPA	no	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

TABLE 7.2: Overall comparison results of segmentation performance on the four datasets (unit: %). NaN indicates that the produced results are infeasible.

7.3.4 Experiments for Joint Segmentation and Feature Extraction

7.3.4.1 Experimental Setup

In this scenario, we investigate the joint segmentation and classification performance of our method. For baseline methods, we apply the segmentation methods mentioned in their papers (for miFV method, sliding window methods are applied), and then conducted the corresponding feature extraction methods. The segmented data is randomly split into training and testing sets with a ratio of 70% : 30%. Both the training and testing data are set to contain activities of all classes. All the results are reported by taking average values together with the standard deviation over 6 repeated experiments. We compare the proposed method with the state-of-the-art baselines. In order to compare segmentation and feature extraction methods, to minimize the impact of classifiers, SVM is chosen as the unique classifier, and we use LIBSVM [Chang and Lin, 2011] for implementation.

7.3.4.2 Baselines

- **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].
- **SAX- a .** Following the settings in [Lin et al., 2007b], 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 $a \in \{3, \dots, 10\}$.

- **miFV- c .** 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 $c \in \{3, 6, 9, 10\}$.

7.3.4.3 Experimental results

Methods	Skoda		WISDM		HCI		PS	
	miF $\pm$ std	maF $\pm$ std	miF $\pm$ std	maF $\pm$ std	miF $\pm$ std	maF $\pm$ std	miF $\pm$ std	maF $\pm$ std
S-SMM _{AR}	51.65$\pm$6.18	42.98$\pm$6.33	28.18$\pm$4.13	27.61$\pm$4.33	23.88$\pm$4.14	15.15$\pm$3.97	86.44$\pm$3.44	85.81$\pm$3.70
ECDF-5	16.29 $\pm$ 7.99	9.48 $\pm$ 4.68	26.16 $\pm$ 3.18	32.08 $\pm$ 3.44	14.42 $\pm$ 3.99	13.44 $\pm$ 3.93	18.11 $\pm$ 5.24	17.63 $\pm$ 5.33
ECDF-15	22.91 $\pm$ 7.86	17.19 $\pm$ 5.41	16.83 $\pm$ 1.86	21.60 $\pm$ 1.42	12.82 $\pm$ 7.06	11.65 $\pm$ 5.31	16.71 $\pm$ 2.99	16.36 $\pm$ 2.85
ECDF-30	23.51 $\pm$ 7.51	19.79 $\pm$ 6.40	10.95 $\pm$ 2.61	13.30 $\pm$ 3.79	11.86 $\pm$ 6.25	9.91 $\pm$ 4.28	16.43 $\pm$ 1.73	16.11 $\pm$ 1.66
ECDF-45	25.96 $\pm$ 5.58	23.36 $\pm$ 5.54	10.43 $\pm$ 3.57	11.18 $\pm$ 4.05	10.74 $\pm$ 6.86	8.67 $\pm$ 3.95	15.87 $\pm$ 1.98	15.48 $\pm$ 2.01
SAX-3	3.92 $\pm$ 3.25	3.48 $\pm$ 2.81	16.06 $\pm$ 2.56	19.46 $\pm$ 3.52	23.88 $\pm$ 7.80	14.56 $\pm$ 9.54	15.13 $\pm$ 2.20	14.58 $\pm$ 2.04
SAX-6	2.11 $\pm$ 1.89	2.06 $\pm$ 1.86	15.28 $\pm$ 2.57	18.39 $\pm$ 3.24	19.39 $\pm$ 3.77	9.48 $\pm$ 3.46	16.95 $\pm$ 1.46	15.87 $\pm$ 1.51
SAX-9	4.15 $\pm$ 3.60	3.99 $\pm$ 3.46	15.98 $\pm$ 2.80	19.24 $\pm$ 3.33	20.83 $\pm$ 1.57	9.90 $\pm$ 2.88	15.48 $\pm$ 2.30	14.91 $\pm$ 2.30
SAX-10	2.69 $\pm$ 2.09	2.65 $\pm$ 2.10	15.27 $\pm$ 3.14	18.73 $\pm$ 3.53	22.44 $\pm$ 5.82	11.67 $\pm$ 6.34	16.43 $\pm$ 1.11	15.49 $\pm$ 1.05
miFV-3	3.08 $\pm$ 5.61	2.04 $\pm$ 3.50	13.43 $\pm$ 0.19	3.18 $\pm$ 0.08	19.23 $\pm$ 0.00	6.20 $\pm$ 0.00	11.95 $\pm$ 0.05	2.55 $\pm$ 0.02
miFV-6	18.22 $\pm$ 7.58	13.70 $\pm$ 4.99	13.43 $\pm$ 0.19	3.18 $\pm$ 0.08	19.23 $\pm$ 0.00	6.20 $\pm$ 0.00	11.95 $\pm$ 0.05	2.55 $\pm$ 0.02
miFV-9	37.38 $\pm$ 4.10	30.55 $\pm$ 3.30	13.43 $\pm$ 0.19	3.19 $\pm$ 0.08	19.23 $\pm$ 0.00	6.20 $\pm$ 0.00	11.95 $\pm$ 0.05	2.55 $\pm$ 0.02
miFV-10	33.57 $\pm$ 4.67	27.30 $\pm$ 4.37	13.43 $\pm$ 0.19	3.18 $\pm$ 0.08	19.23 $\pm$ 0.00	6.20 $\pm$ 0.00	11.95 $\pm$ 0.05	2.55 $\pm$ 0.02

TABLE 7.3: Overall comparison results on joint segmentation and feature extraction on four datasets (unit:%).

As listed in Table 7.3, our proposed S-SMM_{AR} has the best performance on all four datasets. The results clearly demonstrate the efficacy of our proposed unified framework to do segmentation and feature extraction. One potential reason of our proposed method surpassing other baselines may come from two aspects: 1) our segmentation methods are more accurate than the preprocessing step in baselines, which supports our motivation that segmentation is a crucial preprocessing step; 2) our feature extraction has no information loss but the baselines can only extract limited number of features.

7.3.5 Experiments for Classification with Perfect Segmentation

In this scenario, we are given the ground truth segments of the raw data beforehand, and the settings are the same as those in Chapter 4. Therefore, we omit the details here.

7.4 Summary

In this chapter, we propose a novel unified framework, denoted by S-SMM_{AR}, to jointly segment the activity data and 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 investigate the performance of general time-series segmentation methods on the specific activity data. We conduct extensive evaluations and demonstrate the effectiveness of S-SMM_{AR} compared with a number of baseline methods. Moreover, we also present an accelerated version R-SMM_{AR} to solve large-scale problems.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

In this thesis, we focus on the problem of human activity recognition. The research background of activity recognition and the arising research challenges are highlighted in Chapter 1. Existing works are listed in Chapter 2, and Chapter 3 introduces several aspects of preliminaries for our works. In Chapter 4, we introduce our first work on feature learning via learning from distributions in supervised learning setting. In Chapter 5, we propose a novel end-to-end deep neural network structure, which is able to extract statistical features, but also temporal and spatial features. Then in Chapter 6, we further extend the method in semi-supervised learning setting, where only a small fraction of labeled data is required, which greatly reduces the human annotation effort. In Chapter 7, a jointly segmentation and feature learning framework is proposed under weakly-supervised learning setting. Finally, in this chapter, we summarize the contributions of this thesis in the following.

- We propose the SMM_{AR} approach, based on learning from distributions for sensor-based activity recognition. Specifically, we consider sensor readings for each activity as a sample, which can be represented by a feature vector of infinite dimensions in a RKHS using kernel mean embedding techniques. We then train a classifier in the RKHS. To scale-up the proposed method, we further offer

an accelerated version, denoted by R-SMM_{AR} by utilizing an explicit feature map instead of using a kernel function. As far as we know, our work is the first attempt to explore the kernel mean embedding on the task of activity recognition.

- We further propose a Distribution-Embedded Neural Network (DDNN), which is a unified end-to-end trainable deep learning model. DDNN is able to learn three different types of powerful features for activity recognition in an automated fashion.
- To tackle the heavy annotation effort for labeling training data, we propose a novel method, named Distribution-based Semi-Supervised Learning (DSSL). The proposed method is capable of automatically extracting powerful features with no domain knowledge required, meanwhile, alleviating the heavy annotation effort through semi-supervised learning. Specifically, we treat data stream of sensor readings received in a period as a distribution, and map all training distributions, including labeled and unlabeled, into a reproducing kernel Hilbert space (RKHS) using the kernel mean embedding technique. The RKHS is further altered by exploiting the underlying geometry structure of the unlabeled distributions. Finally, in the altered RKHS, a classifier is trained with the labeled distributions.
- We model the weakly-supervised segmentation problem of activity data as a non-convex optimization problem, and propose a novel iterative kernel-based method to solve it. The segmentation method together with a novel feature extraction method are integrated into a unified framework that enables jointly learning of segmentation, feature extraction and classification for sensor-based activity recognition.

8.2 Future Work

As the human activity recognition continues to be an active research field in the near future, here we point out some potential research directions beyond the present works introduced in this thesis.

- Our proposed methods work under batch learning or offline learning fashion where a collection of training data are readily available at training stage. Such learning methods suffer from time-consuming re-training step when more training data becomes available. It is crucial to deal with increasing and evolving new data in the era of big data. To this end, online learning paradigm [Hoi et al., 2018, Lu et al., 2016] becomes important, since online learning algorithms is able to learn a model from a sequence of data instances one at a time, and to painlessly evolve upon new incoming data.
- Currently, the data is usually gathered from wearable sensors, and then transformed to servers before data preprocessing. Then a classification model is trained on CPU or GPU servers before being applicable to new unseen data. It is very promising to implement the entire activity recognition system on a single mobile phone or edge device, which enables the entire data collection, data preprocessing, and training of a classification model procedures to be conducted in the same device. Due to the constraints of memory and battery of an edge device, the capacity of a trained model in an edge device or a mobile phone is much smaller than the models trained on servers. There are attempts to distill knowledge from larger trained models to a smaller model with the technique of knowledge distillation [Hinton et al., 2015, Wang et al., 2018].
- Another promising research direction is to consider the differences and variances caused by participants' differences. It is natural that every person has unique style of conducting activities, and it is very beneficial to take into account the uniqueness and make the activity recognition system to be personalized. Transfer learning [Pan and Yang, 2010, Pan et al., 2009] can solve the problem by treating data from each person as data from different domains.

Bibliography

- Yasemin Altun and Alexander J. Smola. Unifying divergence minimization and statistical inference via convex duality. In *COLT*, volume 4005 of *Lecture Notes in Computer Science*, pages 139–153. Springer, 2006.
- Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. In *NIPS*, pages 561–568, 2002.
- Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *IWAAL*, volume 7657 of *Lecture Notes in Computer Science*, pages 216–223. Springer, 2012.
- Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950.
- Akin Avci, Stephan Bosch, Mihai Marin-Perianu, Raluca Marin-Perianu, and Paul J. M. Havinga. Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey. In *ARCS Workshops*, pages 167–176, 2010.
- Francis R. Bach and Michael I. Jordan. Predictive low-rank decomposition for kernel methods. In *ICML*, pages 33–40, 2005.
- Marc Bächlin, Meir Plotnik, Daniel Roggen, Inbal Maidan, Jeffrey M. Hausdorff, Nir Giladi, and Gerhard Tröster. Wearable assistant for parkinson’s disease patients with the freezing of gait symptom. *IEEE Trans. Information Technology in Biomedicine*, 14(2):436–446, 2010.

- Oresti Baños, Juan Manuel Galvez, Miguel Damas, Héctor Pomares, and Ignacio Rojas. Window size impact in human activity recognition. *Sensors*, 14(4):6474–6499, 2014.
- Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.
- Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006. URL <http://www.jmlr.org/papers/v7/belkin06a.html>.
- Alain Berlinet and Christine Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.
- Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.
- S. Bochner. Monotone funktionen, stieltjessche integrale und harmonische analyse. *Mathematische Annalen*, 108(1):378–410, Dec 1933. ISSN 1432-1807. doi: 10.1007/BF01452844. URL <https://doi.org/10.1007/BF01452844>.
- 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.
- Razvan C. Bunescu and Raymond J. Mooney. Multiple instance learning for sparse positive bags. In *ICML*, pages 105–112, 2007.
- 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.
- Olivier Chapelle and Alexander Zien. Semi-supervised classification by low density separation. In *AISTATS*, 2005.
- Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. *Semi-Supervised Learning*. The MIT Press, 1st edition, 2010. ISBN 0262514125, 9780262514125.
- Ricardo Chavarriaga, Hesam Sagha, Alberto Calatroni, Sundara Tejaswi Digumarti, Gerhard Tröster, José del R. Millán, and Daniel Roggen. The opportunity challenge: Thesis of Hangwei Qian@NTU

- A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters*, 34(15):2033–2042, 2013.
- Jie Chen and Arjun K Gupta. *Parametric statistical change point analysis: with applications to genetics, medicine, and finance*. Springer Science & Business Media, 2011.
- Diane J. Cook, Kyle D. Feuz, and Narayanan Chatapuram Krishnan. Transfer learning for activity recognition: a survey. *Knowl. Inf. Syst.*, 36(3):537–556, 2013.
- Efren Cruz Cortes and Clayton Scott. Scalable sparse approximation of a sample mean. In *ICASSP*, pages 5237–5241. IEEE, 2014.
- F Foerster, M Smeja, and J Fahrenberg. Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring. *Computers in Human Behavior*, 15(5):571–583, 1999.
- Kilian Förster, Daniel Roggen, and Gerhard Tröster. Unsupervised classifier self-calibration through repeated context occurrences: 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*, 2010.
- Piotr Fryzlewicz et al. Wild binary segmentation for multiple change-point detection. *The Annals of Statistics*, 42(6):2243–2281, 2014.
- Erich Fuchs, Thiemo Gruber, Jiri Nitschke, and Bernhard Sick. Online segmentation of time series based on polynomial least-squares approximations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(12):2232–2245, 2010.
- Wei Gao, Sam Emaminejad, Hnin Yin Yin Nyein, Samyuktha Challa, Kevin Chen, Austin Peck, Hossain M Fahad, Hiroki Ota, Hiroshi Shiraki, Daisuke Kiriya, et al. Fully integrated wearable sensor arrays for multiplexed in situ perspiration analysis. *Nature*, 529(7587):509, 2016.
- Thomas Gärtner, Peter A. Flach, Adam Kowalczyk, and Alexander J. Smola. Multi-instance kernels. In *ICML*, pages 179–186, 2002.

- Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander J. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.
- Yann Guédon. Exploring the latent segmentation space for the assessment of multiple change-point models. *Computational Statistics*, 28(6):2641–2678, Dec 2013. ISSN 1613-9658. doi: 10.1007/s00180-013-0422-9. URL <https://doi.org/10.1007/s00180-013-0422-9>.
- 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.
- Nils Y. Hammerla, Shane Halloran, and Thomas Plötz. Deep, convolutional, and recurrent models for human activity recognition using wearables. In *IJCAI*, pages 1533–1540. IJCAI/AAAI Press, 2016.
- Zaid Harchaoui and Olivier Cappé. Retrospective mutiple change-point estimation with kernels. In *Workshop on Statistical Signal Processing*, pages 768–772, 2007.
- Zaïd Harchaoui, Francis R. Bach, and Eric Moulines. Kernel change-point analysis. In *NIPS*, pages 609–616, 2008.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *arXiv e-prints*, art. arXiv:1503.02531, Mar 2015.
- Toby Hocking, Guillem Rigai, and Guillaume Bourque. Peakseg: constrained optimal segmentation and supervised penalty learning for peak detection in count data. In *ICML*, volume 37, pages 324–332, 2015.
- Toby Dylan Hocking, Guillem Rigai, Paul Fearnhead, and Guillaume Bourque. A log-linear time algorithm for constrained changepoint detection. *arXiv preprint arXiv:1703.03352*, 2017.
- Steven C. H. Hoi, Jiale Wang, and Peilin Zhao. LIBOL: a library for online learning algorithms. *J. Mach. Learn. Res.*, 15(1):495–499, 2014.

- Steven C. H. Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao. Online learning: A comprehensive survey. *CoRR*, abs/1802.02871, 2018.
- Andrey Ignatov. Real-time human activity recognition from accelerometer data using convolutional neural networks. *Appl. Soft Comput.*, 62:915–922, 2018.
- 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.
- Roger T Johnson and David W Johnson. Active learning: Cooperation in the classroom. *The annual report of educational psychology in Japan*, 47:29–30, 2008.
- Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. An online algorithm for segmenting time series. In *ICDM*, pages 289–296, 2001.
- Rebecca Killick, Paul Fearnhead, and Idris A Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, 2012.
- 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.
- Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.
- Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. MMD GAN: towards deeper understanding of moment matching network. In *NIPS*, pages 2200–2210, 2017.
- Yujia Li, Kevin Swersky, and Richard S. Zemel. Generative moment matching networks. In *ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1718–1727. JMLR.org, 2015.

- Hsuan-Tien Lin, Chih-Jen Lin, and Ruby C. Weng. A note on platt's probabilistic outputs for support vector machines. *Machine Learning*, 68(3):267–276, 2007a.
- 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, 2007b.
- Jeffrey W. Lockhart and Gary M. Weiss. Limitations with activity recognition methodology & data sets. In *UbiComp*, pages 747–756, 2014.
- Jing Lu, Steven C. H. Hoi, Jialei Wang, Peilin Zhao, and Zhiyong Liu. Large scale online kernel learning. *J. Mach. Learn. Res.*, 17:47:1–47:43, 2016.
- Robert Maidstone, Toby Hocking, Guillem Rigai, and Paul Fearnhead. On optimal multiple changepoint algorithms for large data. *Statistics and Computing*, 27(2):519–533, 2017.
- 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.
- Ryunosuke Matsushige, Koh Kakusho, and Takeshi Okadome. Semi-supervised learning based activity recognition from sensor data. In *GCCE*, pages 106–107. IEEE, 2015.
- David S Matteson and Nicholas A James. A nonparametric approach for multiple change point analysis of multivariate data. *Journal of the American Statistical Association*, 109(505):334–345, 2014.
- 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.
- James Mercer and Andrew Russell Forsyth. Xvi. functions of positive and negative type, and their connection the theory of integral equations. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 209(441-458):415–446, 1909. doi: 10.1098/rsta.

- 1909.0016. URL <https://royalsocietypublishing.org/doi/abs/10.1098/rsta.1909.0016>.
- Donald Michie, David J Spiegelhalter, CC Taylor, et al. Machine learning. *Neural and Statistical Classification*, 13, 1994.
- Francisco Javier Ordóñez Morales and Daniel Roggen. Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1):115, 2016. doi: 10.3390/s16010115. URL <https://doi.org/10.3390/s16010115>.
- K. Muandet. *From Points to Probability Measures: A Statistical Learning on Distributions with Kernel Mean Embedding*. PhD thesis, University of Tübingen, Germany, September 2015.
- Krikamol Muandet, Kenji Fukumizu, Francesco Dinuzzo, and Bernhard Schölkopf. Learning from distributions via support measure machines. In *NIPS*, pages 10–18, 2012.
- Krikamol Muandet, Kenji Fukumizu, Bharath K. Sriperumbudur, Arthur Gretton, and Bernhard Schölkopf. Kernel mean estimation and stein effect. In *ICML*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 10–18. JMLR.org, 2014.
- Krikamol Muandet, Kenji Fukumizu, Bharath K. Sriperumbudur, and Bernhard Schölkopf. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends in Machine Learning*, 10(1-2):1–141, 2017.
- Alfredo Nazábal, Pablo Garcia-Moreno, Antonio Artés-Rodríguez, and Zoubin Ghahramani. Human activity recognition by combining a small number of classifiers. *IEEE J. Biomedical and Health Informatics*, 20(5):1342–1351, 2016.
- Qin Ni, Timothy Patterson, Ian Cleland, and Chris D. Nugent. Dynamic detection of window starting positions and its implementation within an activity recognition framework. *Journal of Biomedical Informatics*, 62:171–180, 2016.
- Feiping Nie, Heng Huang, Xiao Cai, and Chris H. Q. Ding. Efficient and robust feature selection via joint ℓ_2, ℓ_1 -norms minimization. In *NIPS*, pages 1813–1821, 2010.

- Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, 2010. doi: 10.1109/TKDE.2009.191. URL <https://doi.org/10.1109/TKDE.2009.191>.
- Sinno Jialin Pan, James T. Kwok, Qiang Yang, and Jeffrey Junfeng Pan. Adaptive localization in a dynamic wifi environment through multi-view learning. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, pages 1108–1113, 2007. URL <http://www.aaai.org/Library/AAAI/2007/aaai07-176.php>.
- Sinno Jialin Pan, Ivor W. Tsang, James T. Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. In *IJCAI*, pages 1187–1192, 2009.
- Shyamal Patel, Hyung Park, Paolo Bonato, Leighton Chan, and Mary Rodgers. A review of wearable sensors and systems with application in rehabilitation. *Journal of neuroengineering and rehabilitation*, 9(1):21, 2012.
- Thomas Plötz, Nils Y. Hammerla, and Patrick Olivier. Feature learning for activity recognition in ubiquitous computing. In *IJCAI*, pages 1729–1734, 2011.
- Ronald Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010.
- Jun Qi, Po Yang, Atif Waraich, Zhikun Deng, Youbing Zhao, and Yun Yang. Examining sensor-based physical activity recognition and monitoring for healthcare using internet of things: A systematic review. *Journal of biomedical informatics*, 2018.
- Hangwei Qian, Sinno Jialin Pan, and Chunyan Miao. Sensor-based activity recognition via learning from distributions. In *AAAI*. AAAI Press, 2018.
- Hangwei Qian, Sinno Jialin Pan, Bingshui Da, and Chunyan Miao. A novel distribution-embedded neural network for sensor-based activity recognition. In *IJCAI*, 2019a.
- Hangwei Qian, Sinno Jialin Pan, and Chunyan Miao. Distribution-based semi-supervised learning for activity recognition. In *AAAI*. AAAI Press, 2019b.
- Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *NIPS*, pages 1177–1184, 2007.

- Rouhollah Rahmani and Sally A. Goldman. MISSL: multiple-instance semi-supervised learning. In *ICML*, pages 705–712, 2006.
- Sreenivasan Ramasamy Ramamurthy and Nirmalya Roy. Recent trends in machine learning for human activity recognition - A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, 8(4), 2018.
- Daniele Ravì, Charence Wong, Benny Lo, and Guang-Zhong Yang. A deep learning approach to on-node sensor data analytics for mobile or wearable devices. *IEEE J. Biomedical and Health Informatics*, 21(1):56–64, 2017.
- Attila Reiss and Didier Stricker. Introducing a new benchmarked dataset for activity monitoring. In *ISWC*, pages 108–109. IEEE Computer Society, 2012.
- Guillem Rigauill. Pruned dynamic programming for optimal multiple change-point detection. *arXiv preprint arXiv:1004.0887*, 2010.
- Guillem Rigauill. A pruned dynamic programming algorithm to recover the best segmentations with 1 to $k_{\max}$ change-points. *Journal de la Société Française de Statistique*, 156(4):180–205, 2015.
- 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.
- Ahmad Shahi, Brendon J. Woodford, and Hanhe Lin. Dynamic real-time segmentation and recognition of activities using a multi-feature windowing approach. In *PAKDD (Workshops)*, pages 26–38, 2017.
- Muhammad Shoaib, Hans Scholten, and Paul J. M. Havinga. Towards physical activity recognition using smartphone sensors. In *UIC/ATC*, pages 80–87, 2013.
- Muhammad Shoaib, Stephan Bosch, Özlem Durmaz Incel, Hans Scholten, and Paul J. M. Havinga. A survey of online activity recognition using mobile phones. *Sensors*, 15(1):2059–2085, 2015.

- Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.
- Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin. Beyond the point cloud: from transductive to semi-supervised learning. In *ICML*, pages 824–831, 2005.
- 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.
- Thomas Stiefmeier, Daniel Roggen, and Gerhard Tröster. Fusion of string-matched templates for continuous activity recognition. In *ISWC*, pages 41–44, 2007.
- Maja Stikic, Diane Larlus, and Bernt Schiele. Multi-graph based semi-supervised learning for activity recognition. In *ISWC*, pages 85–92. IEEE Computer Society, 2009.
- Maja Stikic, Diane Larlus, Sandra Ebert, and Bernt Schiele. Weakly supervised recognition of daily life activities with wearable sensors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(12):2521–2537, 2011. doi: 10.1109/TPAMI.2011.36. URL <https://doi.org/10.1109/TPAMI.2011.36>.
- Charles Truong, Laurent Oudre, and Nicolas Vayatis. A review of change point detection methods, 2018.
- Vladimir Vapnik. *Statistical learning theory*. Wiley, 1998.
- Ramachandran Varatharajan, Gunasekaran Manogaran, Malarvizhi Kumar Priyan, and Revathi Sundarasekar. Wearable sensor devices for early detection of alzheimer disease using dynamic time warping algorithm. *Cluster Computing*, 21(1):681–690, 2018.
- Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. Deep learning for sensor-based activity recognition: A survey. *CoRR*, abs/1707.03502, 2017.

- Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. KDGAN: knowledge distillation with generative adversarial networks. In *NeurIPS*, pages 783–794, 2018.
- Yan Wang, Shuang Cang, and Hongnian Yu. A survey on wearable sensor modality centred human activity recognition in health care. *Expert Systems with Applications*, 2019.
- 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.
- Jianbo Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiaoli Li, and Shonali Krishnaswamy. Deep convolutional neural networks on multichannel time series for human activity recognition. In *IJCAI*, pages 3995–4001. AAAI Press, 2015.
- Qiang Yang, Sinno Jialin Pan, and Vincent Wenchen Zheng. Estimating location using wi-fi. *IEEE Intelligent Systems*, 23(1):8–13, 2008. doi: 10.1109/MIS.2008.4. URL <https://doi.org/10.1109/MIS.2008.4>.
- Yun Yang and Azin Sahabi. Modular wearable sensor device, March 8 2016. US Patent 9,277,864.
- Lina Yao, Feiping Nie, Quan Z. Sheng, Tao Gu, Xue Li, and Sen Wang. Learning from less for better: semi-supervised activity recognition via shared structure discovery. In *UbiComp*, pages 13–24. ACM, 2016.
- Jie Yin, Dou Shen, Qiang Yang, and Ze-Nian Li. Activity recognition through goal-based segmentation. In *AAAI*, pages 28–34, 2005.
- Ming Zeng, Le T. Nguyen, Bo Yu, Ole J. Mengshoel, Jiang Zhu, Pang Wu, and Joy Zhang. Convolutional neural networks for human activity recognition using mobile sensors. In *MobiCASE*, pages 197–205. IEEE, 2014.
- Wenyu Zhang, Nicholas A. James, and David S. Matteson. Pruning and nonparametric multiple change point detection. In *ICDM Workshops*, pages 288–295, 2017.

- Yu Zhou and Anlong Ming. Semi-supervised multiple instance learning and its application in visual tracking. In *WCSP*, pages 1–5, 2016.
- Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 2017.
- Zhi-Hua Zhou and Jun-Ming Xu. On the relation between multi-instance learning and semi-supervised learning. In *ICML*, volume 227, pages 1167–1174, 2007.
- Zhi-Hua Zhou, Yu-Yin Sun, and Yu-Feng Li. Multi-instance learning by treating instances as non-iid samples. In *Proceedings of the 26th annual international conference on machine learning*, pages 1249–1256. ACM, 2009.
- Jun Zhu, Junhua Mao, and Alan L. Yuille. Learning from weakly supervised data by the expectation loss SVM (e-svm) algorithm. In *NIPS*, pages 1125–1133, 2014.
- Xiaojin Zhu. Semi-supervised learning literature survey. Technical report, Computer Sciences, University of Wisconsin-Madison, 2005.
- Zhihua Zhu, Tao Liu, Guangyi Li, Tong Li, and Yoshio Inoue. Wearable sensor systems for infants. *Sensors*, 15(2):3721–3749, 2015.