Learning multi-level features for sensor-based human action recognition

Yan Xu [a,b], Zhengyang Shen [a], Xin Zhang [a], Yifan Gao [c], Shujian Deng [a], Yipei Wang [a], Yubo Fan [a,*], Eric I-Chao Chang [b,*]

[a] State Key Laboratory of Software Development Environment and Key Laboratory of Biomechanics and Mechanobiology of Ministry of Education and Research Institute of Beihang University in Shenzhen, Beihang University, Beijing 100191, China
[b] Microsoft Research Asia, Beijing 100080, China
[c] School of Electronic and Information Engineering, Beihang University, Beijing 100191, China

HIGHLIGHTS

- A multi-level feature learning framework is proposed on sensor-based human action recognition.
- A novel latent pattern learning method learns semantic representation.
- High-level feature possesses higher representation ability than low- and mid-level features.

ARTICLE INFO

ABSTRACT

This paper proposes a multi-level feature learning framework for human action recognition using a single body-worn inertial sensor. The framework consists of three phases, respectively designed to analyze signal-based (low-level), components (mid-level) and semantic (high-level) information. Low-level features capture the time and frequency domain property while mid-level representations learn the composition of the action. The Max-margin Latent Pattern Learning (MLPL) method is proposed to learn high-level semantic descriptions of latent action patterns as the output of our framework. The proposed method achieves the state-of-the-art performances, 88.7%, 98.8% and 72.6% (weighted F_1 score) respectively, on Skoda, WISDM and OPP datasets.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

In recent years, sensor-based human action recognition (HAR) plays a key role in the area of ubiquitous computing due to its wide application in daily life [1–6]. In most cases, utilization of raw data without special process is impractical since raw recordings confound noise and meaningless components. Therefore, the process of feature extraction and feature learning is supposed to be conducted [7], which is commonly based on the sliding window technique through sampling overlapping frames from signal streams [8]. In general, studies on sensor-based HAR have mainly focused on low-level [9–12] and mid-level [13–15] features.

Low-level features include statistical features [9,10], frequency-domain features [11,12] and some hand-crafted methods [16]. Low-level features are popular owing to their simplicity as well as their acceptable performances across a variety

of action recognition problems. However, their simplicity is often accompanied by less discrimination in representation, considering that actions are often highly complex and diverse in daily-life cases.

Compared with low-level features, mid-level features which are mainly obtained through dictionary learning methods [14,15] have proven to be more robust and discriminative [13]. The representations include the sparse coding method [17] and the bag-of-words (BOW) algorithm [15]. These methods analyze components (motion primitives) and explore inherent structures of signals. However, when the dictionary size is relatively large, mid-level features would suffer high computation, redundancy in representation and further burden the following classification. In this paper, we reduce the redundancy of the mid-level representation by introducing high-level features, which can achieve better overall performances and be robust to dictionary size.

Many studies have applied probabilistic graphical models [18,19] and pattern-mining methods [20–24] in semantic (high-level) recognition tasks. However, few works focus on feature learning methods in sensor-based HAR. In this paper, a semantic feature learning algorithm is proposed on two motivations. First, it can remove the redundancy in mid-level representation. A compact and discriminative description can be achieved by applying a latent pattern learning method on mid-level features. Second, implementing semantic analysis on sensor signals is an efficient and intuitive way to discover and deal with the variety of action patterns, and thus it can remove ambiguities in descriptions of the action. In particular, the ambiguities mainly come from two aspects, physical level and annotation level, both of which make it difficult to generate generalized good features in HAR tasks [8]. At the physical level, one challenge is that different or even the same people who conduct the same action may produce completely different signals due to changes in environment, which is called the intra-class variability [25]. For example, walking style in morning after a good sleep commonly differs from the one after day's hard work. The other challenge is the inter-class similarity [26], which refers to the similarity among different actions such as 'drinking milk' and 'drinking coffee'. As the inter-class similarity is generally resolved through obtaining additional data from different sensors [27] or analyzing co-occurring actions [28], our work focuses on dealing with the intra-class variability. At the annotation level, a specific action performed in many ways can be cognitively identified as the same one [29]. Take 'cleaning the table' as an example. It is rational to consider cleaning table from right to left and from up to down as the same action, though they behave absolutely differently in signal records. For those reasons, generalized features that are directly learned from action perspectives would compromise to the common characteristic shared by different action patterns, resulting in ambiguous feature representations [30]. Inspired by Multiple-Instance Learning (MIL) [31,32], our solution is to mine discriminative latent patterns from actions and construct features based on descriptions of those patterns, which can eliminate ambiguities from both physical and annotation levels. We name this method the Max-margin Latent Pattern Learning (MLPL) method. Instead of being constrained by generic property, the MIL method is a natural solution given that the diversity inside the class can be learnt. Although MIL methods are widely performed in computer vision [33,34], as for sensor-based HAR problems, relevant works are mainly designed to cope with sparse annotation [35,36]. Instead of dealing with sparsely labeled cases, MLPL proposed in this paper implements MIL to learn discriminative latent patterns of actions, by which high-level features would be acquired.

In this paper, we integrate the advantages of low-, mid- and high-level features and propose the framework known as Multi-Level Complementary Feature Learning (MLCFL). To avoid being confused with the high-level feature learned by the latent pattern learning process, the output feature of our framework is denoted as Multi-Level Complementary Feature (MLCF). In particular, this framework learns multi-level features through three phases, which are respectively designed to analyze signal-based (low-level), components (mid-level) and semantic (high-level) information. In the first phase, the low-level feature (statistical values, FFT coefficients, etc.) is extracted from raw signals. In the second phase, from the component perspective, the mid-level representation can be attained through hard coding processes and occurrence statistics. In the third phase, the MLPL method, from the semantic perspective, is implemented on the Compl feature (the concatenation of low- and mid-level features) to obtain MLCF as the output of this framework. Various experiments on Opp [37,38], Skoda [39] and WISDM [40] datasets show that MLCF possesses higher feature representation ability than low- and mid-level features. Moreover, compared with existing methods, the method we proposed achieves state-of-the-art performances. Our contributions in this paper are as follows:

1. A multi-level feature learning framework MLCFL is constructed, which consists of three phases including low-level feature extraction, mid-level components learning and high-level semantic understanding. The output feature is learned level by level, possessing higher representation ability than low- and mid-level features.
2. An efficient and effective latent pattern learning method MLPL is proposed to learn high-level features, each dimension of which refers to the confidence score of corresponding latent action pattern.
3. Our framework is evaluated on three popular datasets and achieves state-of-the-art performances.

The rest of this paper is organized as follows: Section 2 presents related work; Section 3 describes the MLCFL framework for action recognition; Section 4 presents and analyzes experimental results; finally, we conclude the study in Section 5.

2. Related work

Researches in the area of sensor-based HAR have been ever increasing in the past few years [8,16,18,41]. Numerous methods have been proposed in designing, implementing and evaluating action recognition systems. In this section, we

review typically practical methods in terms of feature representation, specifically from low- [9–12,16,42] as well as mid- [13,14,43] perspectives and demonstrate how each method is applied to the specific action recognition task. As few works focus on designing high-level features, an additional review is presented on prevalent approaches involving semantic understanding of actions [18,19,21–24,28,41,44]. Besides, a brief overview about other representative methods [35,45] is summarized.

Low-level features are designed to capture signal-based information. Statistical metrics are the most common approaches, which include mean, variance, standard deviation, energy, entropy and correlation coefficients [9,10]. Fourier Transform (FT), Wavelet Transform (WT) [46], Discrete Cosine Transform (DCT) [47] as well as auto-regressive (AR) coefficients [48] are also commonly applied in HAR tasks for their promising performances. Kang et al. [12] analyzed electromyography (EMG) signals by extracting conventional auto-regressive coefficients and cepstral coefficients as features. Hammerla et al. [16] designed the hand-crafted feature based on the Empirical Cumulative Distribution Function (ECDF) to preserve characteristics of inertial signal distribution. Plötz et al. [42] improved on that work and proposed the ECDF-PCA feature. They implemented the Principal Component Analysis (PCA) method on signals normalized by ECDF and significantly improved performance. In this paper, statistical values, FFT coefficients and ECDF-PCA are calculated as low-level features to demonstrate the generalization ability of the proposed framework.
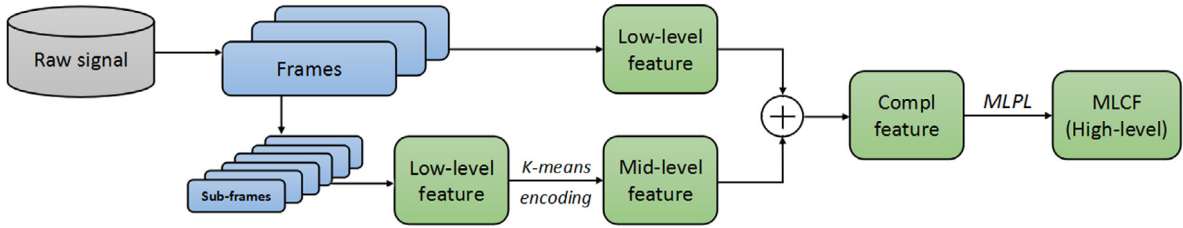
Mid-level features are generally extracted from the low-level ones to explore the components and structural information of signals. They are prevalent in HAR tasks for robustness against noise and discrimination in representations [13,14]. Huynh et al. [15] and Zhang et al. [43] implemented the bag-of-words (BOW) model to obtain statistical descriptions of motion primitives. Their works showed the effectiveness of the BOW model in sensor-based action recognition tasks. Blanke et al. [49] extracted the occurrence statistics feature from low-level actions in a way that is similar to Huynh et al. [15] and then implemented the JointBoosting-framework. One characteristic of their method was to adopt a top-down perspective, using a feature selection algorithm to learn the distinctive motion primitives from the labeled high-level action. Sourav et al. [13] and Christian et al. [14] both utilized sparse coding and adopted the convolution basis, which could resist shifts in time and thus reduce redundancy of basis. Our work is similar to [43], in which the mid-level representation is achieved through hard coding and occurrence statistics.

High-level recognition tasks mainly focus on obtaining intuitive and semantic descriptions of actions. Pattern-mining methods [20–24,28] and probabilistic graphical models [18,19] are the most prevalent approaches. Pattern-mining methods explore the diversity in human actions through learning discriminative action patterns and motion primitives. Huynh et al. [28] applied probabilistic topic models which stemmed from the text processing community to automatically extract action patterns from sensor data. They described the recognition of daily routines as a probabilistic combination of learned patterns. Liu et al. [20] presented an algorithm capable of identifying temporal patterns from low-level actions and utilized these temporal patterns to further represent high-level human actions. Various methods have also been proposed based on probabilistic graphical models [18,19] to capture the temporal, sequential, interleaved or concurrent relationship among motion primitives. However, graphical models are limited to capturing rich temporal relationships in complex actions and also suffer from the exponential increase in calculation when the number of involved actions grows [20]. Instead of modeling temporal relationships, we propose an efficient pattern-mining approach, which takes advantage of being compact in representation, intuitive in understanding and efficient in calculation.

Multiple-Instance Learning (MIL) methods in HAR have been widely applied to cope with scarcity of annotation. Maja et al. [35] proposed a framework involving MIL as a weakly supervised recognition method to deal with scarcity of labels and proved its robustness to erroneous labels. Similarly, MIL methods with several novel extensions were introduced to handle different annotation strategies in action recognition [36]. Instead of dealing with annotation scarcity, MLPL proposed in this paper implements MIL to explore latent patterns in human actions, by which the high-level feature would be acquired.

Other representative works in HAR include template matching methods, heuristic methods and deep learning ones. Template matching methods often derived from DTW and LCSS algorithm. Hartmann et al. [50] proposed Segmented DTW to recognize and bound the gesture in each frame through finding the best match among the object and all templates of different classes. Nonsegmented DTW proposed by Stiefmeier et al. [51] was a more efficient variation through reusing the previous computation. Nguyen-Dinh et al. [52,53] improved LCSS algorithm and proposed WarpingLCSS and SegmentedLCSS, which were more efficient than DTW-based methods and robust to noisy annotation. Besides, heuristic methods are often related to specific tasks [11] and depend on domain knowledge. Reyes-Ortiz et al. [54] designed temporal filters to recognize actions as well as postural transitions. In addition, various studies on deep learning methods have been conducted recently, mainly derived from Convolutional Neural Networks (CNN) [55–57] and Recurrent Neural Network (RNN) [58]. These methods explored relationships between the temporal and spatial dependency of recordings and sensors. Zeng et al. [45] applied CNN with partial weight sharing method. The framework they proposed achieved outstanding performance on three popular datasets.

In this paper, contrary to traditional feature extraction and learning methods restricted in low- and mid-level descriptions of signals, we achieve semantic understanding of sensor-based human actions through learning latent action patterns. To the best of our knowledge, our method proposed is the first attempt to explicitly apply a feature learning method to high-level representations in sensor-based HAR tasks. Furthermore, we present a brand new framework to synthesize multi-level features, integrating signal-based (low-level), components (mid-level) and semantic (high-level) information together.

Fig. 1. The flowchart of Multi-Level Complementary Feature Learning (MLCFL). In the first phase, low-level features are extracted from frames and sub-frames. In the second phase, mid-level representations can be obtained through hard coding processes and occurrence statistics. In the third phase, MLPL would be implemented on the Compl feature (the concatenation of low- and mid-level features), where MLCF can be obtained as the output of the framework.

3. Method

Our framework consists of three phases in interpreting the signal: (i) low-level feature extraction; (ii) mid-level components learning; (iii) high-level semantic understanding. The low-level description analyzes the temporal and frequency property of signals while the mid-level representation is a statistical description of either shared or distinctive components (motion primitives). The high-level feature describes the action by distinguishing the specific action pattern it belongs to. The flowchart of the framework is illustrated in Fig. 1.

3.1. Low-level feature extraction

In this phase, low-level features are extracted from raw signals to learn properties in the time and frequency domain. Three popular features, namely statistical values, FFT coefficients and ECDF-PCA, are involved in this work and a brief description is presented as follows.

3.1.1. Statistical values
Statistical metrics analysis is one of the most common approaches to obtaining feature representations of raw signals. Statistical values in this work refer to time-domain features [9,10], including mean, standard deviation, energy, entropy and correlation coefficients.

3.1.2. FFT coefficients
Frequency domain techniques [11,12] have been extensively applied to capture the repetitive nature of sensor signals. This repetition often correlates to the periodicity property of a specific action such as walking or running. The transformation technique used in this paper is Fourier Transform (FT) through which dominant frequency components in the frequency domain can be procured.

3.1.3. ECDF-PCA
ECDF-PCA [42] was brought up on the expertise analysis of inertial signals recorded by the triaxial accelerator sensor. It applies the Empirical Cumulative Distribution Function (ECDF) on signals. Then ECDF feature [16] is procured by inverse equal probability interpolation, through which data are normalized and preserve its inherent structure at the same time. A PCA method is then implemented on this normalized data.

3.2. Mid-level components learning

The mid-level learning method is a general approach in pattern recognition. Dictionary learning methods such as bag-of-words (BOW) [43] and sparse coding [13] are the most popular approaches for obtaining mid-level representations. Compared with low-level feature extraction which involves analyzing properties in the time and frequency domain, mid-level learning focuses on the structural composition. In this paper, we implement BOW in signal processing. The obtained mid-level feature is the statistical description of components of the signal.

The dictionary is first formed from training data through the K-means algorithm [15]. In particular, frames are broken into overlapping sub-frames which are smaller in length. Low-level features are extracted from sub-frames and then the K-means method is used to construct the motion-primitive dictionary. K clusters are generated and the dictionary is formed by cluster centers. We define a set of samples as $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, $\mathbf{x}_i \in \mathbf{R}^{d \times 1}$, $i \in \{1, \ldots, n\}$ and each sample $\mathbf{x}_i$ is associated with an index $z_i \in \{1, \ldots, K\}$. If $z_i = j \in \{1, \ldots, K\}$, $\mathbf{x}_i$ is in the jth cluster. The center of the jth cluster is denoted as $\mathbf{m}_j = \sum_{i=1}^{n} 1\{z_i = j\}\mathbf{x}_i / \sum_{i=1}^{n} 1\{z_i = j\}$, $\mathbf{m}_j \in \mathbf{R}^{d \times 1}$. $\mathbf{m}_j$ refers to a word while j refers to the corresponding index in the dictionary. When a new sample $\mathbf{x}_i$ comes in, the corresponding index z_i can be determined by $z_i = \arg\min_{j \in \{1,\ldots,K\}} (\mathbf{x}_i - \mathbf{m}_j)(\mathbf{x}_i - \mathbf{m}_j)^\mathsf{T}$.

With the learned dictionary, a new symbolic sequence shown in Fig. 2 can be derived from raw signals by densely extracting low-level features from sub-frames and retrieving them from the dictionary. In the end, the mid-level feature is represented by occurrence statistics of motion primitives in the frame.
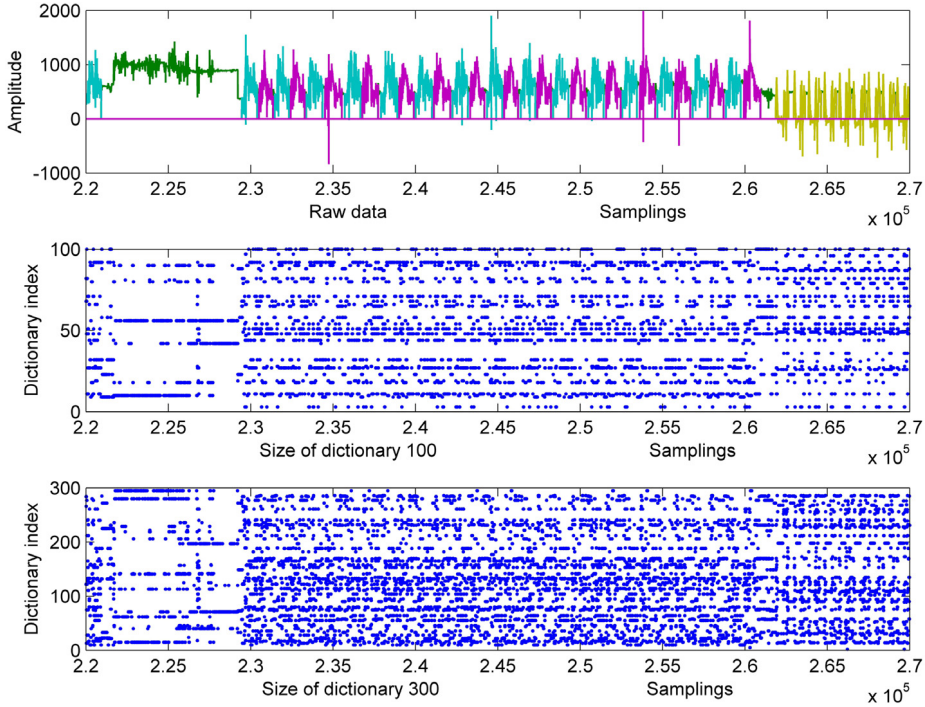
Fig. 2. Illustration of symbolic sequences generated by different dictionary sizes in mid-level learning. From top to bottom: raw data, cluster assignments, of which the dictionary size is 100 and 300 respectively.

3.3. High-level semantic understanding

Generally speaking, one specific action would be highly identifiable if each pattern of the action had been learnt. Similarly, descriptions capturing properties of each distinctive distribution in the feature space would be more discriminative than the ones that only capture the generic but ambiguous characters of the whole distribution. Inspired by Wang's method [32] of learning a weakly-supervised dictionary for discriminative subjects in images, we propose Max-margin Latent Pattern Learning (MLPL) in sensor-based signal processing. In general, the objective of this algorithm is to identify each specific class by learning a set of latent classifiers, each of which can be a discriminative description of a certain action pattern. The high-level feature is represented by the combination of confidence scores belonging to each latent class. Specifically, the latent pattern learning problem in this work, as shown in Fig. 3, can be divided into two aspects: (i) maximizing the inter-class difference, namely the differences between one specific action and other actions; (ii) maximizing the intra-class difference, namely the differences among different patterns of one specific action.

We first present a brief notation of Multiple-Instance Learning (MIL). In MIL, a set of bags are defined as $\mathbf{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_m\}$, and each bag contains a set of instances $\boldsymbol{X}_i = \{\mathbf{x}_{i1}, \ldots, \mathbf{x}_{in}\}$, where $\mathbf{x}_{ij} \in \mathbf{R}^{d \times 1}$. Only one label can be assigned to a bag and instances inside it. The bag would be labeled as positive if there exists at least one positive instance, while being labeled negative only when all of the instances in it are negative. In computer vision, the bag model is a natural description of the image because the image commonly consists of a set of subjects and the label of the image can only be determined by subjects of interest. Compared with the traditional MIL problem, the concept 'bag' is simplified in our problem as labels of the whole signal are provided. Therefore, learning the 'interest' from the background would be evaded. The goal of our method is to learn various patterns of actions.

To simplify the notation, we define instances from the ith class, $i \in \{1, \ldots, m\}$, as a set $\boldsymbol{X}_i = \{\mathbf{x}_{i1}, \ldots, \mathbf{x}_{in}\}$, where $\mathbf{x}_{ij} \in \mathbf{R}^{d \times 1}, j \in \{1, \ldots, n\}$. In addition, each set $\boldsymbol{X}_i$ is associated with the label $Y_i \in \{0, 1\}$. When modeling latent patterns in the ith class, $Y_i = 1$, otherwise $Y_i = 0$. For each class, we assume it contains K latent classes, each of which corresponds to a cluster in the feature space. Intuitively, we assume there are 5 different patterns in class 'check gaps on the front door' and 'close both left door' respectively. Fig. 4 shows examples of patterns (latent classes) learned by MLPL. Multi-scale policy [59] is introduced in this paper to learn latent patterns from various semantic levels as well as mitigate uncertainty in determining concrete number of latent classes.

During the learning phase, each instance $\mathbf{x}_{ij}$ is associated with a latent variable $z_{ij} \in \{0, 1, \ldots, K\}$. Instance $\mathbf{x}_{ij}$ is in the kth positive cluster if $z_{ij} = k$ or in negative cluster if $z_{ij} = 0$. Considering that MIL is applied in our method to find distinctive latent classes, maximizing differences between one latent class and others is required. Besides, latent classes that belong to the ith class need to be distinguished from the jth class, where $i \neq j$. Consequently, from the feature distribution perspective,
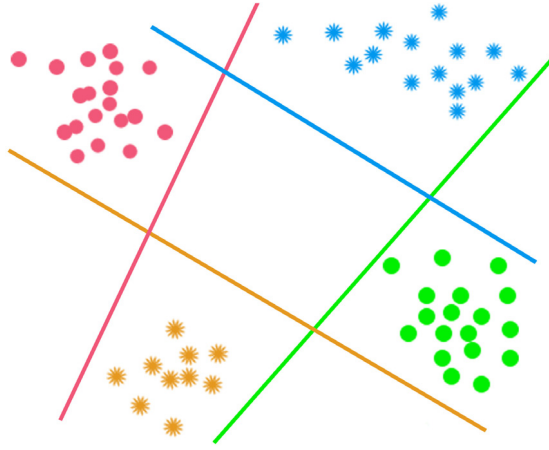
Fig. 3. Illustration of Max-margin Latent Pattern Learning (MLPL). Circle and star represent two specific classes. Two different colors of each class are related to two latent classes which can be distinctive descriptions of action patterns. The confidence score is obtained by calculating the distance between the latent class margin and the instance in the feature space. The combination of the confidence scores of all the latent classes forms the semantic feature representation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

a natural idea is to maximize the margin among intra- and inter-class. To meet that demand, multi-class SVM is an ideal solution. In particular, SVM with linear kernel is used for its generality and efficiency. Each latent class is associated with a linear classifier, in which $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ and weighting matrix is defined as

$$\mathbf{W}_i = [\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_K], \mathbf{w}_k \in \mathbf{R}^{d \times 1}, k \in [0, 1, \ldots, K] \tag{1}$$

where $\mathbf{w}_k$ represents the model of the kth latent class if k is positive; $\mathbf{w}_0$ denotes the negative cluster model. Therefore, we learned $K + 1$ linear classifiers for each class and $m * (K + 1)$ for total.

Intuitively, the latent label of the instance $\mathbf{x}_{ij}$ is determined as the most 'positive' one.

$$z_{ij} = \underset{k \in \{0, \ldots, K\}}{argmax} \mathbf{w}_k^T \mathbf{x}_{ij} \tag{2}$$

The multi-class hinge loss forces latent classifiers to be distinct from each other. It can be defined as:

$$L(\mathbf{W}; (\mathbf{x}_{ij}, z_{ij})) = \sum_{ij} max(0, 1 + \mathbf{w}_{r_{ij}}^T \mathbf{x}_{ij} - \mathbf{w}_{z_{ij}}^T \mathbf{x}_{ij}) \tag{3}$$

where $r_{ij} = \text{argmax}_{k \in \{0 \cdots K\}, k \neq z_{ij}} \mathbf{w}_k^T \mathbf{x}_{ij}$.

The objective function can then be defined as:

$$\min_{\mathbf{W}, z_{ij}} \sum_{k=0}^{K} \|\mathbf{w}_k\|^2 + \alpha \sum_{ij} max(0, 1 + \mathbf{w}_{r_{ij}}^T \mathbf{x}_{ij} - \mathbf{w}_{z_{ij}}^T \mathbf{x}_{ij}),$$
$$s.t. \ if \ Y_i = 1, z_{ij} > 0, \ and \ if \ Y_i = 0, z_{ij} = 0, \tag{4}$$

where $r_{ij} = \text{argmax}_{k \in \{0, \ldots, K\}, k \neq z_{ij}} \mathbf{w}_k^T \mathbf{x}_{ij}$.

The first term in Eq. (4) is for margin regularization and the second term is the multi-class hinge loss maximizing both inter- and intra-class margins. α balances the weight between two terms. In MLPL, all negative instances are utilized in the optimization step. Latent labels in each class are initialized by K-means and updated according to their 'positiveness' to each latent class. Constraint in Eq. (4) forces the function to learn latent patterns in the ith class. Though the optimization solution problem is a non-convex one, a local optimization can be guaranteed once the latent information is given [60]. We take $\mathbf{w}_k^T \mathbf{x}_{ij}$ as confidence scores of $\mathbf{x}_{ij}$ belonging to the kth latent class. Output descriptors are thus represented by the combination of confidence scores belonging to each latent class. Our method is different from [32] in three aspects. First, our work focuses on mining latent patterns rather than differentiating subjects of interests from the background. Second, latent classes are learned from all positive instances. The instance selection is removed as there are no background subjects in our problem. Third, instead of using the fixed number of latent classes, multi-scale strategy is adopted so that latent classes can be learnt at different semantic scales. We concatenate features learned by each scale.

Algorithm 1 Max-margin Latent Pattern Learning: MLPL

> ***Learning latent patterns***
> **Input**: Training Instances $\mathbf{X}=\{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_m\}$, number of latent patterns per class K
> **Loop** ℓ: Positive instances $\boldsymbol{X}_\ell$, negative instances $\mathbf{X} - \boldsymbol{X}_\ell$
> > **Initialization**: For negative instances, z_{ij}=0. For positive instances, z_{ij} is initialized by K-means.
>
> > **Iteration**: *N* times
> > > **Optimize $\mathbf{W}_\ell$**: Solve the multi-class SVM optimization problem
> >
> > $$\min_{\mathbf{W}_\ell} \sum_{k=0}^{K} \|\mathbf{w}_k\|^2 + \alpha \sum_{ij} max(0, \, 1 + \mathbf{w}_{r_{ij}}^\mathsf{T}\mathbf{x}_{ij} - \mathbf{w}_{z_{ij}}^\mathsf{T}\mathbf{x}_{ij}),$$
> >
> > $$where \; r_{ij} = \mathrm{argmax}_{k\in\{0\cdots K\}, k\neq z_{ij}}\mathbf{w}_k^\mathsf{T}\mathbf{x}_{ij}.$$
> >
> > > **Update z_{ij}**: For positive instances:
> >
> > $$z_{ij} = \mathrm{argmax}_{k\in\{1,\cdots,K\}}(\mathbf{w}_k^\mathsf{T}\mathbf{x}_{ij} - \mathbf{w}_0^\mathsf{T}\mathbf{x}_{ij}),$$
>
> **Output**: The learned classifiers $\mathbf{W}=[\boldsymbol{W}_1,\ldots,\boldsymbol{W}_m]$.
> -
> ***Semantic representation***
> **Input**: Instances $\mathbf{X}$
> **Output**: $\mathbf{F}(\mathbf{X}) = \mathbf{W}^\mathsf{T}\mathbf{X}$

In our MLCFL framework, the MLPL process is implemented on the concatenation of low- and mid-level features, where we obtain MLCF as the output of the feature learning stage and input of the classification stage.

4. Experiments and results

In this section, we first describe the datasets, evaluation method and experimental settings for the framework. Moreover, we compare our framework, MLCFL, with other closely related methods and test the framework with three classifiers. Then we demonstrate the effectiveness of Max-margin Latent Pattern Learning (MLPL), the complementary property of low-level and mid-level features. We further conduct intra-personal (tasks performed by one person) and inter-personal (tasks performed by different persons) experiments. In the end of this section, we explore the sensitivity of parameters.

4.1. Dataset

We evaluate the proposed Multi-Level Complementary Feature Learning (MLCFL) framework on three popular datasets, namely Skoda, WISDM and Opp. Experimental settings on the three datasets are listed as follows.

Skoda [39] Skoda Mini Checkpoint contains 16 manipulative gestures, which are performed in a car maintenance scenario and collected by 3D acceleration sensors. Only one worker's data are recorded, including actions such as 'open hood', 'close left hand door', etc. The sampling rate is 96 Hz. In our experiments, 10 actions and a null class from one right arm-sensor are taken into consideration. 4-fold cross validation is conducted.

WISDM [40] The dataset is collected by Android-based smart phones in 36 users' pockets. Records comprise 6 daily actions such as 'jogging', 'walking', etc. Data are collected through controlled laboratory conditions and the sampling rate is 20 Hz. We conduct 10-fold cross validation on this dataset.

Opp [37,38] Opportunity Activity Recognition is collected by 242 attributes in the scene that simulates a studio flat. The sampling rate of triaxial accelerometers is 64 Hz. In this paper, only data collected by the single inertial sensor on the right low arm are utilized, including 13 low-level actions ('clean', 'open' and 'close', etc.) and a null class. 5-fold cross validation is conducted on this dataset.

4.2. Evaluation method

In HAR, data are severely unbalanced. Some classes are overrepresented while others are scarce. To adapt this characteristic, we apply the weighted F_1 score for evaluation.

Weighted F_1 Score We denote TP and FP as the number of true positives and false positives respectively, and FN as the number of false negatives. Thus, weighted F_1 score can be formulated as follows:

$$precision_i = \frac{TP_i}{TP_i + FP_i} \tag{5}$$

a

b
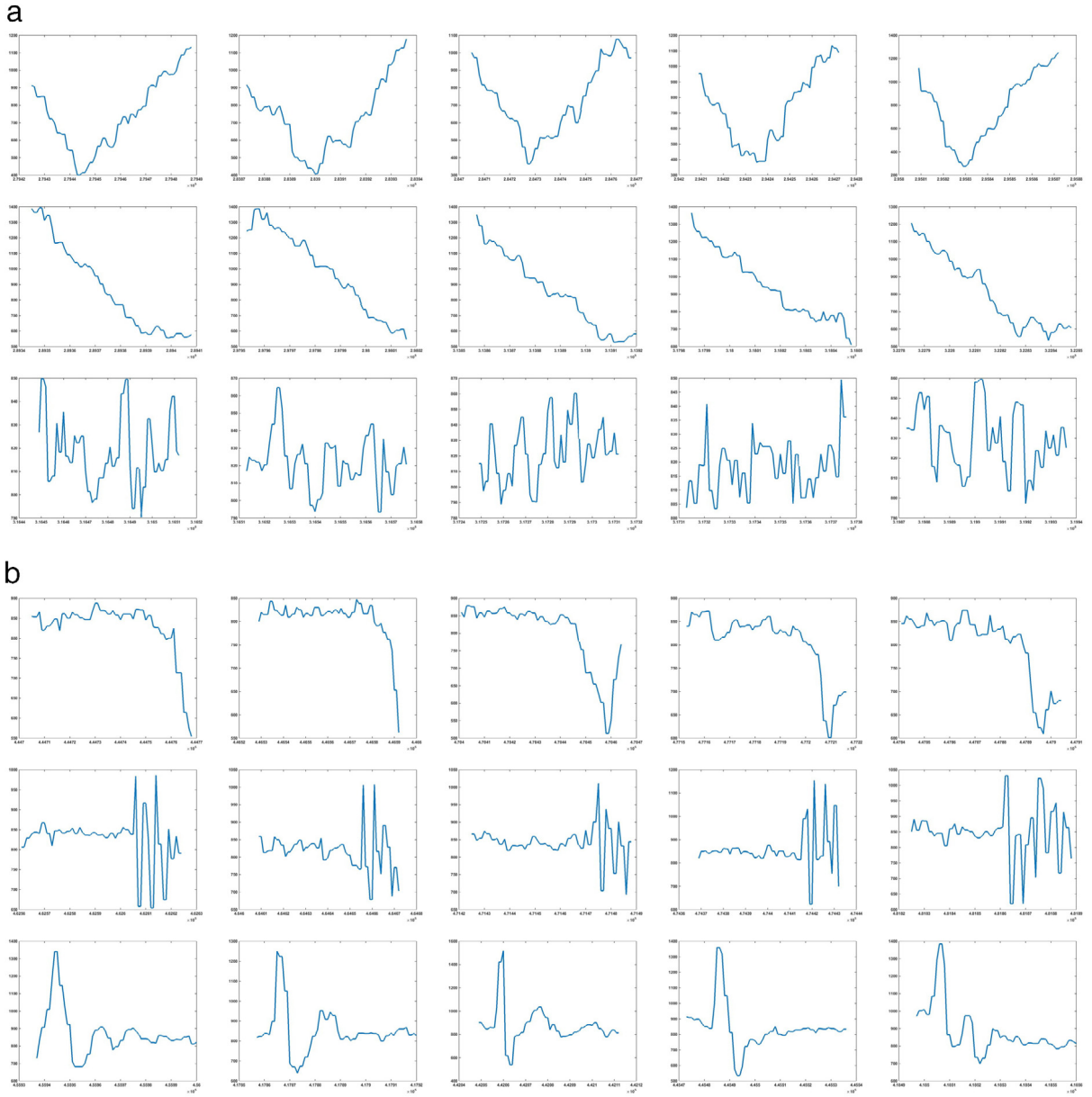
Fig. 4. Illustrations of latent class. Two classes in Skoda, 'check gaps on the front door' (a) and 'close both left door' (b), with their three latent classes are shown in raw signal. Each row refers to a latent class, where five samples are selected. Rows compare differences among latent classes and columns show the similarity in one specific latent class.

$$recall_i = \frac{TP_i}{TP_i + FN_i} \tag{6}$$

$$F_1 = \sum_i 2 \cdot w_i \frac{precision_i \cdot recall_i}{precision_i + recall_i} \tag{7}$$

in which i refers to the class index and w_i is the proportion of the ith class in all the classes. Unless mentioned otherwise, we adopt weighted F_1 score throughout experiments.

4.3. Experiment setup

To compare with other methods, we follow the dataset settings in Zeng's work [45]. Specifically, in all three datasets, we utilize data from a single triaxial inertial sensor. Sensor data are segmented into frames using a sliding window with the

Table 1
Comparison (classification accuracy) of our presented MLCFL algorithm with existing methods on three datasets.

Feature	Skoda	WISDM	Opp
Statistical	83.6	95.8	69.4
FFT	81.3	96.4	69.2
ECDF-PCA [42]	85.4	98.1	69.4
W-CNN [45]	88.2	96.9	/[a]
Stat-MLCF	88.4	98.2	72.8
FFT-MLCF	87.6	98.3	**73.5**
ECDF-PCA-MLCF	**88.7**	**98.8**	72.6

[a] Performances of W-CNN on 13 actions are not accessible.

size of 64 continuous samples and with 50% overlap. We also test window size of 48, 80 and get F1-value 86.8%, 89.5% on Skoda respectively. Since a frame may contain different labels, only frames with a single label are taken into consideration in Skoda and WISDM. But when it comes to Opp, to obtain enough samples, the label of the frame is determined according to the dominant label. Consequently, Skoda, WISDM and Opp contain around 22,000, 33,000 and 21,000 frames, respectively. In cross-validation, folds are created by randomly choosing samples from the dataset.

To demonstrate the generalization ability of the proposed framework, we test three prevalent low-level features, namely statistical values, FFT coefficients and ECDF-PCA. We calculate statistical values (mean, standard deviation, energy, entropy and correlation coefficients) and FFT coefficients the same way as Plötz et al. [42] and obtain 23-dimension and 30-dimension feature vectors for each frame respectively. In our practice, statistical values and FFT coefficients with Z-score normalization gain significant improvement in performances than original ones. For ECDF-PCA, we normalize the raw signal through 60 points inverse equal probability interpolation along each channel. Then the PCA process is conducted, where 30 principal components (30 dimensions) are taken as output features. In the mid-level learning phase, the length of the sub-frame for encoding process is set to 20, and the size of the dictionary is set to 300. We concatenate features from three channels of a single triaxial inertial sensor and yield 900-dimension sparse representations. In the high-level learning phase, the input feature is the concatenation of a low-level feature (statistical values with 23 dimensions, FFT coefficients with 30 dimensions, ECDF-PCA with 30 dimensions) and its corresponding mid-level feature (900 dimensions). The weight parameter α in Eq. (4) is set to 1. Besides, the latent-class number K for each class is set to 5 and 10 in two scales. In the step of optimizing **W**, we apply linear SVM [61] to handle multi-class classification problem. The number of iterations N is set to 3. Without loss of generality, settings for features are the same throughout experiments.

We test the performance of the MLCFL framework with three representative classifiers, namely K-Nearest Neighbors (KNN), Support Vector Machine (SVM) and Nearest Centroid Classifier (NCC). In the K-Nearest Neighbors algorithm (KNN), samples are classified by a majority vote of their neighbors and assigned to the most common label among its K nearest neighbors. K is set to 5 throughout our experiments. Besides, SVM with linear kernel is a prevalent and efficient solution in classification. Specifically, Liblinear [61] is used in our experiment.[1] Nearest Centroid Classifier (NCC) [37] calculates Euclidean Distance between test samples and each centroid (mean) of training classes. The predicted label is allocated according to the nearest class centroid.

4.4. Experiments

In the following sections, we conduct experiments in five respects: (1) comparing with existing methods; (2) conducting classifiers experiments on MLCF; (3) evaluating the effectiveness of MLPL; (4) exploring the complementary property of the low-level feature and mid-level feature; (5) performing intra- and inter-personal experiments.

4.4.1. Comparing with existing methods

In this section, comparisons of our framework with several published works are shown in Table 1, including representative methods: statistical values, FFT coefficients, ECDF-PCA [42] and W-CNN [45]. W-CNN applies CNN with the partial weight sharing technique to perform HAR tasks. This method achieves high performances in single body-sensor based classification tasks on three popular datasets. Statistical values, FFT coefficients and ECDF-PCA are respectively utilized as low-level features in our framework. Their corresponding output features are abbreviated to Stat-MLCF, FFT-MLCF and ECDF-PCA-MLCF. We follow the metrics in the W-CNN work and present results using KNN in the form of classification accuracy.

Table 1 illustrates that the general performance of MLCF is the highest. Since our algorithm can be applied as the refining process of these low-level features, improvements on statistical values, FFT coefficients and ECDF-PCA are obvious. Despite parameters being generalized on all three datasets, results show that our method still achieves better results than the state-of-the-art method, W-CNN [45], on available Skoda and WISDM. This shows the effectiveness of MLCFL which integrates advantages of low-, mid- and high-level analyzes.

---

[1] Different from the Liblinear in high-level feature learning, here we use it as a classifier.

Table 2
Comparison (weighted F_1 score) among low-level features, their mid-level features and MLCF on three datasets using three classifiers.

KNN									
Feature	Skoda			WISDM			Opp		
	low-level	mid-level	MLCF	low-level	mid-level	MLCF	low-level	mid-level	MLCF
Statistical	83.6	85.3	**88.5**	95.6	96.9	**98.2**	67.9	68.5	**70.8**
FFT	81.3	82.7	**87.6**	96.3	97.1	**98.3**	67.3	69.1	**71.7**
ECDF-PCA	85.4	84.3	**88.7**	98.1	98.0	**98.8**	67.6	67.8	**70.5**
SVM									
Feature	Skoda			WISDM			Opp		
	low-level	mid-level	MLCF	low-level	mid-level	MLCF	low-level	mid-level	MLCF
Statistical	54.9	81.8	**84.4**	73.5	92.3	**93.9**	53.5	63.7	**66.2**
FFT	50.8	81.6	**82.6**	86.6	93.6	**95.4**	50.6	64.8	**66.7**
ECDF-PCA	40.3	82.4	**83.7**	77.4	95.5	**96.1**	47.4	64.6	**65.7**
NCC									
Feature	Skoda			WISDM			Opp		
	low-level	mid-level	MLCF	low-level	mid-level	MLCF	low-level	mid-level	MLCF
Statistical	49.6	64.4	**65.4**	73.3	83.1	**83.3**	45.8	48.4	**51.5**
FFT	40.6	61.9	**67.2**	79.6	81.1	**88.6**	47.4	43.2	**53.9**
ECDF-PCA	37.2	63.6	**66.6**	65.8	87.0	**89.8**	49.1	52.0	**52.3**

4.4.2. Results from different classifiers

In this section, we demonstrate the 'good' feature property of MLCF by presenting results from three classifiers. Results from all three datasets are shown in Table 2.

It can be observed from Table 2 that MLCF performs better than low- and mid-level features in all three classifiers and achieves the best performance using KNN. Though the sparse representation is preferred by linear SVM, MLCF of 187 (Skoda), 102 (WISDM) and 238 (Opp) dimensions still show obvious enhancement compared with mid-level features of 900 dimensions. Besides, the decision process of NCC is based on the distribution of the feature, which suggests that the distribution of MLCF in the feature space is more regular and discriminative.

4.4.3. The effectiveness of the MLPL process

In this section, we demonstrate the effectiveness of the Max-margin Latent Pattern Learning (MLPL) method. In particular, we evaluate the MLPL process on three low-level features, their mid-level features and Compl features (the concatenation of low- and mid-level features). Classification results based on KNN are shown in Fig. 5.

Fig. 5 shows that MLPL process achieves remarkable improvements on both mid-level and Compl features. The improvement attributes to their sparse representation which is preferred by the linear SVM solution adopted in MLPL [62]. Compl feature also takes advantage of the complementary property which is further discussed in Section 4.4.4. Results also show that MLCF, which is generated by implementing MLPL on the Compl feature, obtains the highest performance within group comparison on all three datasets. By contrast, performances of low-level features through MLPL degrade, which is partly due to the low-feature being poorly linearly separable. MLPL is designed to explore distinctive distributions in the feature space. Directly implementing MLPL on the low-level feature is not efficient considering that its low dimensional description is not preferred by the linear classifier inside MLPL. In this condition, MLPL would fail to yield strict boundaries among latent classes.

4.4.4. Linear complementary property

In this section, we demonstrate how complementary property of low-level and mid-level features improves the performance. In particular, experiments are conducted to present intuitive comparisons among low-, mid- and Compl features using linear SVM. Results are shown in Fig. 6.

Fig. 6 shows that Compl feature performs relatively better than low-level and mid-level features separately with linear classifier. As the mid-level feature is based on statistical description of motion primitives, it contains information from a structural perspective. By contrast, the low-level feature extracted from the raw data can capture detailed information like statistics property (Statistics values), frequency property (FFT coefficients) and amplitude distribution (ECDF-PCA). Therefore, low- and mid-level features are designed to describe the signal from different perspectives [15]. Besides, SVM with linear kernel shows advantages in this condition. This is because in the modeling process, feature weights have inherent characters for feature selection. This property is especially exploited and enhanced in MLPL considering that clusters of linear classifiers are involved. Furthermore, latent classes are initialized by K-means in MLPL, thus clusters are likely to be discriminative in distribution. All those factors contribute to gaining better performances when implementing MLPL on Compl feature.
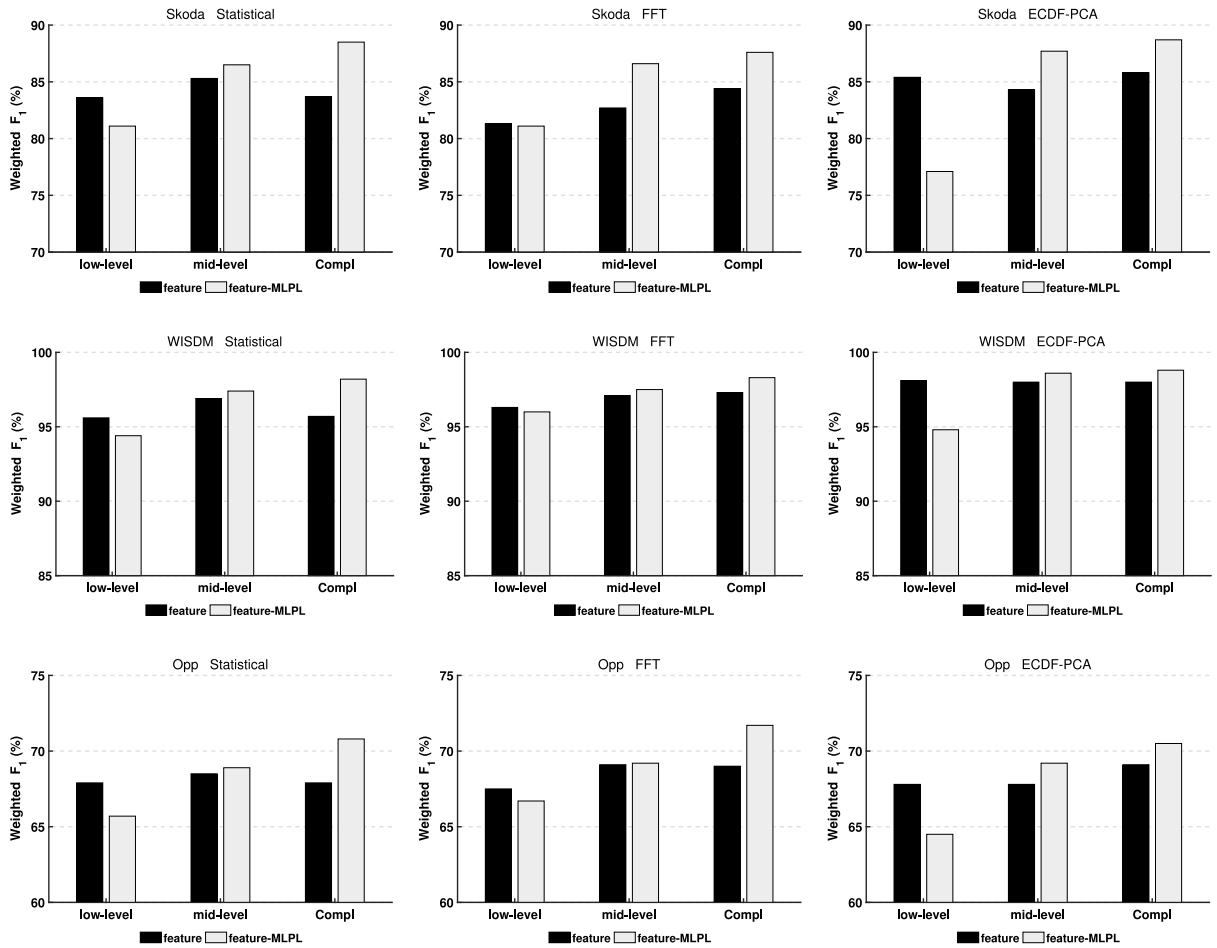
Fig. 5. Illustrations of the effectiveness of the MLPL process. In each figure, from left to right, black bars show results of low-level feature, mid-level feature and Compl feature; white bars show corresponding results after the MLPL process.
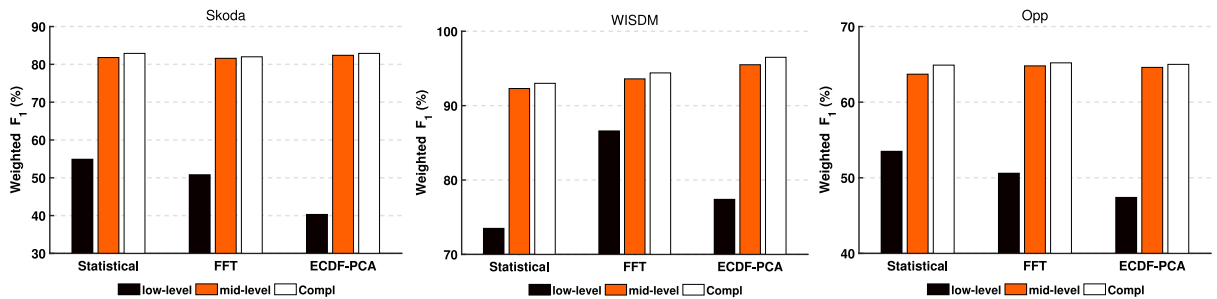

Fig. 6. Comparison of Compl feature toward low- and mid-level features using linear SVM.

4.4.5. Intra- and inter-personal experiments

Intra-personal and inter-personal experiments are also conducted to evaluate our framework in daily life scenes. We perform intra-personal tests on Skoda and inter-personal tests on WISDM. In intra-personal experiments, since data in Skoda is recorded by one subject, we divide each class into 6 parts in time sequences and perform 6-fold cross validation. Results obtained by KNN and SVM are shown in Table 3. In inter-personal experiments, we randomly divide 36 subjects into 10 groups. 10-fold cross validation is conducted on this dataset. The results are shown in Table 4.

MLCF with SVM and KNN generally gain the best performance in both intra-personal and inter-personal test, which can be observed from Table 3 and Table 4, confirming the robustness of our feature learning framework. In intra-personal tests, MLCF yields better performance in KNN than in SVM, suggesting that similarity strategy adopted by KNN is more suitable for

Table 3

Comparison (weighted F_1 score) among low-level, mid-level features and corresponding MLCF in intra-personal experiments on Skoda using KNN and SVM.

Feature	KNN			SVM		
	low-level	mid-level	MLCF	low-level	mid-level	MLCF
Statistical	80.4	77.7	**84.4**	53.4	76.8	**78.9**
FFT	77.9	70.6	**82.5**	50.2	75.3	**76.9**
ECDF-PCA	81.5	75.0	**83.0**	40.0	76.1	**77.6**

Table 4

Comparison (weighted F_1 score) among low-level, mid-level features and corresponding MLCF in inter-personal experiments on WISDM using KNN and SVM.

Feature	KNN			SVM		
	low-level	mid-level	MLCF	low-level	mid-level	MLCF
Statistical	78.0	76.8	**80.7**	71.0	79.3	**81.4**
FFT	**83.7**	68.3	**83.6**	85.1	77.0	**86.5**
ECDF-PCA	77.0	74.5	**79.4**	73.8	81.0	**82.8**
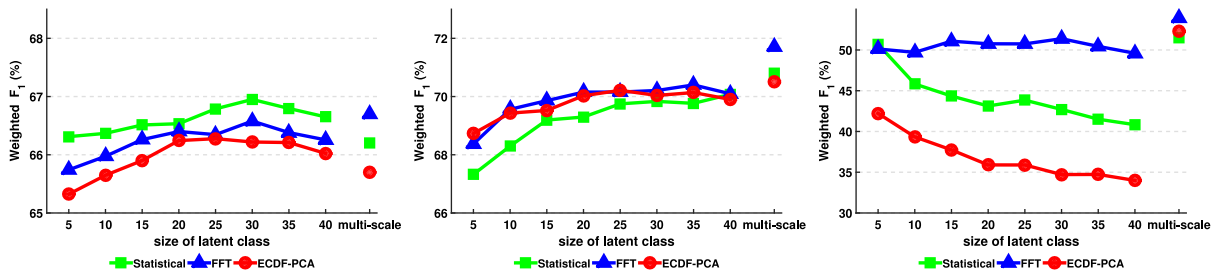
Fig. 7. The effect of the latent class size on MLCF. From left to right, classification results of MLCF are obtained by SVM, KNN and NCC, respectively. The 'multi-scale' refers to two scales of 5 and 10.

MLCF in inter-personal tasks. Besides, high performance MLCF obtained can be contributed to the fact that action patterns of specific subject have been learned during training and thus can be decently represented in testing. By contrast, in inter-personal experiments, MLCF performs better in SVM than in KNN, owing to SVM's flexibility [63] in dealing with varieties among subjects.

4.4.6. Complexity

We also estimate the overall complexity. Approximately, given fixed-length sampling series, in the first (low-level) stage, the complexity depends on types of low-level features extracted. In the second (mid-level) stage, the complexity is proportional to the product of the size of dictionary and frame. In the third (high-level) stage, the complexity is proportional to the size of latent classes. A typical training and testing time in our experiment at Sokda using FFT as the low feature and Liblinear as classifier is 16 min in training 16,500 samples and 1 min in testing 5500 samples (i7-7700HQ, 2.80 GHZ, 8 GB).

4.5. Sensitivity of parameters

This section elucidates the evaluation of the variable sensitivity in our framework, including the number of latent classes in the MLPL phase and the dictionary size in the mid-level feature learning phase.

4.5.1. Size of latent classes

We evaluate the sensitivity of the latent class size in MLPL on the Opp dataset, the results of which are shown in Fig. 7. Latent class sizes are ranging from 5 to 40 and the multi-scale is of 5 and 10. Compared with the single scale, the multi-scale achieves generally the best performance, showing the robustness and effectiveness of learning latent patterns from various semantic levels. From Fig. 7, it can also be observed that the tendency of the weighted F_1 score is inconsistent according to different classifiers. With SVM, the increasing number of latent classes has positive influence before the size reaches around 30 and after that, the performance slightly degrades. The increase occurs under the circumstance that the specific class is clustered into proper number of groups which are linearly separable in feature space while the degradation can be caused by the case that latent classes are too particular and thus over-represented. Specifically, during MLPL, original properly divided latent classes would be further clustered into small clusters, which are not linearly separable. As MLPL outputs the description of the confidence score of each latent class, over clustering would lead to confusion. The max-margin strategy SVM adopts would be sensitive in this confused description. In terms of KNN, the performance is positively related with increment of the latent class's size, suggesting K-nearest similarity strategy is resistant to over-clustered cases.
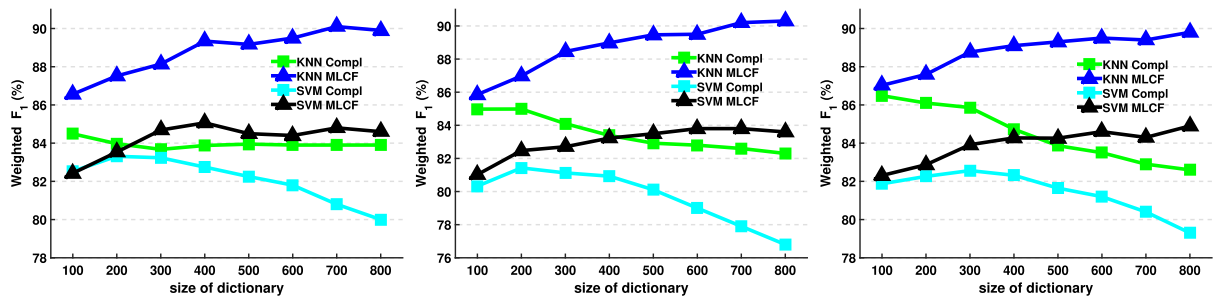
Fig. 8. The influence of the motion-primitive dictionary size on Compl feature and MLCF. From left to right, Compl feature and MLCF are derived from statistical values, FFT coefficients and ECDF-PCA, respectively.

4.5.2. Size of dictionary

We also change the dictionary size in mid-level feature learning phase to evaluate its influences on Compl feature and MLCF. The experiment is conducted on the Skoda dataset with the size of the motion-primitive dictionary ranging from 100 to 800.

As shown in Fig. 8, performances of MLCF positively correlate with the dictionary size in both SVM and KNN while performances of Compl feature increase and then slightly degrade in SVM but continuously degrade in KNN. Although the increase in dictionary size involves more detailed motion-primitive description and the statistical process makes feature robust to noise, larger size may lead to excessively redundant representation, especially when features are concatenated in three channels. Under this condition, when Compl features are directly fed into classifiers, data around 22,000 frames are not sufficient in training models. In contrast, MLCF yields better results when the performance of the Compl feature turns to degrade.

5. Conclusion

In this paper, we present the MLCFL framework for signal processing in HAR. The framework consists of three parts, obtaining low-level, mid-level and high-level features separately. The low-level feature captures property of raw signals. The mid-level feature achieves component-based representation through hard coding process and occurrence statistics. At high level, the latent semantic learning method MLPL is proposed to mine latent action patterns from concatenation of low- and mid-level features, during which the semantic representation can be achieved. Our framework achieves the state-of-the-art performances, 88.7%, 98.8% and 72.6% (weighted F_1 score) respectively, on Skoda, WISDM and OPP datasets. Given that the MLPL method has the ability of discovering various patterns inside the specific class, it is possible to apply this framework in more challengeable scenarios, like tasks without full annotations. So a potential improvement of our future work is to merge instance selection processing into current framework in order to deal with weakly learning problems.

Author contributions

Designed the program and wrote the code: YX ZS XZ SD YG. Provided the data: YX YG. Performed the experiments: ZS XZ SD. Analyzed the data: YG SD YW. Wrote the paper:YX ZS YW YG XZ SD YF EC.

Acknowledgments

References

[1] S. Chennuru, P.-W. Chen, J. Zhu, J.Y. Zhang, Mobile lifelogger–recording, indexing, and understanding a mobile users life, in: MobiCASE, 2010, pp. 263–281.
[2] P. Wu, J. Zhu, J.Y. Zhang, Mobisens: A versatile mobile sensing platform for real-world applications, Mob. Netw. Appl. 18 (1) (2013) 60–80.
[3] P. Wu, H.-K. Peng, J. Zhu, Y. Zhang, Senscare: Semi-automatic activity summarization system for elderly care, in: MobiCASE, 2011, pp. 1–19.
[4] K. Förster, D. Roggen, G. Tröster, Unsupervised classifier self-calibration through repeated context occurences: is there robustness against sensor displacement to gain, in: ISWC, IEEE, 2009, pp. 77–84.
[5] T.T. Ngo, Y. Makihara, H. Nagahara, Y. Mukaigawa, Y. Yagi, Similar gait action recognition using an inertial sensor, Pattern Recognit. 48 (4) (2015) 1289–1301.
[6] L. Santos, K. Khoshhal, J. Dias, Trajectory-based human action segmentation, Pattern Recognit. 48 (2) (2015) 568–579.

[7] O.D. Lara, M.A. Labrador, A survey on human activity recognition using wearable sensors, Comm. Surv. 15 (3) (2013) 1192–1209.

[8] A. Bulling, U. Blanke, B. Schiele, A tutorial on human activity recognition using body-worn inertial sensors, CSUR 46 (3) (2014) 33.

[9] L. Bao, S.S. Intille, Activity recognition from user-annotated acceleration data, in: Pervasive Computing, Springer, 2004, pp. 1–17.

[10] N. Ravi, N. Dandekar, P. Mysore, M.L. Littman, Activity recognition from accelerometer data, in: AAAI, Vol. 5, 2005, pp. 1541–1546.

[11] T. Huynh, B. Schiele, Analyzing features for activity recognition, in: sOc-EUSAI, 2005, pp. 159–163.

[12] W.-J. Kang, J.-R. Shiu, C.-K. Cheng, J.-S. Lai, H.-W. Tsao, T.-S. Kuo, The application of cepstral coefficients and maximum likelihood method in emg pattern recognition, Biomed. Eng. 42 (8) (1995) 777–785.

[13] S. Bhattacharya, P. Nurmi, N. Hammerla, T. Plötz, Using unlabeled data in a sparse-coding framework for human activity recognition, Pervasive Mob. Comput. 15 (2014) 242–262.

[14] C. Vollmer, H.-M. Gross, J.P. Eggert, Learning features for activity recognition with shift-invariant sparse coding, in: ICANN, 2013, pp. 367–374.

[15] T. Huynh, U. Blanke, B. Schiele, Scalable recognition of daily activities with wearable sensors, in: LoCA, 2007, pp. 50–67.

[16] N.Y. Hammerla, R. Kirkham, P. Andras, T. Ploetz, On preserving statistical characteristics of accelerometry data using their empirical cumulative distribution, in: ISWC, 2013, pp. 65–68.

[17] R. Grosse, R. Raina, H. Kwong, A.Y. Ng, Shift-invariance sparse coding for audio classification, in: UAI, 2007, pp. 4–5.

[18] Y. Zhang, Y. Zhang, E. Swears, N. Larios, Z. Wang, Q. Ji, Modeling temporal interactions with interval temporal bayesian networks for complex activity recognition, TPAMI 35 (10) (2013) 2468–2483.

[19] H.H. Bui, D.Q. Phung, S. Venkatesh, H. Phan, The hidden permutation model and location-based activity recognition, in: AAAI, Vol. 8, 2008, pp. 1345–1350.

[20] Y. Liu, L. Nie, L. Liu, D.S. Rosenblum, From action to activity: Sensor-based activity recognition, Neurocomputing 181 (2015) 108–115.

[21] B. Chikhaoui, S. Wang, H. Pigot, A frequent pattern mining approach for adls recognition in smart environments, in: AINA, IEEE, 2011, pp. 248–255.

[22] T. Gu, S. Chen, X. Tao, J. Lu, An unsupervised approach to activity recognition and segmentation based on object-use fingerprints, Data Knowl. Eng. 69 (6) (2010) 533–544.

[23] E. Kim, S. Helal, D. Cook, Human activity recognition and pattern discovery, Pervasive Comput. 9 (1) (2010) 48–53.

[24] P. Palmes, H.K. Pung, T. Gu, W. Xue, S. Chen, Object relevance weight pattern mining for activity recognition and segmentation, Pervasive Mob. Comput. 6 (1) (2010) 43–57.

[25] A. Zinnen, C. Wojek, B. Schiele, Multi activity recognition based on bodymodel-derived primitives, in: LoCA, 2009, pp. 1–18.

[26] O. Amft, M. Kusserow, G. Tröster, Probabilistic parsing of dietary activity events, in: BSN, 2007, pp. 242–247.

[27] M. Stikic, T. Huynh, K.V. Laerhoven, B. Schiele, ADL recognition based on the combination of RFID and accelerometer sensing, in: PervasiveHealth, 2008, pp. 258–263.

[28] T. Huynh, M. Fritz, B. Schiele, Discovery of activity patterns using topic models, in: Ubicomp, 2008, pp. 10–19.

[29] J.K. Aggarwal, M.S. Ryoo, Human activity analysis: A review, CSUR 43 (3) (2011) 16.

[30] T. Malisiewicz, A. Gupta, A.A. Efros, Ensemble of exemplar-svms for object detection and beyond, in: ICCV, 2011, pp. 89–96.

[31] T.G. Dietterich, R.H. Lathrop, T. Lozano-Pérez, Solving the multiple instance problem with axis-parallel rectangles, Artificial Intelligence 89 (1) (1997) 31–71.

[32] X. Wang, B. Wang, X. Bai, W. Liu, Z. Tu, Max-margin multiple-instance dictionary learning, in: ICML, 2013, pp. 846–854.

[33] M. Ziaeefard, R. Bergevin, Semantic human activity recognition: a literature review, Pattern Recognit. 48 (8) (2015) 2329–2345.

[34] Y. Yi, M. Lin, Human action recognition with graph-based multiple-instance learning, Pattern Recognit. (2015).

[35] M. Stikic, D. Larlus, S. Ebert, B. Schiele, Weakly supervised recognition of daily life activities with wearable sensors, TPAMI 33 (12) (2011) 2521–2537.

[36] M. Stikic, B. Schiele, Activity recognition from sparsely labeled data using multi-instance learning, in: LoCA, 2009, pp. 156–173.

[37] R. Chavarriaga, H. Sagha, A. Calatroni, S.T. Digumarti, G. Tröster, J.d.R. Millán, D. Roggen, The opportunity challenge: A benchmark database for on-body sensor-based activity recognition, Pattern Recognit. Lett. 34 (15) (2013) 2033–2042.

[38] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha, et al., Collecting complex activity datasets in highly rich networked sensor environments, in: INSS, 2010, pp. 233–240.

[39] P. Zappi, C. Lombriser, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, G. Tröster, Activity recognition from on-body sensors: accuracy-power trade-off by dynamic sensor selection, in: WSN, 2008, pp. 17–33.

[40] J.R. Kwapisz, G.M. Weiss, S.A. Moore, Activity recognition using cell phone accelerometers, SIGKDD 12 (2) (2011) 74–82.

[41] M. Field, D. Stirling, Z. Pan, M. Ros, F. Naghdy, Recognizing human motions through mixture modeling of inertial data, Pattern Recognit. 48 (8) (2015) 2394–2406.

[42] T. Plötz, N.Y. Hammerla, P. Olivier, Feature learning for activity recognition in ubiquitous computing, in: IJCAI, Vol. 22, 2011.

[43] M. Zhang, A.A. Sawchuk, Motion primitive-based human activity recognition using a bag-of-features approach, in: IHI, 2012, pp. 631–640.

[44] T.M. Hospedales, J. Li, S. Gong, T. Xiang, Identifying rare and subtle behaviors: A weakly supervised joint topic model, TPAMI 33 (12) (2011) 2451–2464.

[45] M. Zeng, L.T. Nguyen, B. Yu, O.J. Mengshoel, J. Zhu, P. Wu, J. Zhang, Convolutional neural networks for human activity recognition using mobile sensors, in: MobiCASE, 2014, pp. 197–205.

[46] T. Tamura, M. Sekine, M. Ogawa, T. Togawa, Y. Fukui, Classification of acceleration waveforms during walking by wavelet transform, Methods Inf. Med. 36 (4–5) (1997) 356–359.

[47] Z. He, L. Jin, Activity recognition from acceleration data based on discrete consine transform and svm, in: SMC, Vol. 4, 2009, pp. 5041–5044.

[48] Z.-Y. He, L.-W. Jin, Activity recognition from acceleration data using AR model representation and SVM, in: ICMLC, Vol. 4, 2008, pp. 2245–2250.

[49] U. Blanke, B. Schiele, Daily routine recognition through activity spotting, in: LoCA, 2009, pp. 192–206.

[50] B. Hartmann, N. Link, Gesture recognition with inertial sensors and optimized DTW prototypes, in: Systems Man and Cybernetics, SMC, 2010 IEEE International Conference on, IEEE, 2010, pp. 2102–2109.

[51] T. Stiefmeier, D. Roggen, G. Ogris, P. Lukowicz, G. Tröster, Wearable activity tracking in car manufacturing, IEEE Pervasive Comput. 7 (2) (2008).

[52] L.-V. Nguyen-Dinh, D. Roggen, A. Calatroni, G. Tröster, Improving online gesture recognition with template matching methods in accelerometer data, in: Intelligent Systems Design and Applications, ISDA, 2012 12th International Conference on, IEEE, 2012, pp. 831–836.

[53] L.-V. Nguyen-Dinh, A. Calatroni, G. Tröster, Robust online gesture recognition with crowdsourced annotations, J. Mach. Learn. Res. 15 (2014) 3187–3220.

[54] J.-L. Reyes-Ortiz, L. Oneto, A. Ghio, A. Samá, D. Anguita, X. Parra, Human activity recognition on smartphones with awareness of basic activities and postural transitions, in: International Conference on Artificial Neural Networks, Springer, 2014, pp. 177–184.

[55] S. Ha, J.-M. Yun, S. Choi, Multi-modal convolutional neural networks for activity recognition, in: SMC, 2015, pp. 3017–3022.

[56] J.B. Yang, M.N. Nguyen, P.P. San, X.L. Li, S. Krishnaswamy, Deep convolutional neural networks on multichannel time series for human activity recognition, in: IJCAI, 2015, pp. 25–31.

[57] F.J. Ordóñez, D. Roggen, Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition, Sensors 16 (1) (2016) 115.

[58] F. Palumbo, C. Gallicchio, R. Pucci, A. Micheli, Human activity recognition using multisensor data fusion based on reservoir computing, JAISE 8 (2) (2016) 87–107.

[59] S. Park, H. Kautz, Hierarchical recognition of activities of daily living using multi-scale, multi-perspective vision and RFID, in: Intelligent Environments, IET, 2008, pp. 1–4.
[60] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, TPAMI 32 (9) (2010) 1627–1645.
[61] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, Liblinear: A library for large linear classification, JMLR 9 (2008) 1871–1874.
[62] X. Li, H. Wang, B. Gu, C.X. Ling, Data sparseness in linear SVM, in: IJCAI, 2015, pp. 3628–3634.
[63] A. Belousov, S. Verzakov, J. Von Frese, A flexible classification approach with optimal generalisation performance: support vector machines, Chemometr. Intell. Lab. Syst. 64 (1) (2002) 15–25.