

Action2Activity: Recognizing Complex Activities from Sensor Data

Ye Liu[†], Liqiang Nie[†], Lei Han[§], Luming Zhang[†], David S. Rosenblum[†]

[†] School of Computing, National University of Singapore

[§] Department of Computer Science, Hong Kong Baptist University

{liuye, david}@comp.nus.edu.sg, leihan@comp.hkbu.edu.hk, {nieliqiang, zglumg}@gmail.com

Abstract

As compared to simple actions, activities are much more complex, but semantically consistent with a human’s real life. Techniques for action recognition from sensor generated data are mature. However, there has been relatively little work on bridging the gap between actions and activities. To this end, this paper presents a novel approach for complex activity recognition comprising of two components. The first component is temporal pattern mining, which provides a mid-level feature representation for activities, encodes temporal relatedness among actions, and captures the intrinsic properties of activities. The second component is adaptive Multi-Task Learning, which captures relatedness among activities and selects discriminant features. Extensive experiments on a real-world dataset demonstrate the effectiveness of our work.

1 Introduction

We are living in an era of wearable and environmental sensors. Activity recognition from sensor data plays an essential role in many applications. Consider application scenarios in healthcare as an example [Nie *et al.*, 2015]. Caregivers use sensors to track and analyze the Activities of Daily Living (ADL) of elderly people, which enables the caregivers to provide proactive assistance [Pansiot *et al.*, 2007]. Another application scenario is context-aware music recommendation [Wang *et al.*, 2012], which senses context information about the activity a user is doing and recommends music suitable for the activity.

Several research efforts have been dedicated to the recognition of simple activities¹ with simple features [Wang *et al.*, 2012; Pansiot *et al.*, 2007; Ravi *et al.*, 2005]. For example, Ravi *et al.* [2005] designed a classifier to distinguish eight actions, namely *standing*, *walking*, *running*, *climbing up stairs*, *climbing down stairs*, *vacuuming*, *brushing teeth*, and *sit-ups*. In real life, however, human activity is much more complex than such disjoint occurrences of simple actions. Consider cooking as an example. It involves a sequence of actions over

¹In this work, we use the term activity to stand for a complex activity, and we use the term action to refer to a simple activity.

time, and some of those actions may happen simultaneously or concurrently, such as walking followed by reaching for the fridge, or standing while fetching or cutting up food.

Recognizing complex activities from sensor data is non-trivial due to the following reasons. First, the actions underlying an activity are not independent. In particular, within one activity, the temporal relatedness among actions may manifest itself in many forms. Such sophisticated temporal combinations lead to semantic meanings for activity understanding. Second, commonality or semantic relatedness exists across multiple activities. For example, making coffee may be much more similar to making bread than to relaxing, in terms of action patterns. Third, features used to represent activities usually suffer from the curse of dimensionality, and in fact not all features are discriminative.

To tackle the above challenges, we present an approach that consists of two components. The first is an algorithm for temporal pattern mining, which encodes various temporal relations among actions, including sequential, interleaved, and concurrent relations. In particular, we presume that each pattern is a set of temporally inter-related actions, and that each activity can be characterized by a set of patterns. Our algorithm automatically discovers frequent temporal patterns from action sequences and uses the mined patterns to characterize activities. The second component is activity recognition. It treats each activity as a task and uses an adaptive multi-task learning (aMTL) algorithm to capture and model the relatedness among these tasks. In addition, it is capable of identifying discriminant task-specific and task-sharing features. As an added benefit, it alleviates the problem of insufficient training samples, since it enables sharing of training instances among tasks.

We summarize the contributions as follows:

- We present a novel approach to identify temporal patterns among actions for activity representation.
- We present a novel multi-task learning approach to boost the performance of activity recognition.

2 Related Work

Recognizing simple actions from sensor data has attracted much attention [Wang *et al.*, 2012; Pansiot *et al.*, 2007; Ravi *et al.*, 2005]. For example, the work introduced by

Ravi *et al.* [2005] classified eight daily actions using shallow classifiers, such as kNN, SVM and Naïve Bayes, and achieved overall accuracy of 95%. Another work by Wang *et al.* [2012] utilized a Naïve Bayes model to recognize six daily actions (*working, studying, running, sleeping, walking and shopping*) for music recommendation and obtained promising performance. However, as the nature of human activity is complex, people often perform not just a single action in isolation, but several actions in diverse combinations. The key to modeling activities is to capture the temporal relations among actions [Zhuo *et al.*, 2009]. Few of the previous efforts have been dedicated to capturing the relatedness among actions and the high-level semantics over groups of actions.

A set of approaches have been proposed to explore the simple relations among actions [Yang, 2009; Ryoo and Aggarwal, 2006]. Dynamical model approaches (e.g., HMM [Rabiner, 1989] and CRF [Lafferty *et al.*, 2001]) could capture the simple relations between actions, such as sequential relations. However, they are unable to characterize the complex relations in real-world activity data. This is because human activities may contain several overlapped actions, and treating these activities simply as sequential data may lead to information loss. Moreover, they fail to capture the higher-order temporal relatedness among actions. Bayesian network-based approaches [Zhang *et al.*, 2013] are also able to model the temporal relations among actions. Bayesian network models use the directed acyclic graph for both learning and inference, they hence face the problem of handling temporal relation conflicts among actions. Pattern-based approaches try to capture the complex temporal relatedness via temporal patterns and have demonstrated their advantages in handling the relatedness problem in the medical and finance domains [Patel *et al.*, 2008; Wu and Chen, 2007]. However, as far as we know, the literature on temporal pattern-based representations for sensor-based activity recognition is relatively sparse. Gu *et al.* [2009] proposed an Emerging Pattern (EP)-based approach for activity recognition. Their approach is able to handle sequential, interleaved and concurrent relations between pairwise actions. In contrast to their work, ours provides a more general way to describe temporal relations among more than two actions. Moreover, our method can capture the intrinsic relatedness among various activities, which can further enhance overall recognition performance.

Multi-task learning (MTL) is a learning paradigm that jointly learns multiple related tasks and can achieve better generalization performance than learning each task individually, especially with those insufficient training samples. The relations among tasks can be pairwise correlations [Zhang and Yeung, 2010], or pairwise correlation within a group [Zhou *et al.*, 2011a], as well as higher-order relationships [Zhang and Yeung, 2013]. However, for activity recognition, encoding only task relatedness is not enough. Since not all features are discriminative for the prediction tasks, it is reasonable to assume that only a small set of features is predictive for specific tasks. In the light of this, group Lasso [Yuan and Lin, 2006] is a technique used for selecting group variables that are key to the prediction tasks. As an important extension of Lasso [Tibshirani, 1996], group Lasso combines the feature strength over all tasks and tends to select

the features based on their overall strength. It ensures that all tasks share a common set of features, while each one keeps its own specific features [Zhou *et al.*, 2011b].

3 Temporal Pattern Mining

A training collection for activity recognition consists of multiple activities, and each activity is a sequence of actions with its corresponding start-time and end-time. We aim to mine the frequent temporal patterns from these sequences and use the patterns to represent activities for subsequent learning. We define some important concepts first.

Definition 3.1. Let Θ denote the action space. An *action* $act \in \Theta$ that occurs during a period of time is denoted as a triplet $a = (t_s, id_a, t_e)$, where id_a is the action id , t_s is the start-time, t_e is the end-time, and $t_s < t_e$. Each *activity* over Θ is defined as a sequence of actions ordered by start-time. Formally, $activity = \langle a^1, a^2, \dots, a^p \rangle$, with the constraints $a^i = (t_s^i, id_a^i, t_e^i)$, and $t_s^i \leq t_s^{i+1}$ for $1 \leq i \leq p-1$.

Definition 3.2. A *temporal pattern* P of dimension $k > 1$ is defined as a pair $(\mathcal{S}, \mathbf{R})$, where $\mathcal{S} \subseteq \Theta$ is a set of actions. \mathbf{R} is a $k \times k$ matrix, where $R_{i,j}$, its (i, j) -th element, indicates the temporal relatedness between $act_i \in \mathcal{S}$ and $act_j \in \mathcal{S}$. The temporal relation is encoded with Allen’s temporal interval logic [Allen, 1983]. The dimension of a temporal pattern P is written as $dim(P)$, which equals $|\mathcal{S}|$. If $dim(P) = k$, then temporal pattern P is called a *k-pattern*.

Definition 3.3. We denote the *partial order* over temporal patterns as \sqsubseteq and define it as follows: The temporal pattern $(\mathcal{S}_A, \mathbf{R}_A)$ is a subpattern of temporal pattern $(\mathcal{S}_B, \mathbf{R}_B)$ (or $(\mathcal{S}_A, \mathbf{R}_A) \sqsubseteq (\mathcal{S}_B, \mathbf{R}_B)$), if and only if it satisfies these two conditions: (a) $dim((\mathcal{S}_A, \mathbf{R}_A)) < dim((\mathcal{S}_B, \mathbf{R}_B))$; and (b) there exists an injective mapping $\pi : \{1, \dots, dim((\mathcal{S}_A, \mathbf{R}_A))\} \rightarrow \{1, \dots, dim(\mathcal{S}_B, \mathbf{R}_B)\}$ such that $\forall i, j \in \{1, \dots, dim((\mathcal{S}_A, \mathbf{R}_A))\} : \mathcal{S}_A(i) = \mathcal{S}_B(\pi(i)) \wedge \mathbf{R}_A[i, j] = \mathbf{R}_B[\pi(i), \pi(j)]$. In particular, the equality of actions in condition (b) depends only on the action id and is independent of the start and end times.

Definition 3.4. The *support* of a temporal pattern P is defined as $sup(P) = \frac{L_P}{L}$, where L_P refers to the total time that the pattern can be observed within a sliding window. $L = t_{win} + t_{activity}$ is a normalizing term, where t_{win} denotes the width of the sliding window and $t_{activity}$ refers to the total time of the activity [Höppner, 2001]. In practice, the window length can be set as the maximum or average length of actions in the dataset. We can interpret the support $sup(P)$ as the observation probability of pattern P within the given activity. This gives us the intuition of choosing this support for describing action patterns: the larger the support value for a given pattern is, the more correlated the pattern will be with a complex activity.

Definition 3.5. Denote the minimum support threshold as $minsup$. Then a pattern P is regarded as a *frequent pattern* if $sup(P) \geq minsup$.

Basically, the notion of frequent pattern is utilized to bridge the semantic gap between actions and activities. It captures the intrinsic descriptions of activities and hence provides a

natural way to informatively represent activities for further recognition. To discover the discriminant temporal patterns, we initially estimate the support value of each action (which is the simplest pattern, i.e., a 1-*pattern*). After that, we remove the 1-*patterns* with small support values. Based upon the remaining 1-*patterns*, we generate the 2-*patterns* and calculate their support values. Similarly, we prune the infrequent 2-*patterns*. By iterating this procedure k times, we ultimately obtain a set of patterns with dimensions up to k . The algorithm terminates when no more frequent patterns are found. The pruning process is effective due to the **Apriori Principle** and **Monotonicity Property** [Agrawal and Srikant, 1994]. Notedly, the support of a pattern is always less than or equal to the support of any of its sub-patterns. In other words,

$$\forall \text{patterns } P, Q : Q \subseteq P \Rightarrow \text{sup}(Q) \geq \text{sup}(P).$$

This property is guaranteed by the definition of partial order and the support value of temporal patterns [Höppner, 2001]. With the above property, our algorithm does not miss any frequent pattern. Moreover, this property ensures the correctness of our pruning step since all the sub-patterns of a frequent temporal pattern must also be frequent.

With the mined temporal patterns, we put all of them together to construct a joint pattern feature space. Thereby, each activity can be represented within this feature space, and each entry in the feature vector is the support value of the corresponding pattern. According to the definition, the support of a temporal pattern is conditioned on a specific activity, and the relevance of the support with respect to a specific activity is implied in the support estimation step. Thus, it is intuitive to describe activities in this way, since the higher the support value for a given pattern, the more relevant or important the pattern is for the given activity. It is worth mentioning that the feature space is generated from the training data only, and both the testing and training data share the same pattern feature space.

4 Adaptive Multi-Task Learning

Similar activities may share some patterns. For example, “playing badminton” shares many similar temporal patterns with “playing tennis”, but greatly differs from “fishing”. Moreover, the dimension of temporal pattern features is usually very high, but not all temporal patterns are sufficiently discriminative for activity recognition. To capture the relatedness among different activities and select the discriminative patterns simultaneously, we regard each activity recognition as a task and present an **aMTL** model. It is able to adaptively capture the relatedness among tasks, as well as learn the task-sharing and task-specific features.

4.1 Problem Formulation

We first define some notations. In particular, we use bold capital letters (e.g., \mathbf{X}) and bold lowercase letters (e.g., \mathbf{x}) to denote matrices and vectors, respectively. We employ non-bold letters (e.g., x) to represent scalars, and Greek letters (e.g., λ) as parameters. Unless stated, otherwise, all vectors are in column form. Assume that we have M kinds of activities/tasks $\{T_1, T_2, \dots, T_M\}$ in the given training set Φ . Φ

is composed of N samples $\{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \dots, (\mathbf{x}_N, \mathbf{y}_N)\}$. Each training sample is a sequence of actions, represented by temporal pattern-based feature vector $\mathbf{x}_i \in \mathbb{R}^D$ and their corresponding label vector $\mathbf{y}_i \in \mathbb{R}^M$, where \mathbf{y}_i is the label vector with a single one and all other entries zero. Each instance is only one activity, and D is the feature dimension. The prediction model for task T_i of a given sample is defined as $f_i(\mathbf{x}) = \mathbf{x}^T \mathbf{w}^i$, where $\mathbf{w}^i \in \mathbb{R}^D$ is the weight vector for task T_i . Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]^T \in \mathbb{R}^{N \times D}$ be the data matrix and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N]^T \in \mathbb{R}^{N \times M}$ be the label matrix. The weight matrix over M tasks is denoted as $\mathbf{W} = [\mathbf{w}^1, \mathbf{w}^2, \dots, \mathbf{w}^M] \in \mathbb{R}^{D \times M}$.

We formulate activity recognition as,

$$\begin{aligned} \min_{\mathbf{W}, \Omega} \quad & \frac{1}{2} \|\mathbf{X}\mathbf{W} - \mathbf{Y}\|_F^2 + \lambda \text{tr}(\mathbf{W}\Omega^{-1}\mathbf{W}^T) + \gamma \|\mathbf{W}\|_F^2 + \theta \|\mathbf{W}\|_{2,1}, \\ \text{s.t.} \quad & \Omega \succeq 0, \quad \text{tr}(\Omega) = 1, \end{aligned} \quad (1)$$

where the first term measures the empirical error; the second term adaptively encodes the relatedness among different tasks; the third term controls the generalization error; and the last one is the group Lasso penalty which helps to select the desired features automatically. λ, γ, θ are the regularization parameters, and $\Omega \in \mathbb{R}^{M \times M}$ is a positive semi-definite matrix that we aim to learn. The (i, j) -th entry in Ω represents the relations between task i and task j . As an improvement over a uniform or pre-defined relatedness model [Kato *et al.*, 2008], we adaptively learn the relatedness among tasks. The $\ell_{2,1}$ -norm of a matrix \mathbf{W} is defined as $\|\mathbf{W}\|_{2,1} = \sum_{i=1}^d \sqrt{\sum_{j=1}^M W_{ij}^2}$. In particular, $\ell_{2,1}$ -norm applies an ℓ_2 -norm to each row of \mathbf{W} and these ℓ_2 -norms are combined through an ℓ_1 -norm. Thus, the weights of one feature over M tasks are combined through ℓ_2 -norm and all features are further grouped via ℓ_1 -norm. The $\ell_{2,1}$ -norm thereby plays the role of selecting features based on their strength over all tasks. As we assume that only a small set of features are predictive for each recognition task, the group Lasso penalty ensures that all activities share a common set of features while still keeping their activity-specific features.

4.2 Optimization

Eqn.(1) is convex with respect to \mathbf{W} and Ω . We adopt the alternative optimization procedure to solve it.

Optimizing \mathbf{W} with Ω fixed. When Ω is fixed, the optimization problem in Eqn.(1) becomes an unconstrained convex optimization problem. The optimization problem can be rewritten as follows,

$$\min_{\mathbf{W}} \quad \frac{1}{2} \|\mathbf{X}\mathbf{W} - \mathbf{Y}\|_F^2 + \lambda \text{tr}(\mathbf{W}\Omega^{-1}\mathbf{W}^T) + \gamma \|\mathbf{W}\|_F^2 + \theta \|\mathbf{W}\|_{2,1}. \quad (2)$$

Since the objective function in Eqn.(2) is convex and non-smooth, we use the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [Beck and Teboulle, 2009] to solve it.

Optimizing Ω with \mathbf{W} fixed. When \mathbf{W} is fixed, the optimization problem in Eqn.(1) for solving Ω becomes

$$\begin{aligned} \min_{\Omega} \quad & \text{tr}(\Omega^{-1}\mathbf{W}^T\mathbf{W}), \\ \text{s.t.} \quad & \Omega \succeq 0, \quad \text{tr}(\Omega) = 1. \end{aligned} \quad (3)$$

Denote $\mathbf{A} = \mathbf{W}^T \mathbf{W}$

$$\begin{aligned} \text{tr}(\boldsymbol{\Omega}^{-1} \mathbf{A}) &= \text{tr}(\boldsymbol{\Omega}^{-1} \mathbf{A}) \text{tr}(\boldsymbol{\Omega}) \\ &= \text{tr}((\boldsymbol{\Omega}^{-\frac{1}{2}} \mathbf{A}^{\frac{1}{2}})(\mathbf{A}^{\frac{1}{2}} \boldsymbol{\Omega}^{-\frac{1}{2}})) \text{tr}(\boldsymbol{\Omega}^{\frac{1}{2}} \boldsymbol{\Omega}^{\frac{1}{2}}) \\ &\geq (\text{tr}(\boldsymbol{\Omega}^{-\frac{1}{2}} \mathbf{A}^{\frac{1}{2}} \boldsymbol{\Omega}^{\frac{1}{2}}))^2 = (\text{tr}(\mathbf{A}^{\frac{1}{2}}))^2. \end{aligned} \quad (4)$$

The equality holds if and only if $\boldsymbol{\Omega}^{-\frac{1}{2}} \mathbf{A}^{\frac{1}{2}} = a \boldsymbol{\Omega}^{\frac{1}{2}}$ for some constant a and $\text{tr}(\boldsymbol{\Omega}) = 1$. Therefore, there exists a closed-form solution for $\boldsymbol{\Omega}$, i.e.,

$$\boldsymbol{\Omega} = \frac{(\mathbf{W}^T \mathbf{W})^{\frac{1}{2}}}{\text{tr}((\mathbf{W}^T \mathbf{W})^{\frac{1}{2}})}. \quad (5)$$

5 Experiments

5.1 Dataset

The *Opportunity* dataset [Chavarriga *et al.*, 2013] contains human activities recorded in a room with kitchen, deckchair, and outdoors using 72 sensors on the body and objects. Four subjects were invited to perform five complex and high-level activities, namely, *relaxing (RL)*, *early morning (EM)*, *coffee time (CT)*, *sandwich time (ST)* and *cleanup (CU)* in this environment. Each subject is recorded with ADL runs and a drill run. The ADL runs consist of temporally unfolding actions without pre-defined rules on how to perform the tasks, and the drill run comprises of a pre-defined set of instructions on how to perform the tasks. We selected ADL runs to verify our model due to its realistic scenarios of human activities. The activity *relaxing* has 40 samples; for the other four activities, each has 20 samples. Each activity in our dataset is composed of 30 low-level actions that have been well-labeled with action *id*, start and end times. In particular, these actions include four body locomotions (*sit*, *stand*, *walk*, *lie*) and 13 actions for each of the left and right hands (e.g, *close*, *reach*, *open*, *move*). In this experiment, we utilized the 30 low-level actions as input and the five high-level activities as output. The performance reported in this paper was measured based on 10-fold cross-validation classification accuracy.

5.2 Performance of Temporal Patterns

To verify the representativeness of our temporal patterns, we compare the following approaches:

1. *bag-of-actions*: It represents activities as a multiset of actions, discarding the action order but keeping multiplicity.
2. *1-patterns*: Generated by our method.
3. *{1, 2}-patterns*: Combination of 1 and 2-*patterns*.
4. *{1, 2, 3}-patterns*: Combination of 1, 2 and 3-*patterns*.

The results for bag-of-actions are: **SVM** (80.0%), **kNN** (79.8%), **Lasso** (90.8%), **MTL** (91.6%), **GL** (91.5%), **aMTL** (**93.3%**). The results for temporal patterns are presented in Figure 1. From the above results and Figure 1, it can be seen that approaches based on *1-patterns* outperform those based on the bag-of-actions approach. This is because *1-patterns* take not only the action frequency into

Table 1: Pairwise significance test between different features on **aMTL**.

| Pairwise Significance Test | <i>p-values</i> |
|--|-----------------|
| <i>1-patterns</i> vs bag-of-actions | 9.3e-3 |
| <i>{1, 2}-patterns</i> vs <i>1-patterns</i> | 3.5e-3 |
| <i>{1, 2, 3}-patterns</i> vs <i>1-patterns</i> | 3.3e-6 |

account, but also how long the action appears in this activity. This is intuitive as the longer an action appears, the more important the action will be. In addition, we can observe that the higher-order temporal patterns, such as *{1, 2}-patterns* and *{1, 2, 3}-patterns* show superiority over others. This demonstrates that the temporal relatedness among the actions is able to enhance the description of activities.

In addition, we performed pairwise significance test among various representation approaches based on the same **aMTL** model. The results are summarized in Table 1. All the *p-values* are smaller than 0.05, which indicates that our proposed temporal pattern mining approach is significantly better than the bag-of-actions, and the improvements by higher patterns are statistically significant.

5.3 Learning Model Comparison

To validate our proposed **aMTL** model, we compared it with five baselines:

- **SVM**: We implemented this method with the help of LIBSVM². We selected a linear kernel.
- **kNN**: We employed the k-Nearest Neighbors in OpenCV³ and set $K = 7$.
- **Lasso**: **Lasso** [Tibshirani, 1996] tries to minimize the objective function $\frac{1}{2} \|\mathbf{X}\mathbf{W} - \mathbf{Y}\|_F^2 + \alpha \|\mathbf{W}\|_F^2 + \beta \|\mathbf{W}\|_1$ and encodes the sparsity over all weights in \mathbf{W} . It keeps task-specific features but ignores the task-sharing features.
- **MTL**: As a typical example of traditional multi-task learning, the work of Zhang and Yeung [2010] aims to capture the task relationship in multi-task learning. We can derive **MTL** from our model by setting $\theta = 0$.
- **GL**: The last baseline is group Lasso regularization method with a $\ell_{2,1}$ -norm penalty for group feature selection $\frac{1}{2} \|\mathbf{X}\mathbf{W} - \mathbf{Y}\|_F^2 + \mu \|\mathbf{W}\|_F^2 + \nu \|\mathbf{W}\|_{2,1}$ [Yuan and Lin, 2006]. This model encodes the group sparsity but fails to take task relatedness into account. We can derive **GL** from **aMTL** by setting $\lambda = 0$.

We retained the same parameter settings over all the experiments. For **aMTL**, we set $\text{minsup} = 0.01$ and $t_{win} = 2 \times L_{avg}$ over all the experiments, where L_{avg} is the average length of action intervals in an activity.

The experimental results are demonstrated in Figure 1. From this figure, it can be seen that **MTL** outperforms the single task learning methods, which verifies that there exists relatedness among these activities and such relatedness can

²<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

³<http://opencv.org/>

Table 2: Pairwise significance test between our proposed **aMTL** model and each of the baselines. For each pair, their 10-*fold* results were utilized to conduct the t-test.

| Pairwise Significance Test | aMTL vs SVM | aMTL vs kNN | aMTL vs Lasso | aMTL vs MTL | aMTL vs GL |
|----------------------------|--------------------|--------------------|----------------------|--------------------|-------------------|
| 1- <i>patterns</i> | 1.0e-2 | 1.0e-2 | 4.3e-4 | 2.3e-2 | 1.1e-3 |
| {1, 2}- <i>patterns</i> | 2.4e-2 | 9.9e-3 | 7.9e-7 | 2.2e-3 | 3.6e-5 |
| {1, 2, 3}- <i>patterns</i> | 3.3e-2 | 8.1e-3 | 1.0e-8 | 2.7e-4 | 2.7e-7 |

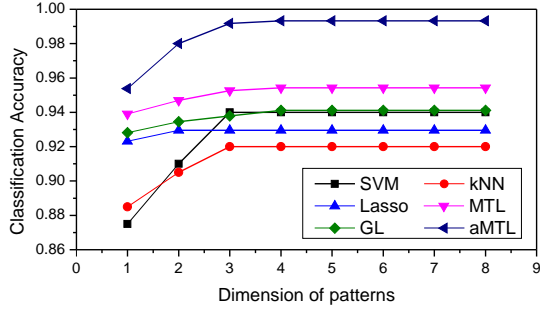


Figure 1: Comparative performance illustration of activity recognition with respect to various models on different pattern dimensions.

boost the learning performance. Moreover, as compared to the single task learning methods, **Lasso** achieves a bit higher accuracy due to the fact that the temporal pattern features for representing complex activities are quite sparse. However, as **Lasso** can only keep the task-specific features, **GL** shows a slightly better performance. This is because group sparsity is addressed during the learning, and the task-sharing features are also learned. This further justifies the assumption that only a small set of temporal patterns are predictive for activity recognition tasks. In addition, our **aMTL** model outperforms **MTL** by 2%-4% in overall accuracy. This is because our model encodes the group sparsity during learning and learns the activity-sharing and activity-specific temporal features simultaneously. Moreover, our method significantly outperforms **GL** with improvement of 3%-6%. This is to be expected since the **aMTL** model can capture the relatedness among activities and further improve performance. Moreover, the activity relatedness learned by **aMTL** is illustrated in Table 3. The matrix in Table 3 is encoded in matrix Ω automatically learned by our proposed **aMTL** model. It has been normalized, and its entries represent the pairwise similarities between activities. Larger value indicates more correlated relations between two activities. The diagonal elements are removed since they are self-correlated and less attractive. From Table 3, it can be seen that there exists different relatedness among activities. In particular, *coffee time* has higher correlations with *sandwich time* rather than *early morning*, and *relaxing* is more related to *early morning* rather than *cleanup*. This is consistent with our human perception, which, in turn, further verifies the assumption that there exists relatedness among activities and these relatedness can boost the performance.

We have also performed pairwise significance test between **aMTL** model and each of the baselines under various dimen-

Table 3: Activity relatedness learned by **aMTL**.

| | <i>RL</i> | <i>CT</i> | <i>EM</i> | <i>CU</i> | <i>ST</i> |
|-----------|-----------|-----------|-----------|-----------|-----------|
| <i>RL</i> | — | 0.101 | 0.219 | 0.102 | 0.106 |
| <i>CT</i> | 0.101 | — | 0.036 | 0.142 | 0.212 |
| <i>EM</i> | 0.219 | 0.036 | — | 0.117 | 0.008 |
| <i>CU</i> | 0.102 | 0.142 | 0.117 | — | 0.139 |
| <i>ST</i> | 0.106 | 0.212 | 0.008 | 0.139 | — |

sions of temporal patterns, and the results are shown in Table 2. It can be seen that all the *p-values* are smaller than 0.05, which demonstrates that our **aMTL** model is consistently and significantly better than the baselines across various temporal pattern features.

5.4 Overall Scheme Evaluation

To validate our proposed scheme (temporal patterns + **aMTL**) for activity recognition, we also compared it against three baselines: two dynamical model approaches (**HMM** [Rabiner, 1989] and **CRF** [Lafferty *et al.*, 2001]) and a state-of-the-arts activity recognition approach (**ITBN** [Zhang *et al.*, 2013]).

The results are displayed in Table 4. From Table 4, we have the following observations: 1) **HMM** performs much worse than other methods due to the following reasons. First, **HMM** has strong first-order Markovian assumption for the state sequences, which may not be adequate for activity sequences; Second, it does not consider the complicate pairwise temporal relationships between actions, which may contribute significantly to activity recognition. 2) As compared to **HMM**, **CRF** achieves better performance by modeling the pairwise action relations. 3) our approach (*{1, 2}-patterns* + **aMTL**) outperforms **CRF**, since the temporal patterns can capture more complex pairwise relations between actions (e.g., overlapping). This clearly reflects that temporal data contains rich temporal relatedness information than sequential data and this kind of information, in turn, can further boost the performance. Similarly, *{1, 2, 3}-patterns* + **aMTL** outperforms **CRF** due to its ability of capturing higher-order temporal relations among actions (i.e., *{3}-patterns*). And 4) our proposed method achieves higher performance than **ITBN**. Several reasons lead to this result. First of all, **ITBN** is a Bayesian network with directed acyclic graph and it will automatically remove samples that contain temporal relation conflicts. This process greatly reduces the training size and hence the training performance. Second, since **ITBN** tends to remove temporal relation conflicts, it will lose some kinds of temporal relatedness among actions which may affect its performance as well.

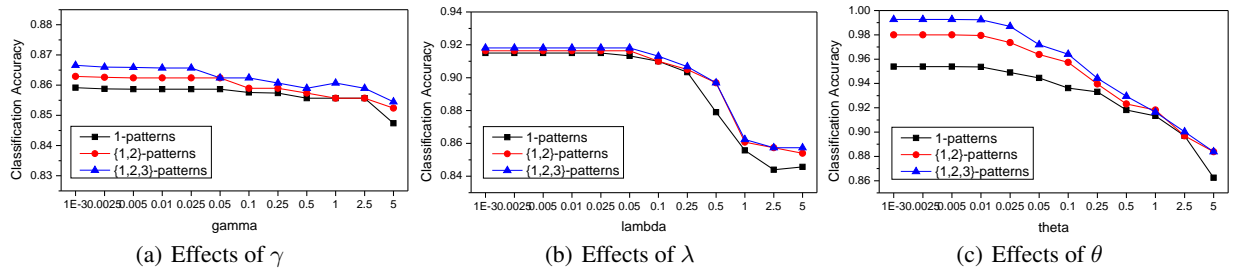


Figure 2: Performance illustration of activity recognition with respect to the sensitivity of **aMTL** parameters: γ , λ , θ .

Table 4: Performance comparison with state-of-the-art activity recognition approaches.

| Method | Accuracy % |
|----------------------------------|--------------|
| HMM | 54% |
| CRF | 95% |
| ITBN | 88% |
| {1, 2}-patterns + aMTL | 98.0% |
| {1, 2, 3}-patterns + aMTL | 99.2% |

5.5 Sensitivity of Parameters

Our algorithm involves several parameters, hence it is necessary to study the performance sensitivity over them.

We first investigated the performance of activity recognition over the pattern dimension. Figure 1 comparatively illustrates the experimental results of activity recognition with respect to various pattern dimensions. It can be seen that when k is larger than 3, the accuracy performance tends to be stable. The reason is that as k gets larger, fewer discriminative patterns are found, so the algorithm terminates automatically. Notably, larger patterns usually introduce high-dimensional features and lead to expensive time complexity for feature extraction. These experimental results reveal that $\{1, 2, 3\}$ -patterns are descriptive enough.

We then examined the effects of three key parameters involved in our **aMTL** model. They are γ , λ , θ , which respectively balance the trade-off among generalization error, activity-relatedness and group sparsity. We initially fixed λ and θ , and then varied γ from 0.001 to 5 and doubled the value at each step. The experimental results over different γ are shown in Figure 2(a). It can be seen that γ does not affect performance too much as it only increases a little when γ decreases and gives a slightly better result when $\gamma = 0.001$. We then set $\gamma = 0.001$, $\theta = 1$ and varied λ . The results are illustrated in Figure 2(b). It can be seen that the performance increases as λ decreases but stabilizes at 0.05. Finally, we set $\gamma = 0.001$, $\lambda = 0.05$ and varied θ . We can see that small θ leads to better performance, as illustrated in Figure 2(c). However, it remains steady when θ equals or is less than 0.01.

5.6 Discussion

In this section, we discuss the efficiency and scalability of our proposed scheme. The efficiency of temporal pattern mining algorithm is guaranteed by its convergence at $\{1, 2, 3\}$ -patterns as shown in Figure 1, i.e., we can stop at $\{3\}$ -

patterns. On the other hand, the computational cost of **aMTL** is not expensive. In particular, for the optimization of Eqn.(2), the complexity for each iteration in the FISTA algorithm is $O((N + M)DM)$. Moreover, the FISTA algorithm converges within $O(1/\epsilon^2)$ iterations, where ϵ is the desired accuracy. For the optimization of Eqn.(3), calculating the closed-form solution in Eqn.(5) scales up to $O(DM^2 + M^3)$. Therefore, the total time cost for one iteration in the alternating algorithm is $O(\frac{(N+M)DM}{\epsilon^2} + DM^2 + M^3)$. From the experiments, we find that the alternating algorithm only needs very few iterations to converge (usually within 20 iterations), making the whole procedure very efficient.

Our proposed scheme is also scalable to big data and other activity recognition. The scalability of temporal pattern mining algorithm is ensured by its pruning step, since it tends to select only frequent temporal patterns from the activities, which makes the algorithm scalable to large datasets. For the **aMTL** algorithm, according to its time complexity analysis, it is linearly dependent to the number of features D and has a relation $O(M^3)$ with the number of activities. In most of the real applications, we only focus on a small number of important activities. Therefore, our proposed scheme can easily scale to large datasets with high-dimensional features. In addition, the relatedness among different activities are automatically learned from given dataset. Hence our scheme is generalizable to other activity recognition.

6 Conclusion and Future Work

This paper presents a scheme to recognize activities from sensor data. It comprises of two components. The first component is temporal pattern mining. It works towards mining frequent patterns from low-level actions. The second one trains an adaptive multi-task learning model to capture the relatedness among activities. Extensive experiments on real-world data show significant gains of these two components and their overall performance as compared to state-of-the-arts methods.

In future, we plan to extend our work to deal with error propagation problems in the output of low-level action recognition systems.

Acknowledgments

This work was supported in part by grants R-252-000-473-133 and R-252-000-473-750 from the National University of Singapore.

References

- [Agrawal and Srikant, 1994] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the International Conference on Very Large Data Bases*, 1994.
- [Allen, 1983] James F Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 1983.
- [Beck and Teboulle, 2009] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *Society for Industrial and Applied Mathematics Journal on Imaging Sciences*, 2009.
- [Chavarriaga et al., 2013] Ricardo Chavarriaga, Hesam Sagha, Alberto Calatroni, Sundara Tejaswi Digumarti, Gerhard Tröster, José del R Millán, and Daniel Roggen. The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters*, 2013.
- [Gu et al., 2009] Tao Gu, Zhanqing Wu, Xianping Tao, Hung Keng Pung, and Jian Lu. epsicar: An emerging patterns based approach to sequential, interleaved and concurrent activity recognition. In *Proceedings of IEEE International Conference on Pervasive Computing and Communications*, 2009.
- [Höppner, 2001] Frank Höppner. Discovery of temporal patterns. In *Principles of Data Mining and Knowledge Discovery*. 2001.
- [Kato et al., 2008] Tsuyoshi Kato, Hisashi Kashima, Masashi Sugiyama, and Kiyoshi Asai. Multi-task learning via conic programming. In *Advances in Neural Information Processing Systems*, 2008.
- [Lafferty et al., 2001] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.
- [Nie et al., 2015] Liqiang Nie, Yi-Liang Zhao, M. Akbari, Jialie Shen, and Tat-Seng Chua. Bridging the vocabulary gap between health seekers and healthcare knowledge. *IEEE Transactions on Knowledge and Data Engineering*, 2015.
- [Pansiot et al., 2007] Julien Pansiot, Danail Stoyanov, Douglas McIlwraith, Benny P.L. Lo, and G.Z. Yang. Ambient and wearable sensor fusion for activity recognition in healthcare monitoring systems. In *Proceedings of International Workshop on Wearable and Implantable Body Sensor Networks*. 2007.
- [Patel et al., 2008] Dhaval Patel, Wynne Hsu, and Mong Li Lee. Mining relationships among interval-based events for classification. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2008.
- [Rabiner, 1989] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 1989.
- [Ravi et al., 2005] Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, and Michael L. Littman. Activity recognition from accelerometer data. In *Proceedings of the Conference on Innovative Applications of Artificial Intelligence*, 2005.
- [Ryoo and Aggarwal, 2006] Michael S Ryoo and Jake K Aggarwal. Recognition of composite human activities through context-free grammar based representation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2006.
- [Tibshirani, 1996] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, 1996.
- [Wang et al., 2012] Xinxin Wang, David Rosenblum, and Ye Wang. Context-aware mobile music recommendation for daily activities. In *Proceedings of the ACM International Conference on Multimedia*, 2012.
- [Wu and Chen, 2007] Shin-Yi Wu and Yen-Liang Chen. Mining nonambiguous temporal patterns for interval-based events. *IEEE Transactions on Knowledge and Data Engineering*, 2007.
- [Yang, 2009] Qiang Yang. Activity recognition: Linking low-level sensors to high-level intelligence. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2009.
- [Yuan and Lin, 2006] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B*, 2006.
- [Zhang and Yeung, 2010] Yu Zhang and Dit-Yan Yeung. A convex formulation for learning task relationships in multi-task learning. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2010.
- [Zhang and Yeung, 2013] Yu Zhang and Dit-Yan Yeung. Learning high-order task relationships in multi-task learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2013.
- [Zhang et al., 2013] Yongmian Zhang, Yifan Zhang, Eran Swears, Natalia Larios, Ziheng Wang, and Qiang Ji. Modeling temporal interactions with interval temporal bayesian networks for complex activity recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.
- [Zhou et al., 2011a] Jiayu Zhou, Jianhui Chen, and Jieping Ye. Clustered multi-task learning via alternating structure optimization. In *Advances in Neural Information Processing Systems*, 2011.
- [Zhou et al., 2011b] Jiayu Zhou, Lei Yuan, Jun Liu, and Jieping Ye. A multi-task learning formulation for predicting disease progression. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2011.
- [Zhuo et al., 2009] Hankz Hankui Zhuo, Derek Hao Hu, Chad Hogg, Qiang Yang, and Hector Munoz-Avila. Learning htn method preconditions and action models from partial observations. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2009.