21st International Conference on Pattern Recognition (ICPR 2012)
November 11-15, 2012. Tsukuba, Japan

Sparse Representation for Motion Primitive-Based Human Activity Modeling and Recognition Using Wearable Sensors

Mi Zhang[1], Wenyao Xu[2], Alexander A. Sawchuk[1], and Majid Sarrafzadeh[2]
[1]*Signal and Image Processing Institute, University of Southern California, USA*
[2]*Wireless Health Institute, University of California, Los Angeles, USA*
mizhang@usc.edu, xuwenyao@ucla.edu, sawchuk@sipi.usc.edu, majid@cs.ucla.edu

Abstract

The use of wearable sensors for human activity monitoring and recognition is becoming an important technology due to its potential benefits to our daily lives. In this paper, we present a sparse representation-based human activity modeling and recognition approach using wearable motion sensors. Our approach first learns an overcomplete dictionary to find the motion primitives shared by all activity classes. Activity models are then built on top of these motion primitives by solving a sparse optimization problem. Experiments on a dataset including nine activities and fourteen subjects show the advantages of using sparse representation for activity modeling and demonstrate that our approach achieves a better recognition performance compared to the conventional motion primitive-based approach.

1 Introduction

The recognition of various human activities has been a research focus in computer vision for decades. However, the major concern of this vision-based solution is that it would fail if people are out of cameras' field of view. In recent years, using wearable sensors to track human activities becomes popular since it opens the door to applications such as fitness monitoring, physical rehabilitation and assisted living for elderly people that would provide enormous benefits to our lives [3]. In this paper, we focus on developing algorithms to recognize human activities using wearable motion sensors.

Human activity recognition using wearable motion sensors is challenging because of the complexity of human physical body kinematics. Over the years, many types of activity models have been explored. One standard method uses a "global" model that maps each activity segment to a single point in the feature space and trains a classifier to find the appropriate classification boundaries [2]. Another popular method uses the temporal-spatial activity trajectories constructed from either the raw sensor data [5] or the embedded nonlinear low-dimensional manifolds [7] to characterize each activity. Recognition is then casted as a trajectory matching problem. However, these models are limited by their robustness to outliers and their scalability in handling large number of activity classes and activity style variations among human subjects.

Recently, a new modeling method which uses motion primitive to capture the local information of human activity signals has shown promising performance to tackle the issues mentioned above [8]. In this motion primitive-based model, each activity segment is first partitioned into a sequence of tiny window cells. Then an overcomplete dictionary is learned through unsupervised clustering techniques (such as K-means) from a set of training samples. Each dictionary element is referred to as a motion primitive shared by all the activity classes. Finally, the partitioned window cells are mapped to the motion primitives and activity models are built in the primitive space via primitive distribution.

The performance of the motion primitive-based model relies on the quality of the dictionary. In this work, we are inspired by the recent success of the sparse coding theory and propose a method which follows the motion primitive-based model with the goal of further improving its performance by learning a dictionary and building activity models based on sparse representation-based techniques. Specifically, we first apply the K-SVD algorithm proposed in [1], which learns an overcomplete dictionary via an optimization problem with sparsity constraints. It is a direct generalization of the K-means algorithm such that the dictionary learned by K-SVD has more representation power and better fits the training data. Activity models are then built based on the accumulated sparse coefficients related to the dictionary elements. These coefficients can be seen as a natural extension to the primitive distribution used in the motion primitive-based model. Here we study the

978-4-9906441-0-9

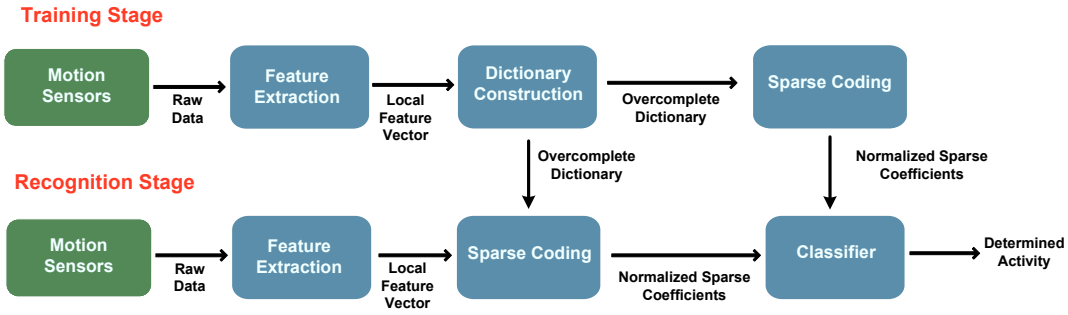

Figure 1. The block diagram of the sparse representation-based motion primitive framework

robustness of this method with respect to different window cell sizes, dictionary sizes and sparsity constraints, and compare its performance to the baseline motion primitive-based model based on K-means.

2 Our Approach

Figure 1 shows an overview of our framework. In training stage, the streaming sensor data sampled from activity segments is first divided into a sequence of fixed-length tiny window cells whose length is much smaller than the duration of the activity segment itself. Features are extracted from each window cell and stacked together to form a local feature vector. The local feature vectors from all training activity segments are then pooled together to learn the overcomplete dictionary. By incorporating sparse coding, activity models are built and represented through sparse coefficients related to the dictionary elements. Finally, these coefficients are used as global features to train the classifier. In recognition stage, the test activity segment is first transformed into a sequence of local feature vectors in the same manner as in training stage. Its sparse coefficients related to the dictionary elements are then computed and imported into the classifier for classification. We now present the details of each component.

2.1 Sensing Platform and Feature Extraction

We use a MotionNode[1] wearable sensor that integrates a 3-axis accelerometer ($\pm$6g) and a 3-axis gyroscope ($\pm$500dps) to collect human activity signals. The sampling rate is set to 100 Hz which is high enough to capture all details of normal human activities. For each axis of both accelerometer and gyroscope, we extract five features including mean, standard deviation, root mean square, derivative, and mean crossing rate. These features have proven to be useful to capture activity characteristics in many previous studies [2]

[1] http://www.motionnode.com/

2.2 Dictionary Learning

In this work, we employ the K-SVD algorithm proposed in [1] to learn the overcomplete dictionary from the training data. Specifically, assume that there are L distinct activity classes to classify and n_c training window cells from class c, $c \in [1, 2, \ldots L]$. Recall that each window cell is represented as a m-dimensional local feature vector (m is equal to 30 in our case). To learn the dictionary, we first pool the local feature vectors from all the activity classes together and arrange them as columns to construct the data matrix:

$$\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N] \in R^{m \times N} \tag{1}$$

where $N = n_1 + n_2 + \ldots + n_L$ denotes the total number of training windows cell samples. Given $\boldsymbol{Y}$, the K-SVD algorithm intends to learn a reconstructive overcomplete dictionary $\boldsymbol{D} = [\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_K] \in R^{m \times K}$ with K elements, over which each $\boldsymbol{y}_i$ in $\boldsymbol{Y}$ can be sparsely represented as a linear combination of no more than T_0 dictionary elements. This can be formulated as an optimization problem which constructs the desired dictionary by minimizing the reconstructive error while satisfying the sparsity constraints:

$$\arg\min_{\boldsymbol{D},\boldsymbol{X}} \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_2^2 \quad \text{s.t. } \forall i, \|\boldsymbol{x_i}\|_0 \leq T_0 \tag{2}$$

where $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N] \in R^{K \times N}$ are the sparse coefficients of the data matrix $\boldsymbol{Y}$ related to $\boldsymbol{D}$, $\|\boldsymbol{x_i}\|_0$ is the ℓ^0 norm of the coefficient vector $\boldsymbol{x_i}$, which is equivalent to the number of non-zero components in the vector, and the term $\|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_2^2$ represents the reconstruction error of $\boldsymbol{Y}$ over $\boldsymbol{D}$ in terms of ℓ^2 norm.

Here, it is worthwhile to note the connection and the difference between the K-SVD algorithm and the K-means algorithm in the baseline motion primitive-based model for the task of dictionary learning. By using K-means, each local feature vector $\boldsymbol{y_i}$ is represented by the dictionary element which has the minimum ℓ^2 norm distance to it. Moreover, the coefficient multiplying the

closest dictionary element is forced to be integer one. In comparison, the K-SVD algorithm is designed to look for a more general solution, in which each local feature vector $\boldsymbol{y_i}$ is represented as a linear combination of as many as T_0 dictionary elements. In addition, the corresponding coefficients can be any real numbers. Therefore, the K-SVD algorithm can be regarded as a generalization of the K-means algorithm. As a consequence, the dictionary learned by K-SVD is expected to have more representation power of the data matrix $\boldsymbol{Y}$.

2.3 Sparse Coding for Activity Modeling

Given the overcomplete dictionary $\boldsymbol{D}$ learned in the previous step, any window cell $\boldsymbol{y}$ can be decomposed as a linear combination of the dictionary elements. Its coefficients $\boldsymbol{x}$ can be computed by solving the following standard sparse coding problem:

$$\arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_0 \quad \text{s.t. } \|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{x}\|_2 \leq \epsilon \tag{3}$$

where ϵ is the noise level. Finding the exact solution to (3) proves to be an NP hard problem [1]. However, if the signal is sparse enough, approximate solutions can be found by pursuit algorithms such as the matching pursuit (MP) [4] and orthogonal matching pursuit (OMP) [6]. Based on our experiments, OMP achieves better performance than MP. Therefore the results reported here are based on the OMP method.

Since each activity segment consists of a sequence of window cells, to build the activity model, we need to find a meaningful way to accumulate information from all the window cells within each segment. Assume there are M window cells in each activity segment. As mentioned above, each window cell is represented as a linear combination of dictionary elements $[\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_K]$ with the corresponding coefficients $[x_{\boldsymbol{d}_1}, x_{\boldsymbol{d}_2}, \ldots, x_{\boldsymbol{d}_K}]$. These coefficients can be viewed as the weights of the dictionary elements for reconstructing the window cells. Therefore, if we aggregate the coefficients from all window cells in each activity segment together, we will obtain a class-related distribution of the dictionary elements for each activity segment. After normalization, the distribution is transformed into the conditional probability defined as:

$$P(\boldsymbol{d}_i|c) = \frac{\sum_j x_{\boldsymbol{d}_i,j}}{\sum_i \sum_j x_{\boldsymbol{d}_i,j}} \tag{4}$$

where $i \in [1, 2, \ldots, K], j \in [1, 2, \ldots, M], c \in [1, 2, \ldots, L]$, and $P(\boldsymbol{d}_i|c)$ represents the probability of observing the dictionary element $\boldsymbol{d}_i$ given activity class c. Since these conditional probabilities capture the global information of the activity segments, we use them as features by concatenating them together as a K-dimensional global feature vector $\boldsymbol{f} = [P(\boldsymbol{d}_1|c), P(\boldsymbol{d}_2|c), \ldots, P(\boldsymbol{d}_K|c)]^T$ for classification.

2.4 Classifier

The size of the overcomplete dictionary can be potentially large. In this work, we use the multi-class Support Vector Machine (SVM) with linear kernel as our classifier. This classifier has proved to be very effective in handling high dimensional data in a wide range of pattern recognition applications.

3 Evaluation

3.1 Dataset and Experiment Setup

To evaluate our approach, fourteen participants with diverse gender, age, height and weight were asked to perform nine common activities from daily life: *walk forward*, *walk left*, *walk right*, *go upstairs*, *go downstairs*, *jump*, *run*, *stand*, and *sit*. A MotionNode is attached onto the participant's right front hip during data collection. To capture day-to-day variations and minimize inter-individual correlation, each participant performs five trials for each activity on different days without supervision. The captured activity signals in each trial are then segmented into 4 second activity segments.

To achieve reliable results, we adopt a leave-one-trial-out cross validation strategy. Specifically, since each participant performs five trials for each activity, we use four trials of all participants for dictionary learning and activity model training while the left-out trial is for testing. This process iterates for every trial, and the final result is the average value across all five trials.

3.2 Experiment Results and Discussions

As our first experiment, Figure 2 shows the recognition accuracy of our approach with different window cell sizes from 0.1 to 2 seconds. As shown, the accuracy reaches the maximum when the window cell size is 0.3 second. As the window cell size increases, the recognition accuracy declines as a general trend. This is partially attributed to the fact that as the window cell size increases, the local feature vector constructed from each window cell can not capture the local information of the activity signal anymore. Moreover, the increase of window cell size leads to the reduction of the total number of window cells included in each activity segment. Thus the statistical power of our activity model that is built on top of the primitive distribution is diluted.

Next, we fix the window cell size to 0.3 second and examine the impact of sparsity (T_0) on the classification performance. As shown in Figure 3, when T_0 is less

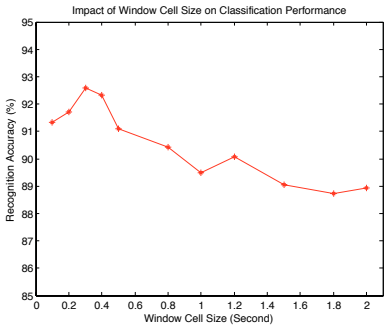

Figure 2. Impact of Window Cell Sizes

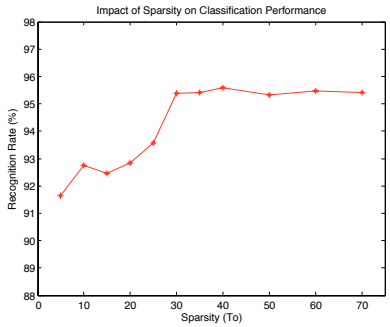

Figure 3. Impact of Sparsity (T_0)

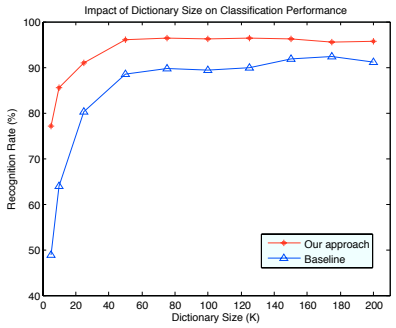

Figure 4. Impact of Dictionary Sizes (K)

than 30, the recognition accuracy rises in general as T_0 increases. This observation demonstrates the superiority of K-SVD over K-means for the task of dictionary learning. In other words, it shows significant benefits in using more than one dictionary element with real valued coefficients to represent window cells within each activity segment. It should also be noted that when T_0 is bigger than 30, the recognition accuracy only varies slightly. This indicates that using 30 elements in the dictionary is sufficient to reconstruct any window cell for our activity dataset.

Finally, we set T_0 to 30 and measure the performance using different dictionary sizes (K). We also compare our approach with the baseline motion primitive-based model under the same condition. The baseline algorithm uses K-means for dictionary learning and the raw primitive distribution (histogram of motion primitives) for activity modeling. As shown in Figure 4, the accuracy of our approach starts to rise at the very beginning and stabilizes when the dictionary size reaches 50. A maximum accuracy of 96.47% is achieved when the dictionary size is 75. More importantly, our approach achieves a much better performance compared to the baseline algorithm across all dictionary sizes, with an improvement of 10% for the same ditionary size on average. This result indicates that by leveraging sparse coding techniques, the motion primitive-based model can achieve a significant performance improvement.

4 Conclusion and Future work

In this paper, we presented a sparse representation-based approach for motion primitive learning and human activity recognition using wearable motion sensors. To conclude, our approach exhibits great robustness with a wide range of sparsity constraints and dictionary sizes. Furthermore, our approach achieves an average 10% performance improvement compared to the baseline motion primitive-based method. As shown, our approach separates dictionary learning and classifier training into two different steps. For future work, we plan to explore the possibility of jointly learning the dictionary and the classifier in one single step.

References

[1] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, November 2006.

[2] L. Bao and S. S. Intille. Activity recognition from user-annotated acceleration data. In *International Conference on Pervasive Computing*, pages 1–17, 2004.

[3] T. Choudhury and et al. The Mobile Sensing Platform: An Embedded Activity Recognition System. *IEEE Pervasive Computing*, 7(2):32–41, April 2008.

[4] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, December 1993.

[5] T. Stiefmeier and et al. Gestures are strings: efficient online gesture spotting and classification using string matching. In *BodyNets*, pages 1–8, Florence, Italy, 2007.

[6] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. Inf. Theory*, 53(12):4655–4666, 2007.

[7] M. Zhang and et al. Manifold learning and recognition of human activity using body-area sensors. In *ICMLA*, pages 7–13, Hawaii, USA, December 2011.

[8] M. Zhang and et al. Motion primitive-based human activity recognition using a bag-of-features approach. In *IHI*, pages 631–640, Florida, USA, January 2012.