Unsupervised Embedding Learning for Human Activity Recognition Using Wearable Sensor Data

Taoran Sheng, Manfred Huber

Department of Computer Science and Engineering
University of Texas at Arlington
taoran.sheng@mavs.uta.edu, huber@cse.uta.edu

Abstract

The embedded sensors in widely used smartphones and other wearable devices make the data of human activities more accessible. However, recognizing different human activities from the wearable sensor data remains a challenging research problem in ubiquitous computing. One of the reasons is that the majority of the acquired data has no labels. In this paper, we present an unsupervised approach, which is based on the nature of human activity, to project the human activities into an embedding space in which similar activities will be located closely together. Using this, subsequent clustering algorithms can benefit from the embeddings, forming behavior clusters that represent the distinct activities performed by a person. Results of experiments on three labeled benchmark datasets demonstrate the effectiveness of the framework and show that our approach can help the clustering algorithm achieve improved performance in identifying and categorizing the underlying human activities compared to unsupervised techniques applied directly to the original data set.

Introduction

The typical process of sensor-based human activity recognition (HAR), as shown in Figure1, consists of three important stages: data segmentation, feature extraction, and recognizing the type of the activity. Extensive studies have been conducted in all the stages of the HAR process. However, existing HAR methods rely heavily on labeled data to supervise the model training and to perform the recognition, thus a huge challenge for HAR systems is collecting annotated data. In the meantime, more and more wearable devices, smartphones, smart watches, etc., are used in people's daily lives. These wearable devices are usually equipped with various sensors, such as accelerometers, gyroscope, GPS sensors, etc., which can provide a massive amount of unlabeled sensor activity data. Due to the above mentioned facts, most of the existing HAR systems can not take advantage of the accessible unlabeled data efficiently. Therefore, we aim at developing an unsupervised method to leverage the unlabelled data to recognize the physical activities without requiring labels.

The Autoencoder (AE)is an unsupervised learning framework to find efficient encodings of data. It encodes important features of the inputs into a hidden representation, and then reconstructs the inputs based on their hidden representations. It is applied for dimension reduction, deep hierarchical model pre-training, etc. Because of its simplicity and efficiency, we also design our model based on the AE architecture. Yet only using reconstruction to guide the learning can lead the AE to encode a significant amount of unnecessary information, such as task-irrelevant information, or even destructive information, such as noise.

Motivated by this observation, our approach utilizes the intrinsic properties of the sensor activity data to project the data into a clustering-friendly embedding space. Two fundamental observations contribute to the design and formation of this space.

Firstly, slow feature analysis (Wiskott and Sejnowski 2002) finds that many properties in the real world changes lowly over time. Physical objects have inertia and their states usually change gradually and infrequently. This rule also applies to human activity. In most situations, the period of a person performing an activity will take a a relatively significant amount of time. It is rare that a person will switch between different activities very frequently. On the other hand, while during the course of an activity the type of the activity remains the same, the body pose of the activity varies over time. Hence, we include the temporal coherence property in the loss function, which allows the model to learn the essential features of the activity and ignore the irrelevant temporal details in the body pose.

Secondly, distinct activity and person characteristics are commonly co-present in the sensor data. For the HAR task, only the activity characteristics are relevant. For example, two persons can walk in two different styles, but people can still identify they are performing the same type of activity: walking, because the activity-relevant characteristics are used in the recognition process and the irrelevant person characteristics are disregard. Based on this property, another local neighborhood based objective function, which aims at removing irrelevant personal or individual details in the data, is used to guide the learning of the model.
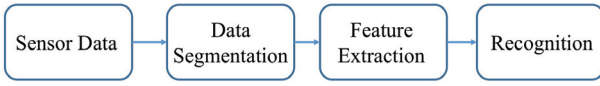
Figure 1: The typical process of HAR.

Figure 2: The overall architecture of the approach.

Related Work

Many works have been proposed to recognize human activity with wearable sensor data. As illustrated in Figure. 1, the first step in HAR is generally to segment the sensor data sequence. One common method used to do this is to use a sliding window. We also adopted this method in our model for its simplicity and tractability. Feature extraction and recognition are conducted on the segmented data. To recognize the activity type, discriminative features are needed. They can be designed with domain knowledge or extracted automatically using, for example, neural networks (NN).

Handcrafted features, when designed properly, have proven to be very useful in HAR systems. In (Kwapisz, Weiss, and Moore 2011), statistical features are derived from the time series sensor data. In (He and Jin 2009), discrete cosine transform is used to convert the sensor signal from the time domain into the transformed domain. Then features derived from the transformed domain are applied in the recognition process.

With the development of more competent deep learning techniques, and in particular NNs, automatic feature extraction has become another effective way to obtain discriminative features. In (Abu Alsheikh et al. 2016), a deep belief network was used as the emission matrix of a hidden Markov model. In (Morales and Roggen 2016), convolutional neural networks are employed to extract the features and recognize the type of the activity.

The features derived by these methods are then applied in a subsequent supervised model learning phase because they are usually not sufficient for direct use in an unsupervised model. There are only a few works that are recognizing the activity type in an unsupervised manner. In (Lu et al. 2017), a protein interaction method was used to cluster the activity data. In (Kwon, Kang, and Bae 2014), a set of time-frequency domain features are adopted, and unsupervised methods, in particular DBSCAN and mixture of Gaussian are used to cluster five basic activities.

Approach

Our approach differs from other works by using the aforementioned properties with regard to the nature of the activities. Specifically, our approach attempts to leverage two types of relationships: the temporal coherence of time series data, and locality preservation in the feature space.

Architecture

As shown in Figure 2, the foundation of the overall architecture of our approach is an AE framework. It consists of two parts: an encoder, and a decoder. The encoder defines the transformation: $E(\cdot)$, which transforms the input data sample x_i to the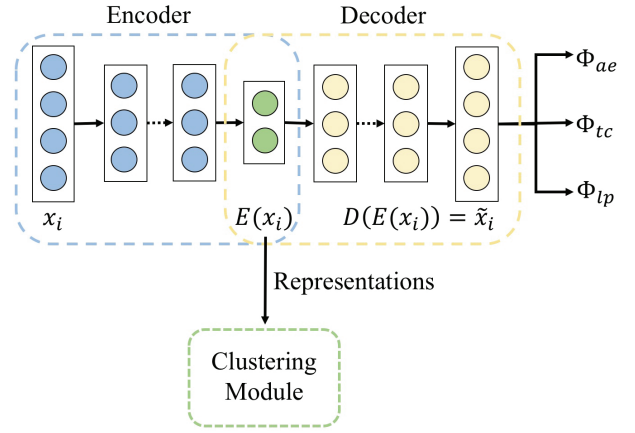 representation $E(x_i)$; and the decoder defines another transformation: $D(\cdot)$, which attempts to reconstruct the original input x_i based on the its representation $E(x_i)$:

$$\tilde{x}_i = D(E(x_i))$$

where, $\tilde{x}_i$ is the reconstructed input. In the traditional AE, the loss function of one data sample can be defined as:

$$\Phi_{ae}(x_i) = ||x_i - \tilde{x}_i||^2$$

where $||\cdot||$ denotes the Euclidean distance. This loss function forces the reconstruction $\tilde{x}_i$ to be as similar as possible to the original input x_i. If a good reconstruction $\tilde{x}_i$ can be decoded from the representation $E(x_i)$, it means that the representation $E(x_i)$ has retained much of the information that is important in the input x_i, so that the reconstruction $\tilde{x}_i$ can be very similar to the original input x_i. Thus the representation $E(x_i)$ can be used in other tasks, such as classification or clustering.

However, merely retaining information for reconstruction is usually not enough. The aim of the traditional AE is to learn a representation $E(x_i)$ that contains sufficient information to reconstruct the input, so an exact reconstruction also means to reconstruct noise and all the details in the input data. But not all the information in the learned representation is relevant to the subsequent task (e.g. noise as well as some task-irrelevant details might not only be unnecessary but could even be detrimental, especially in the context of subsequent clustering). Therefore, more task-oriented loss functions are imposed in our approach to guide the learning of the AE and make the learned representations more useful in the subsequent clustering task.

Temporal Coherence

Intuitively, a human activity can be decomposed into two components, a temporally varying component and a temporally stationary component. Specifically, certain dynamic properties of a single activity can vary over the time. For example, while walking the body pose varies over time: left foot and right foot alternatively step forward. This type of dynamic property is recorded in the sensor data, too, and we refer to it here as the temporally varying component.

On the other hand, no matter how the body pose varies over time, the semantic content of the activity remains the same. Namely, left foot and right foot can step forward alternatively, but the type of the activity is still walking. We refer to this part as the temporally stationary component.

Based on this nature of the human activity, the temporal coherence loss forces temporally close data samples to be similar to one another, and ignore the difference in the temporal varying component. It is motivated by the intention that the semantic content, i.e. the type of the activity, in which we are interested, should vary relatively infrequently over time. If the data samples are temporally close to each other, they may represent the same type of activity, even as they may be very distant in terms of the Euclidean distance in the sensor data space. The temporal coherence loss preserves the temporal continuity of the sensor data.

More formally, let x_i^t denote a data sample i, which occurs at time t during the course of an activity. Let M_i^t denote the index set of m temporal neighbors, x_j, of x_i^t. Then the temporal coherence loss Φ_{tc} for x_i^t is defined as:

$$\Phi_{tc}(x_i^t) = \frac{1}{m} \sum_{j \in M_i^t} ||x_j - \tilde{x_i^t}||^2$$

The temporal coherence loss encourages the reconstruction $\tilde{x_i^t}$ to be similar to its temporal neighbors so that the encoder can extract useful features from the temporally stationary component and ignore irrelevant time varying details.

Locality Preservation

Locality preservation is inspired by the observation that different persons perform the same type of activity in different fashions, but different fashions don't hinder other people to identify the activity type. Hence we assume that the personal or individual features in the activity data may not be necessary in the activity clustering stage, and the featureswhich are commonly present across multiple data pointsmay be the essential features of the activity. The locality preserving loss function is based on this assumption.

In past research works, the combination of carefully designed handcrafted high level features to represent the main characteristics of a temporally varying signal value, and of the k-Nearest Neighbor algorithm proved to be a powerful method to classify the sensor data of human activities. Due to its effectiveness and simplicity, it is employed in this approach to define the local neighborhood of a data sample. The locality preserving loss then aims to preserve the high level featurecharacteristics that are generally present in the local neighborhood.

The locality preserving loss forces the decoder to decode a data sample by using the learned representation of its nearby data samples. The rationale here is thatif the data samples are close to each other in the handcrafted feature space, they may represent the same type of activity. Thus the features shared across multiple nearby data samples should be the essential features of that type of activity. If the features do not exist in all the nearby data samples, then the features may represent personal or individual features, but not activity features.

Formally, let x_i^f denote data sample i in the feature space, and $\tilde{x_i^f}$ the reconstruction of x_i^f. Let N_i^f denote the index set of n local neighbors, x_k, of x_i^f in the handcrafted feature space. Then the locality preserving loss Φ_{lp} for x_i^f is defined as:

$$\Phi_{lp}(x_i^f) = \frac{1}{n} \sum_{k \in N_i^f} ||x_k - \tilde{x_i^f}||^2$$

The locality preserving loss forces the model to recover data sample x_k with the representation of its nearby point x_i^f. It drives the encoder to encode the information that has generally occurred across the neighborhood and to disregard individual features of single samples.

Joint Loss Function

The joint loss function is the sum of the temporal coherence loss and the locality preserving loss. It is used to train the model and is defined as follows:

$$min \sum_{i=1}^{S} (1 - \alpha - \beta)\Phi_{ae}(x_i) + \alpha\Phi_{tc}(x_i^t) + \beta\Phi_{lp}(x_i^f)$$

where i is the index of the sample, S is the size of the dataset, α and β are the parameters to balance the contribution of Φ_{ae}, Φ_{tc}, and Φ_{lp}. While Φ_{tc} and Φ_{lp} preserve more task relevant information in the representation, the Φ_{ae} component is also necessary in the learning processbecause without the reconstruction loss Φ_{ae}, the risk of learning trivial solutions or worse representations will increase(Aljalbout et al. 2018).

Feature Extraction

After the description of the proposed model, this section focuses on the features used in the experiments. In the feature extraction stage, the segmented raw sensor signals are converted into feature vectors. Formally, let r_i denote the sample i in the set of the segmented raw sensor signals, x_i the converted feature vector, and C the feature extraction function. Then the feature extraction can be defined as:

$$x_i = C(r_i)$$

x_i is used as the input to the proposed model. Table 1 illustrates the statistical high level features that are used in the proposed approach. Mean, variance, standard deviation, and median, which are the most commonly adopted features in the HAR research works, are used in the approach. In addition, some other features, which have been shown to be efficient in previous works (Reyes-Ortiz et al. 2016), are included here as well. For example, the feature interquartile range (iqr), Quartiles (Q_1, Q_2 and Q_3) divide the time series signal into quarters. Using this, iqr is the measure of variability between the upper and lower quartiles, $iqr = Q_3 - Q_1$.

Table 1: List of the used statistical features.

Feature extraction function	Description
$mean(r_i) = \frac{1}{N}\sum_{j=1}^{N} r_{i_j}$	Mean
$var(r_i) = \frac{1}{N}\sum_{j=1}^{N}(r_{i_j} - mean(r_i))^2$	Variance
$std(r_i) = \sqrt{var(r_i)}$	Standard deviation
$median(r_i)$	Median values
$max(r_i)$	Largest values in array
$min(r_i)$	Smallest value in array
$iqr(r_i) = Q_3(r_i) - Q_1(r_i)$	Interquartile range

Table 2: The architecture of our approach for different datasets. Here, only the architecture of the encoder is shown. The decoder reverses the encoder.

Dataset	The number of neurons in each layer
PAMAP2	Input - 128 - 64
REALDISP	Input - 256 - 128
SBHAR	Input - 30 - 20

All these features are computed for each axis separately. Since the data from different sensors is synchronized, combining different sensor data is achievable. In the training process, the model takes these derived features as input and learns to retain the task-relevant information in the features and to disregard the unnecessary task-irrelevant parts.

Cluster Construction

To train the network, the standard backpropagation algorithm with stochastic gradient descent is used. All the weights are initialized to small values and the network is evaluated on the validation data after each epoch. When the validation error stops decreasing for a predefined number of epochs, the training process is complete.

After the learning process to establish the hidden representation within the AE architecture, the trained model (i.e. the encoder of the architecture) can project the input data sample x into a clustering-friendly embedding space. More specifically, with the temporal coherence loss and the locality preserving loss, the encoder in the model is learned to encode the essential features across multiple data samples and disregard individual or temporal details that are irrelevant to the clustering task. The learned representations are evaluated in the subsequent clustering task. k-means (KM), which is arguably the most popular clustering algorithm, is used in the experiments.

Evaluation and Experiments

To evaluate the proposed network, three publicly available benchmark datasets, which contain wearable sensor data of different human activities, are used in the experiments to verify the effectiveness of the approach. The architectures of the models for the three datasets are manually chosen, including the number of layers and the number of neurons in each layer. The architecture information is summarized in Table 2. The activation function used in the model is LeakyReLu.

Datasets

The three HAR datasets used here are PAMAP2 (Reiss and Stricker 2012a), REALDISP (Baños et al. 2012), and SB-HAR (Anguita et al. 2013). We use five-fold cross-validation to measure performance and all the sensor data sequences are segmented with the sliding window method.

PAMAP2 is collected from 9 participants performing 12 activities using 3 inertial measurement units placed on the wrist, chest and ankle. The dataset contains data of sport exercises (rope jumping, nordic walking, etc.), and household activities (vacuum cleaning, ironing, etc.). During the experiments, heart rate, accelerometer, gyroscope, magnetometer, and temperature data is recorded. In accordance with previous research on this dataset, a sliding window of 5.12 seconds with one second step size is used to segment the data.

REALDISP is recorded from 17 volunteers carrying out 33 activities using 9 sensors placed on both arms, both legs, and the back. Each sensor provides acceleration, gyroscope and magnetic field orientation. This dataset contains data of fitness exercises, warm up, and cool down. The sliding window used here has a size of 2 seconds without overlapping.

SBHAR is gathered from 30 participants performing 6 basic activities, such as walking, lying, etc., and 6 postural transitions, such as sit-to-lie and sit-to-stand. In our experiments, all the postural transitions are treated as one general transition. The dataset was collected using a smartphone placed on the waist of the participants. The data is segmented using a sliding window of 2.56 seconds with a step size of 1.28 seconds.

Validation Metrics

Three evaluation metrics are adopted to measure the performance of the approach: clustering accuracy (ACC), adjusted Rand index (ARI), and normalized mutual information (NMI). The ARI and NMI are computed as follows:

$$\text{ARI} = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i\binom{n_i}{2}\sum_j\binom{n_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i\binom{n_i}{2} + \sum_j\binom{n_j}{2}] - [\sum_i\binom{n_i}{2}\sum_j\binom{n_j}{2}]/\binom{n}{2}}$$

$$\text{NMI} = \frac{\sum_i\sum_j n_{ij}\log(\frac{n\cdot n_{ij}}{n_i\cdot n_j})}{\sqrt{\sum_i n_i\log\frac{n_i}{n}\sum_j n_j\log\frac{n_j}{n}}}$$

where n_{ij} is the number of samples in cluster i and class j, n_i is the number of samples in cluster i formed using the unsupervised approach, n_j is the number of samples in class j as indicated by the labels in the dataset, and n is the number of samples.

Results and Analysis

Our proposed approach is a domain-specific extension based on traditional AE, so we compare the performance of the proposed method with the traditional AE and principal components analysis (PCA). The results of the experiments are summarized in Table 3.

As shown in the table, our approach achieves improved results over all three datasets. KM is applied in the embedding space. The number of clusters is chosen manually, and different cluster numbers are tested. The true number of classes
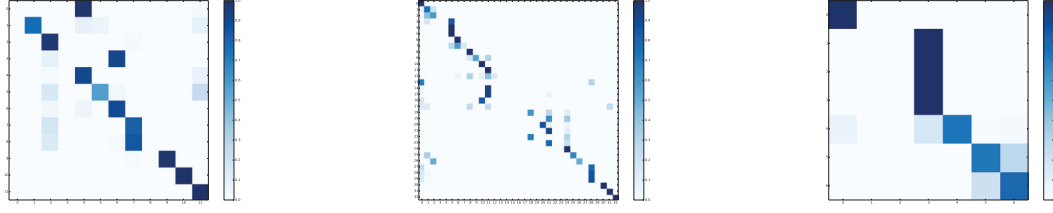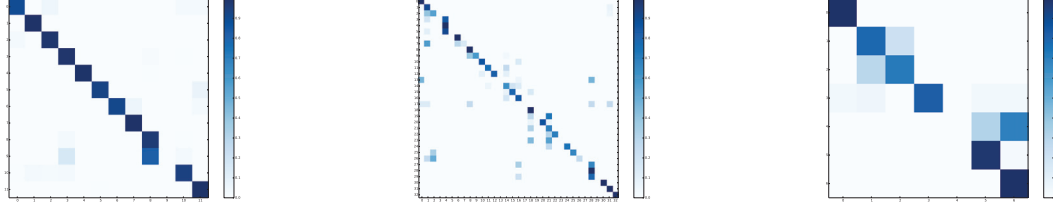
(a): PAMAP2   (b): REALDISP   (c): SBHAR

Figure 3: Confusion Matrices of Traditional AE on PAMAP2, REALDISP, and SBHAR.

(a): PAMAP2   (b): REALDISP   (c): SBHAR

Figure 4: Confusion Matrices of Proposed Method on PAMAP2, REALDISP, and SBHAR.

Table 3: The comparison between the proposed unsupervised approach and other unsupervised methods on the wearable sensor-based human activity datasets.

Methods	ACC	ARI	NMI
PAMAP2			
PCA + KM	0.6993	0.6440	0.7905
AE + KM	0.7706	0.6862	0.7994
Proposed Method + KM (tn)	0.8543	0.8016	0.8730
Proposed Method + KM ($tn + 1$)	0.8622	0.8089	0.8898
Proposed Method + KM ($tn + 2$)	**0.9211**	0.8288	0.8909
Proposed Method + KM ($tn + 3$)	0.9150	**0.8590**	**0.9144**
REALDISP			
PCA + KM	0.5723	0.4035	0.6890
AE + KM	0.6401	0.5446	0.7764
Proposed Method + KM (tn)	0.6812	0.6051	0.8043
Proposed Method + KM ($tn + 1$)	0.6829	0.6062	0.7965
Proposed Method + KM ($tn + 2$)	0.7057	0.6349	0.8215
Proposed Method + KM ($tn + 3$)	**0.7149**	**0.6508**	**0.8282**
SBHAR			
PCA + KM	0.6589	0.5784	0.7194
AE + KM	0.6369	0.5090	0.7048
Proposed Method + KM (tn)	0.7401	0.6343	0.7569
Proposed Method + KM ($tn + 1$)	0.7596	**0.6718**	**0.7982**
Proposed Method + KM ($tn + 2$)	0.8018	0.6548	0.7552
Proposed Method + KM ($tn + 3$)	**0.8073**	0.6645	0.7652

Table 4: The comparison between the proposed unsupervised approach and other supervised methods on the wearable sensor-based human activity datasets.

Methods	ACC
PAMAP2	
Probability SVM with Filter (Reyes-Ortiz et al. 2016)	0.9304
Decision Tree (C4.5) (Reiss and Stricker 2012b)	0.9709
Boosted C4.5 (Reiss and Stricker 2012b)	**0.9980**
Proposed Method + KM ($tn + 2$)	0.9211
REALDISP	
Probability SVM with Filter (Reyes-Ortiz et al. 2016)	**0.9952**
kNN (Baños et al. 2012)	0.9600
Proposed Method + KM ($tn + 3$)	0.7149
SBHAR	
Probability SVM with Filter (Reyes-Ortiz et al. 2016)	0.9678
CNN (Berggren 2018)	**0.9870**
Proposed Method + KM ($tn + 3$)	0.8073

in each dataset is used as the basis: tn. The results show that our approach can derive meaningful features to the subsequent activity clustering tasks. Figure. 3 and 4 also show and compare the performance of the traditional AE and the proposed approach by means of confusion matrices.

In addition, Table 4 shows the comparison between the best results from our unsupervised approach and the results

from previous published supervised methods on these three datasets. Note that, because ARI and NMI are metrics used to measure the performance of clustering algorithms, only ACC is adopted here to compare the results. As shown in the table, supervised methods can still achieve much better performance than unsupervised methods, but, as discussed before, the labeled data is usually difficult to acquire.

The results of the experiments showthe efficiency of the approach, but we also noticed some inaccuracies introduced by this approach. One problem is locality preserving loss will mix some similar activities. For example, the activities jogging and running are located closely in the embedding space. The possible reason is that the difference between jogging and running is subtle. Jogging can be seen as a slow

form of running. Moreover, different people jog or run at different speeds. Thus, the locality preserving loss can drive the model to project these two different activities to adjacent locations in the embedding space. Another problem is that when a person switches between different activities, the temporal coherence assumption does not hold. During the activity transition process, the temporally adjacent data samples may represent different types of activities, hence the temporal coherence assumption will introduce inaccuracy into the model.

However, the problems mentioned above are usually infrequent. Therefore, the proposed approach can still learn useful representations and boost the performance of the subsequent clustering algorithm.

Ablation Studies

To further understand the effect of the different loss terms, a set of ablation experiments are conducted to measure the influences of the different loss terms separately. In the ablation experiments, the original joint loss function is transformed into two separate objectives: (i) the temporal coherence (TC) loss with the AE reconstruction loss; (ii) the locality preservation (LP) loss with the AE reconstruction loss. Each objective is used to train the model separately, thus comparing their results should provide insight into the benefit of each loss term. The results of the ablation studies are listed in Table 5.

Table 5: Ablation studies on the effect of each loss term.

Methods	ACC	ARI	NMI
PAMAP2			
TC Loss + AE Loss	0.7859	0.7253	0.8140
LP Loss + AE Loss	0.8065	0.7493	0.8337
Joint Loss	**0.8543**	**0.8016**	**0.8730**
REALDISP			
TC Loss + AE Loss	0.6341	0.5348	0.7639
LP Loss + AE Loss	0.6610	0.5890	0.7829
Joint Loss	**0.6812**	**0.6051**	**0.8043**
SBHAR			
TC Loss + AE Loss	0.6449	0.5173	0.7087
LP Loss + AE Loss	0.7308	0.6182	0.7440
Joint Loss	**0.7401**	**0.6343**	**0.7569**

We notice that including both loss terms in the objective function can clearly improve the clustering performance. These results suggest that both loss terms guided the model to capture different useful information in the human activity sensor data in an unsupervised manner.

Conclusion

In this work, we have presented an unsupervised embedding learning approach, which is based on an autoencoder framework and uses the properties of human activities: temporal coherence and locality preservation, to project the activity data into the embedding space. We have demonstrated the effectiveness of the approach by applying it to three widely used HAR benchmark datasets. The results of the experiments show that our approach can group similar activities together in the embedding spaceand therefore help improve the performance of the subsequent clustering task.

References

Abu Alsheikh, M.; Selim, A.; Niyato, D.; Doyle, L.; Lin, S.; and Tan, H. 2016. Deep activity recognition models with triaxial accelerometers. In *AAAI Conference on Artificial Intelligence*, volume WS-16-01 - WS-16-15, 8–13. United States: AI Access Foundation.

Aljalbout, E.; Golkov, V.; Siddiqui, Y.; and Cremers, D. 2018. Clustering with deep learning: Taxonomy and new methods. *CoRR* abs/1801.07648.

Anguita, D.; Ghio, A.; Oneto, L.; Parra, X.; and L Reyes-Ortiz, J. 2013. A public domain dataset for human activity recognition using smartphones.

Baños, O.; Damas, M.; Pomares, H.; Rojas, I.; Tóth, M. A.; and Amft, O. 2012. A benchmark dataset to evaluate sensor displacement in activity recognition. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, 1026–1035. New York, NY, USA: ACM.

Berggren, K. 2018. Human activity recognition using deep learning and sensor fusion. Master's thesis, Lund University.

He, Z., and Jin, L. 2009. Activity recognition from acceleration data based on discrete consine transform and svm. In *2009 IEEE International Conference on Systems, Man and Cybernetics*, 5041–5044.

Kwapisz, J. R.; Weiss, G. M.; and Moore, S. A. 2011. Activity recognition using cell phone accelerometers. *SIGKDD Explor. Newsl.* 12(2):74–82.

Kwon, Y.; Kang, K.; and Bae, C. 2014. Unsupervised learning for human activity recognition using smartphone sensors. *Expert Syst. Appl.* 41:6067–6074.

Lu, Y.; Wei, Y.; Liu, L.; Zhong, J.; Sun, L.; and Liu, Y. 2017. Towards unsupervised physical activity recognition using smartphone accelerometers. *Multimedia Tools and Applications* 76(8):10701–10719.

Morales, F. J. O., and Roggen, D. 2016. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. In *Sensors*.

Reiss, A., and Stricker, D. 2012a. Introducing a new benchmarked dataset for activity monitoring. In *2012 16th International Symposium on Wearable Computers*, 108–109.

Reiss, A., and Stricker, D. 2012b. Creating and benchmarking a new dataset for physical activity monitoring. In *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '12, 40:1–40:8. New York, NY, USA: ACM.

Reyes-Ortiz, J.-L.; Oneto, L.; Samà, A.; Parra, X.; and Anguita, D. 2016. Transition-aware human activity recognition using smartphones. *Neurocomput.* 171(C):754–767.

Wiskott, L., and Sejnowski, T. J. 2002. Slow feature analysis: Unsupervised learning of invariances. *Neural Comput.* 14(4):715–770.