Activity Recognition Using a Few Label Samples

Heidar Davoudi[1], Xiao-Li Li[2], Nguyen Minh Nhut[2],
and Shonali Priyadarsini Krishnaswamy[2]

[1] School of Computer Engineering, Nanyang Technological University, Singapore
[2] Data Analytics Department, Institute for Infocomm Research, A*Star, Singapore

Abstract. Sensor-based human activity recognition aims to automatically identify human activities from a series of sensor observations, which is a crucial task for supporting wide range applications. Typically, given sufficient training examples for all activities (or activity classes), supervised learning techniques have been applied to build a classification model using sufficient training samples for differentiating various activities. However, it is often impractical to manually label large amounts of training data for each individual activities. As such, semi-supervised learning techniques sound promising alternatives as they have been designed to utilize a small training set L, enhanced by a large unlabeled set U. However, we observe that directly applying semi-supervised learning techniques may not produce accurate classification. In this paper, we have designed a novel *dynamic temporal extension* technique to extend L into a bigger training set, and then build a final semi-supervised learning model for more accurate classification. Extensive experiments demonstrate that our proposed technique outperforms existing 7 state-of-the-art supervised learning and semi-supervised learning techniques.

Keywords: Activity Recognition, Semi-Supervised Learning, Dynamic Temporal Extension.

1 Introduction

Sensor-based human activity recognition has received considerable attention due to its diverse applications such as healthcare systems [1], pervasive and mobile computing [5], and smart homes [10] etc. For example, in the healthcare applications, understanding elderly people's activities can not only provide real-time useful information to caregivers, but also facilitate a monitoring system to take proper actions (e.g. alert clinicians) in emergency cases (e.g. falling down).

While privacy and complexity are major issues in video-based activity recognition [11], different *supervised* learning approaches have been applied to activity recognition based on body-worn sensors. For example, Naïve Bayes (NB) [23], Hidden Markov Model (HMM) [12, 27], Conditional Random Field (CRF) [27] and Support Vector Machine (SVM) [23] etc. These approaches require *sufficient* labeled samples to train accurate classifiers. However, sample labeling is the most labor intensive and time consuming process in activity recognition [11] as different classes of related activities could have nondeterministic natures —

some steps of the activities can be performed in arbitrary order, and the starting and ending points of activities are difficult to define due to the overlaps among different activities, leading to activities could be concurrent or even interwoven [11]. As such, it would be extremely hard to get *sufficient* training samples.

Semi-supervised learning approaches, on the other hand, have been designed to build a classification model using a small training set L and a large unlabeled set U. Particularly, an initial classifier can be built from L by using NB, SVM or another classifier. Then the classifier can be used to classify U to assign a (probabilistic) class label to each sample in U [14, 15], which can be in turn used to refine the classifier, using Expectation-Maximization (EM) [22] or other classifiers (like SVM) iteratively [13, 16–19]. The combination of NB and EM, and transductive Support Vector Machine (TSVM) [8] are two well known semi-supervised learning methods. In addition, Guan *et al* [7] designed En-Co-training algorithm for activity recognition based on co-training algorithm [2] which trains two independent classifiers based on two views of data. Stikic *et al* [26] compared two semi-supervised techniques, namely self-training [4] and co-training, and showed co-training can be applied to activity recognition by using sensors with different modalities, i.e. accelerometers and infra-red sensors. While a few semi-supervised learning algorithms have been proposed, they still need sizable labeled samples for training to achieve accurate results. The reason is that these methods will encounter a cold-start issue when very few labeled samples are present for learning a decent initial classifier. Its poor performance subsequently affects the iterative model refinement process. As such, directly applying semi-supervised learning techniques may not always produce accurate prediction results.

In this paper, we address the challenging issue by proposing a novel *Dynamic Temporal Extension* (DTE) algorithm. Particularly, based on a few labeled samples, we automatically infer some *reliable labeled samples* from U in temporal space [21]. For each labeled sample, we extend it to include its near neighbors along its time axis, if these neighbors are predicted mostly to have a same label with the labeled sample. In effect, our DTE algorithm tries to fill the gap between the *limited* available labeled samples and *sizable* labeled samples required for semi-supervised learning. With our inferred reliable labeled samples, we can thus build a more robust and accurate semi-supervised learning model. Extensive experiments demonstrate that our proposed technique outperforms existing 7 state-of-the-art supervised learning and semi-supervised learning techniques.

2   The Proposed Technique

In this section, we present our proposed technique. First, section 2.1 provides our overall algorithm. Then, we introduce how to build a classifier using our designed semi-supervised learning approach in section 2.2. Finally, section 2.3 elaborates how to extend limited labeled samples using our proposed DTE technique.

2.1   Overall Algorithm

Fig. 1 shows an overall algorithm of our proposed technique. Step 1 is a preprocessing that deals with missing value problem as well as faulty sensors. Missing

---

Input: L, U ▷ Set of Initial Labeled Samples and Unlabeled Samples respectively
Output: M ▷ Classification Model

1: Handle missing values and faulty sensors in a given activity recognition data set
2: Build an initial semi-supervised classifier IC using L and U
3: Apply IC to predict the labels of samples in U and perform smoothing
4: Compute continuous same-label segment size SS of activities using predicted labels in Step 3
5: Extend labeled samples into set E using DTE with the estimated segment size SS
6: Build a final semi-supervised learning model M based on initial labeled set L, the inferred labeled set E, as well as the remaining unlabeled set $U - E$
7: Classify the test samples using the model M

Fig. 1. Overall DTE algorithm for activity recognition using few labeled samples

value problem can be caused by some faulty sensors or unreliable communication channels. We first eliminate sensors which have more than 30% missing values as they can not provide useful and sufficient information for accurate activity recognition. We then apply cubical spline interpolation for replacing those missing values in our data sets [3] and eliminate samples with no activity label.

In order to prepare for labeled sample extension, we want to estimate the median size of continuous same-label activity sequences in $L + U$, as this is useful information that indicates how far we should extend each labeled sample. Since we do not have the label information for all the samples in U, we first design a semi-supervised learning method to build an initial classifier IC in step 2 — we provide its detailed description in section 2.2. Then, we apply IC to classify the unlabeled samples in U in step 3. Given that sensor sampling rates are usually significantly higher than the rates of change between different human activities, we thus perform *smoothing* step to improve the prediction accuracy by removing impulse noises. Our smoothing process moves a sliding window on predicted labels of U and chooses the labels with majority as the label of all the samples in that window. Step 4 computes the median continuous same-label segment size SS of activities based on the predicted labels in U. Step 5 extends all the labeled samples into the inferred reliable labeled set E based on our designed DTE techniques using the prediction results from IC as well as the estimated segment size SS. We have provided the details of step 5 in section 2.3.

Finally, we build the final classification model M using our semi-supervised learning method, based on labeled set L, the extended set E, and the remaining unlabeled set $U - E$ in step 6, and perform final classification in step 7. Note the same method is used for both initial classifier IC and final classifier although the final one is more accurate due to additional reliable labeled sample E.

2.2   Classifier Building Based on Semi-supervised Learning

First, labeled set L and unlabeled set U are defined as follows:

$$L = \{(A_k, C_k) | k = 1, 2, \ldots, |L|\} \tag{1}$$

$$U = \{A_k | k = 1, 2, \ldots, |U|\} \tag{2}$$

$A_k = (a_{k1}, a_{k2}, ..., a_{kn})$ is a n-dimensional feature vector where $a_{ki}(i = 1, 2, ..., n)$ is the i'th feature of A_k and $C_k \in C(C = \{C_1, C_2, ..., C_{|C|}\})$ is the associated activity/class label of A_k. The objective of this research is to learn a classification model M that can be used to classify any test sample.

We adopt a mixture model to generate each activity A_k as mixture models are very useful to characterize heterogeneous data or multiple types of activities.

$$P(A_k|\theta) = \sum_{j=1}^{|C|} P(C_j|\theta)P(A_k|C_j; \theta) \tag{3}$$

Where θ parameterized a mixture model. Given any class C_j, assuming that the probabilities of the features are independent as well as normal distribution (based on the central limit theorem) for each attribute in C_j, we have

$$P(A_k|C_j; \theta) = P(A_k|C_j; \mu_j, \sigma_j) = \prod_{i=1}^{n} P(a_{ki}|C_j; \mu_{ij}, \sigma_{ij}) \tag{4}$$

and

$$P(a_{ki}|C_j; \mu_{ij}, \sigma_{ij}) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} e^{-\frac{(a_{ki}-\mu_{ij})^2}{\sigma_{ij}^2}} \tag{5}$$

Note $\mu_j = (\mu_{1j}, \mu_{2j}, ..., \mu_{nj})$ and $\sigma_j = (\sigma_{1j}, \sigma_{2j}, ..., \sigma_{nj})$ where μ_{ij} and σ_{ij} are the mean and standard deviation of feature i in class j respectively.

In order to perform classification, we compute the posterior probability of class C_j for a given example A_k:

$$P(C_j|A_k; \mu, \sigma) = \frac{P(C_j|\mu, \sigma) \prod_{i=1}^{n} P(a_{ki}|C_j; \mu_{ij}, \sigma_{ij})}{P(A_k|\mu, \sigma)} \tag{6}$$

Where $\mu = [\mu_1, \mu_2, ..., \mu_{|C|}]$ and $\sigma = [\sigma_1, \sigma_2, ..., \sigma_{|C|}]$. The class C_k with $k = \arg\max_j P(C_j|A_k; \mu, \sigma)(j = 1, ..., |C|)$ will be assigned to the example A_k as its predicted class label. Note in formula (6), the denominator $P(A_k|\mu, \sigma)$ is a constant, i.e. same for all classes. We now elaborate how to calculate $P(C_j|\mu, \sigma)$. Based on formula (3), the probability of both labeled and unlabeled training set $(S = L \cup U)$ can be written as:

$$P(S|\mu, \sigma) = \prod_{A_k \in U} \sum_{j=1}^{|C|} P(C_j|\mu, \sigma)P(A_k|C_j; \mu, \sigma) \times$$
$$\prod_{(A_k, C_k) \in L} P(C_k|\mu, \sigma)P(A_k|C_k; \mu, \sigma) \tag{7}$$

Local maximum of log likelihood indicated by (7) can be found using Expected Maximization (EM) in an iterative manner [6]. It can be shown that parameters in M-step of EM algorithm can be computed as follow:

$$\mu_{ij}^{t+1} = \frac{\sum_k a_{ki} P(C_j|A_k; \mu^t, \sigma^t)}{\sum_k P(C_j|A_k; \mu^t, \sigma^t)} \tag{8}$$

$$\sigma_{ij}^{2\ t+1} = \frac{\sum_k (a_{ki} - \mu_{ij}^t)^2 P(C_j|A_k; \mu^t, \sigma^t)}{\sum_k P(C_j|A_k; \mu^t, \sigma^t)} \tag{9}$$

Input: L, U ▷ Set of Initial Labeled Samples and Unlabeled Samples respectively
Output: E ▷ Set of Extended Samples

```
 1: E ← ∅
 2: CfTable[ ][ ] ← 0
 3: for all s ∈ L do
 4:     Cj = label(s)
 5:     ExtS ← EXTENDSAMPLE(s, Cj)
 6:     for all p ∈ ExtS do
 7:         if p ∉ E then
 8:             E ← E ∪ {p}
 9:             Add p into CfTable
10:             CfTable[p][Cj] ← 1
11:         else
12:             CfTable[p][Cj] ← CfTable[p][Cj] + 1
13:         end if
14:     end for
15: end for
16: for all p ∈ E do
17:     label(p) ← arg max CfTable[p][j]
                   j
18: end for
```

Fig. 2. DTE algorithm on labeled example extension

Finally, the class prior probability can be calculated:

$$P(C_j|\mu^{t+1}, \sigma^{t+1}) = \frac{\sum_k P(C_j|A_k; \mu^{t+1}, \sigma^{t+1})}{|L| + |U|} \tag{10}$$

Note that we have modified the original EM algorithm by setting the posterior class probabilities of initially labeled samples in L as their original labels, at the start of each iteration. This guarantees our classification model is built using *correct* labeled examples during its iterative training process.

2.3 Dynamic Temporal Extension for Labeled Samples

In section 2.2, we have proposed a semi-supervised method to learn a classification model from a small labeled set L and a large unlabeled set U. In our case, since L is very small, say a few training examples for each class, the semi-supervised learning method will not capture the characteristics for each class. As such, we have proposed a novel *dynamic temporal extension* algorithm DTE to extend the initial labeled samples into an extended reliable labeled set E by exploring the dependencies between samples in time domain to effectively boost the performance of semi-supervised learning.

The proposed DTE algorithm for labeled example extension is shown in Fig. 2. After we initialize the extended set E into an empty set in step 1, a confidence table $CfTable[\ ][\ ]$ is created and initialized in step 2 which records the support of an unlabeled example to a given class, e.g. $CfTable[p][C_j]$ indicates how likely and the degree of confidence that an example p belongs to a class C_j. From steps 3–15, we extend each labeled sample one by one and compute the confidence table correspondingly. Particularly, for each seed labeled sample s with class label C_j, we extend it to include its neighbor samples to form $ExtS$
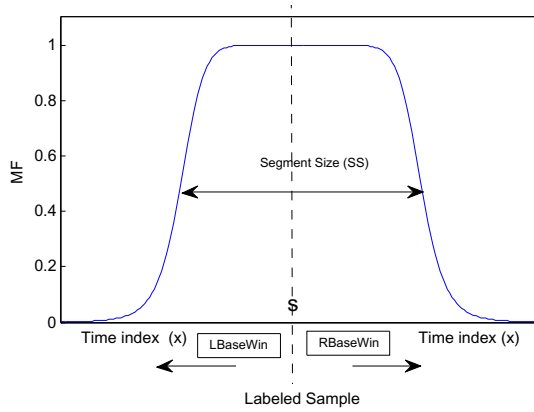
Fig. 3. Membership Function (MF)

along a time line (left/ right hand sides of s) if we judge these samples in $ExtS$ belong to the same class, i.e. C_j, by calling a function EXTENDSAMPLE() that will be described in Fig 4 later.

Steps 6 to 14 is a loop, which computes the support for each extended sample in $ExtS$ by updating the confidence table $CfTable[p][C_j]$. When selected sample p does not exist in the table, it is inserted to the set E (step 8) and *CfTable* with the initial value 1 (steps 9 to 10); otherwise our algorithm increases its respective support value by 1. Finally, steps 16 to 18 assign each sample in E with a label with a maximal class support score wrt to its associated classes in *CfTable*.

Now, we are ready to introduce the function EXTENDSAMPLE() that is used in Fig. 2. We first illustrate the key intuition behind it in Fig. 3. As the sampling frequency of sensors is much higher than the change frequency between different human activities, given an initial labeled sample s at time point t_0, the neighbor samples at its near left time points and at its near right time points are highly likely to share the same label with s, as long as these left/right time points are not too far away from t_0. However, one challenging problem is how long/far we should extend s to its left/right hand side.

In this paper, we use a sliding window to estimate right and left label-consistent neighbor samples of s. Two base sliding windows (left/right sliding window LBaseWin/RBaseWin) are moved forward from both sides of s individually and our algorithm decides whether the extension should be continued or stopped based on the analysis of the neighbor samples' labels inside both sliding windows (Fig. 3). Two important information sources have been used as a stop criteria, namely, 1) neighbor label consistency with s where the labels of s's neighbors are predicted by our initial classification model IC, and 2) the distance of neighbor samples to s. We design a membership function MF based on a distance measure to s as follows:

$$MF(x, s, SS) = \frac{1}{1 + |\frac{x-s}{0.5*SS}|^{SS}} \tag{11}$$

where x is a current neighbor sample inside in a left/right window; s and SS are the center and width of membership function respectively (Fig 3) where SS is

estimated in our overall algorithm by computing the median size of continuous same-label activity sequences. The membership function defined in (11) is a bell shaped fuzzy membership function. As sliding windows move forward from s to left/right side, membership value decreases corrspondingly. The bigger this distance $x - s$, the less likely the neighbor sample x should be included. We also take the label-consistency into consideration by computing a support score $Support\_Score_{C_j}$ for all the predicted class labels $C_j \in C$ in current base window (LBaseWin/RBaseWin):

$$Support\_Score_{C_j} = \sum_{\substack{x \in RBaseWin \\ label(x)=C_j}} MF(x, s, SS) \qquad (12)$$

Score $Support\_Score_{C_j}$ defined in (12) basically calculates the label support inside current (right) base window based two information sources mentioned. We will continue our extending process if these labels in the base window are consistent with s's label and they are not too far away from s. Depending on the two factors, we extend s *dynamically*, and the numbers of extended examples are different for each initial labeled sample as well as for the left and right extension.

The detailed description of function ExtendSample() is shown in Fig 4. Note we only show how to extend right samples from s as extending left samples is the same except the extension direction. After initializing the extended sample set $ExtS$ as empty set in step 2, we perform a loop from steps 3 to 18. Particularly, steps 5 to 9 sum up all the class membership scores for each example in the current base window. Steps 10 to 11 get the *Winner score* and *Winner Label* from all the classes. Steps 12 to 17 decide if we want to repeat our loop. If the *Winner Label* is consistent with s and the *Winner score*>=50%, then the samples inside the current window are reliable, and we thus add them into $ExtS$ and shift the right window to continue our extension process; Otherwise, we stop the loop from here. Finally, step 19 returns our extension results.

3   Empirical Evaluation

In this section, we evaluate our proposed DTE technique. We compare it with 7 existing state-of-the-art techniques, including 3 *supervised learning techniques*, namely, 1NN (which performs best for time series data) [9], SVM [23], NB (Naïve Bayes) [23], as well as 4 *semi-supervised learning techniques*, namely, Transductive Support Vector Machine (TSVM) [8], NB+EM (NB+Expected Maximization) [22], Self-training [20] and En-Co-training [7] etc.

3.1   Datasets

For evaluation, we used 3 datasets from Opportunity Challenge [24, 25]. Each dataset represents 1 subject/person who performed various activities with attached on-body sensors. Particularly, each subject performed one *Drill* session which is 20 predefined atomic activities and 5 *Activities Daily Life* (ADLs) sessions where the subject performed high level activities with more freedom on sequence of atomic activities. Datasets contain 4 locomotion activities/classes,

Input: s, C_j ▷ s and C_j are a labeled sample to extend and its associated class respectively
Output: $ExtS$ ▷ A set of extended samples for the labeled sample s

```
 1: function EXTENDSAMPLE(s, C_j)
 2:     ExtS ← ∅
 3:     Acceptable ← true
 4:     repeat                                         ▷ Extension towards right
 5:         for i ← 1 to |C| do
 6:             for all x located inside right based window RBaseWin do
 7:                 Support_Score_i ← ∑_{label(x)==i} MF(x, s, SS)
 8:             end for
 9:         end for
10:         WinnerScore ← max Support_Score_i(i = 1, 2, . . . , |C|)
11:         WinnerLabel ← arg max Support_Score_i(i = 1, 2, . . . , |C|)
                            k
12:         if WinnerLabel==C_j and WinnerScore ≥ 50% then
13:             ExtS ← ExtS∪ {the samples inside RBaseWin}
14:             RBaseWin ← ShiftRight(RBaseWin)
15:         else
16:             Acceptable ← false
17:         end if
18:     until Acceptable == false
19:     Return ExtS
20: end function
```

Fig. 4. EXTENDSAMPLE procedure of DTE algorithm

namely, *stand*, *walk*, *sit*, and *lie*. All the experiments are performed across 3 different datasets. Drill and first 3 ADLs (ADL 1, ADL2, ADL3) are used for training while the last 2 ADLs (ADL4, ADL5) are used for testing.

Table 1. Opportunity Challenge Data Sets After Preprocessing

	Dataset 1	Dataset 2	Dataset 3
Number of Features	107	110	110
Training Set	139427	126595	141263
Test Set	47291	48486	45774

Table 1 shows the details of the 3 datasets that we used in our experiments after the preprocessing step introduced in section 2.1. To simulate the challenging scenario of learning from limited training samples, for each dataset we randomly select only 3 samples from Training Set for each of the 4 classes, i.e. 12 (3×4) samples to serve as the labeled set L. We then randomly select 1000 training samples as unlabeled set U which are used by all the semi-supervised learning methods after ignoring their labels. We have also tested how the sizes of U affect the final classification performance. In addition, the base window size (RBaseWin) is set to 10 (Fig 4) and we will report its sensitivity study results. Finally, to make the evaluation reliable, all the experiments are repeated for 10 times and the performances of all the 8 techniques are evaluated on the same test sets in terms of average *Accuracy* and *F-measure*, which are typically used for evaluating the performance of classification models. Note that locomotion activities/classes is moderately imbalanced [3] so we use *Accuracy* and *F-measure* as performance measurements.

Table 2. Comparison of different supervised and semi-supervised learning methods

Methods	Dataset 1		Dataset 2		Dataset 3	
	Accuracy	*F-Measure*	*Accuracy*	*F-Measure*	*Accuracy*	*F-Measure*
1NN	75.5	78.8	58.0	65.1	59.9	67.7
SVM	9.1	24.0	16.0	27.9	15.5	26.7
NB	64.3	76.9	66.0	71.8	50.5	58.2
TSVM	61.4	69.8	54.6	57.4	40.4	49.0
NB+EM	77.7	82.9	75.9	77.6	57.0	56.1
Self-Training	41.8	16.8	36.8	13.5	42.5	14.9
En-Co-Training	59.1	65.8	47.4	64.5	56.6	64.3
DTE	79.2	83.6	77.0	78.4	68.2	68.6
DTE+smoothing	80.2	84.3	77.6	78.9	68.8	69.0

3.2 Experimental Results

The experimental results are shown in Table 2. The first 3 rows show the performance of 3 supervised classification models, i.e. 1NN, SVM, NB, where only initial labeled set (L) is used for training. We observe that SVM performs badly compared to 1NN and NB as SVM can not build accurate hyperplanes based on 12 labeled samples for 4-class classification problem.

The next 4 rows illustrate the experimental results of 4 semi-supervised methods, which have used unlabeled set U to boost their performance. Compared with supervised learning methods, we observe that TSVM is much better than SVM for all datasets, and NB+EM outperforms NB for datasets 1 and 2, while it is slightly worse in dataset 3 according to *F-measure* only. In general, NB+EM works better than TSVM, Self-training and En-Co-Training most of times.

The last two rows in Table 2 show the performance of DTE method and the effect of smoothing for test results. We observe DTE method consistently works better than all the other 7 techniques across all the datasets. Especially for dataset 3, DTE method achieves 11.2% and 12.5% better results than the second best technique NB+EM in terms of average Accuracy and F-measure respectively. We also perform the post-processing by applying smoothing (choose majority labels inside a sliding window) on our predicted test results to remove the impulse prediction errors, which further improve our method in all cases.

Recall that our proposed DTE method has dynamically extended initial labeled samples to left/right windows to include those neighbor samples that have consistent-label and are not too far away from the labeled samples. For comparison, we have also used a fixed extension size (FES) which is obtained by averaging the segments sizes of activities based on the actual labels in U. We observe that our proposed DTE perform much better than those semi-supervised methods (e.g. TSVM and NB+EM) that use the FES to include neighbor samples, indicating our method is very effective to automatically extend initial labeled samples dynamically to include those reliable labeled samples so that it can boost the performance of semi-supervised learning significantly.
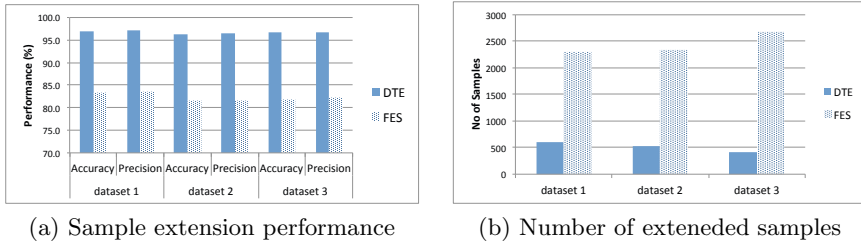
(a) Sample extension performance

(b) Number of exteneded samples

Fig. 5. No/Performance of different sample extension methods

(a) Accuracy

(b) F-measure
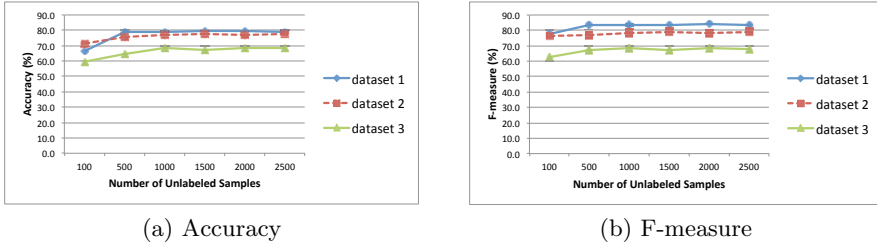
Fig. 6. Performance of DTE using diffrent unlabeled samples
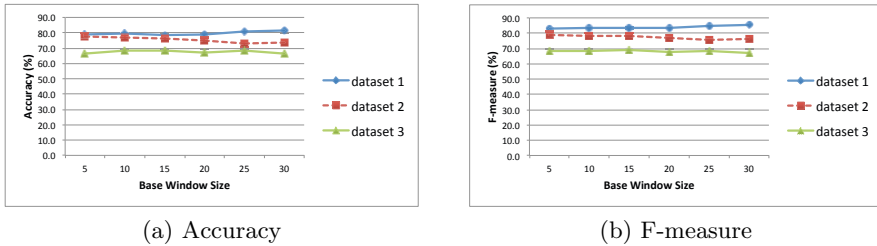

(a) Accuracy

(b) F-measure

Fig. 7. Performance of DTE using diffrent base window size

Fig. 5 shows the comparison between the performance of proposed DTE sample extension and FES sample extension. From Fig. 5(a), the proposed DTE algorithm can produce much more accurate labeled samples than FES, i.e. more than 15% better in terms of both Accuracy and Precision. While FES produces more samples (Fig. 5(b)), they are not accurate and thus bring noisy labels for the subsequent training process, leading to inferior classification results.

Fig. 6 illustrates the performance of proposed DTE method for different datasets in terms of the number of unlabeled samples used. From Fig. 6, with the increase of unlabeled samples, the performance of classification increases in the beginning and it becomes stable, with at least 1000 unlabelled samples, indicating our method is not sensitive to the number of unlabelled samples used.

Finally, Fig. 7 shows the effect of changing base window size on DTE approach performance (Accuracy/F-measure). As we can be observed, by increasing the base window size, the performance of our approach only changes slightly across

all the datasets in terms of both average Accuracy and F-measure, indicating that our method is not sensitive to the base window size.

4   Conclusions

Learning from a few labeled samples becomes crucial for real-world activity recognition applications as it is labor-intensive and time-consuming to manually label large numbers of training examples. While semi-supervised learning techniques have been proposed to enhance supervised learning by incorporating the unlabelled samples into classifier building, it still can not perform very well. In this paper, we propose a novel dynamic temporal extension technique to extend the limited training examples into a larger training set which can further boost the performance of semi-supervised learning. Extensive experimental results show that our proposed technique significantly outperforms existing 7 state-of-the-art supervised learning and semi-supervised learning techniques. For our future work, we will investigate how to select the initial labeled samples intelligently so that they can benefit our classification models more effectively.

References

1. Avci, A., Bosch, S., Marin-Perianu, M., Marin-Perianu, R., Havinga, P.: Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey. In: The 23rd International Conference on Architecture of Computing Systems (ARCS), pp. 1–10 (2010)
2. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the 11th Annual Conference on Computational Learning Theory, pp. 92–100 (1998)
3. Cao, H., Nguyen, M.N., Phua, C., Krishnaswamy, S., Li, X.: An integrated framework for human activity classification. In: Ubicomp (2012)
4. Chapelle, O., Schölkopf, B., Zien, A.: Semi-supervised learning, vol. 2. MIT Press, Cambridge (2006)
5. Choudhury, T., Consolvo, S., Harrison, B., Hightower, J., LaMarca, A., LeGrand, L., Rahimi, A., Rea, A., Bordello, G., Hemingway, B., et al.: The mobile sensing platform: An embedded activity recognition system. IEEE Pervasive Magazine, Spec. Issue on Activity-Based Computing 7(2), 32–41 (2008)
6. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society. Series B (Methodological), 1–38 (1977)
7. Guan, D., Yuan, W., Lee, Y.K., Gavrilov, A., Lee, S.: Activity recognition based on semi-supervised learning. In: 13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, pp. 469–475 (2007)
8. Joachims, T.: Making large-scale svm learning practical. In: Schölkopf, B., Burges, C., Smola, A. (eds.) Advances in Kernel Methods-support Vector Learning (1999)
9. Keogh, E., Kasetty, S.: On the need for time series data mining benchmarks: a survey and empirical demonstration

10. Kidd, C.D., Orr, R., Abowd, G.D., Atkeson, C.G., Essa, I.A., MacIntyre, B., My-natt, E., Starner, T.E., Newstetter, W.: The aware home: A living laboratory for ubiquitous computing research. In: CoBuild, pp. 191–198 (1999)
11. Lara, O., Labrador, M.: A survey on human activity recognition using wearable sensors. IEEE Communications Surveys Tutorials PP(99), 1–18 (2002)
12. Lester, J., Choudhury, T., Borriello, G.: A practical approach to recognizing phys-ical activities. In: Fishkin, K.P., Schiele, B., Nixon, P., Quigley, A. (eds.) PERVA-SIVE 2006. LNCS, vol. 3968, pp. 1–16. Springer, Heidelberg (2006)
13. Li, X., Liu, B.: Learning to classify texts using positive and unlabeled data. In: IJCAI, pp. 587–592 (2003)
14. Li, X.-L., Liu, B.: Learning from positive and unlabeled examples with different data distributions. In: Gama, J., Camacho, R., Brazdil, P.B., Jorge, A.M., Torgo, L. (eds.) ECML 2005. LNCS (LNAI), vol. 3720, pp. 218–229. Springer, Heidelberg (2005)
15. Li, X.-L., Liu, B., Ng, S.-K.: Learning to classify documents with only a small positive training set. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 201–213. Springer, Heidelberg (2007)
16. Li, X., Liu, B., Ng, S.: Learning to identify unexpected instances in the test set. In: IJCAI, pp. 2802–2807 (2007)
17. Li, X., Liu, B., Ng, S.: Negative training data can be harmful to text classification. In: EMNLP, pp. 218–228 (2010)
18. Liu, B., Lee, W., Yu, P., Li, X.: Partially supervised classification of text docu-ments. In: ICML, pp. 387–394 (2002)
19. Liu, B., Yang, D., Li, X., Lee, W., Yu, P.: Building text classifiers using positive and unlabeled examples. In: ICDM, pp. 179–186 (2003)
20. Longstaff, B., Reddy, S., Estrin, D.: Improving activity classification for health applications on mobile devices using active and semi-supervised learning. In: 4th International Conference on Pervasive Computing Technologies for Healthcare, pp. 1–7 (2010)
21. Nguyen, M., Li, X., Ng, S.: Positive unlabeled learning for time series classification. In: IJCAI, pp. 1421–1426 (2011)
22. Nigam, K., McCallum, A., Thrun, S., Mitchell, T.: Learning to classify text from labeled and unlabeled documents. In: AAAI, pp. 792–799 (1998)
23. Ravi, N., Dandekar, N., Mysore, P., Littman, M.L.: Activity recognition from ac-celerometer data. In: AAAI, vol. 20, p. 1541. AAAI Press, MIT Press, Menlo Park, Cambridge (1999, 2005)
24. Roggen, D., Calatroni, A., Rossi, M., Holleczek, T., Forster, K., Troster, G., Lukow-icz, P., Bannach, D., Pirkl, G., Ferscha, A., et al.: Collecting complex activity datasets in highly rich networked sensor environments. In: 7'th International Con-ference on Networked Sensing Systems, pp. 233–240 (2010)
25. Sagha, H., Digumarti, S.T., del Millan, J., Chavarriaga, R., Calatroni, A., Roggen, D., Troster, G.: Benchmarking classification techniques using the opportunity hu-man activity dataset. In: IEEE International Conference on Systems, Man, and Cybernetics, pp. 36–40 (2011)
26. Stikic, M., Van Laerhoven, K., Schiele, B.: Exploring semi-supervised and active learning for activity recognition. In: 12th IEEE International Symposium on Wear-able Computers, pp. 81–88 (2008)
27. Vail, D.L., Veloso, M.M., Lafferty, J.D.: Conditional random fields for activity recognition. In: Proceedings of 6th International Joint Conference on Autonomous Agents and Multiagent Systems, p. 235 (2007)