Activity Recognition with the Aid of Unlabeled Samples

Donghai Guan
Ubiquitous Computing Lab
Kyung Hee University, Korea
donghai@oslab.khu.ac.kr

Young-Koo Lee
Ubiquitous Computing Lab
Kyung Hee University, Korea
yklee@khu.ac.kr

Sungyoung Lee
Ubiquitous Computing Lab
Kyung Hee University, Korea
sylee@oslab.khu.ac.kr

ABSTRACT
Activity recognition is an important topic in ubiquitous computing. In activity recognition, supervised learning techniques have been widely applied to learn the activity models. However, most of them can only utilize labeled samples for learning even though a large amount of unlabeled samples exist. In our previous work, we have proposed a semi-supervised learning method which can utilize both labeled and unlabeled samples for learning. As an alternative, a new learning method is proposed in this work. It makes use of the unlabeled samples to remove the noises from labeled samples, so that the learning performance is improved. Experimental results show the effectiveness of our method.

Categories and Subject Descriptors
I.5.4 [**Pattern Recognition**]: Applications

General Terms
Algorithms

Keywords
Activity Recognition, Supervised Learning, Semi-supervised Learning, Noise Filtering

1. INTRODUCTION
The purpose of activity recognition is to infer people's behaviors from low-level data acquired through sensors in a given setting, based on which other critical decisions are made. For example, in smart home environments for aged care monitoring [1][2], based on the information provided by cameras and other pervasive sensors, the system needs to automatically monitor the occupant and determine when they need assistance, raising an alarm if required.

Machine learning is a key aspect in activity recognition. For a system to automatically infer what activity is being performed, it must have a detailed model of the activity which is generated by machine learning method.

Most existing learning methods for activity recognition can only utilize labeled activity samples for learning, although usually large amounts of unlabeled samples exist as they do not need human's labeling effort. Rather than only focusing on the limited labeled data, we are interested in how to utilize the cheap unlabeled data to boost the learning performance.

In our previous work [3], we have proposed a learning method for this purpose. From the viewpoint of machine learning, that method belongs to semi-supervised learning category [4]. That method mainly consists of predicting the labels for the unlabeled samples and then augments the training samples by incorporating these new labeled samples. The success of that method depends on the augmentation of training samples. However, the bigger training data set has disadvantage of degrading the performance for some machine learning algorithms, such as k nearest neighbors.

In order to overcome this performance degradation due to the big size of training set, we require a method which can improve the recognition accuracy without increasing the number of training data. In this work, we adopt the learning method in [18] for activity recognition. Noise filtering is adopted by our method. It removes the noises in the labeled samples to improve the recognition performance. Traditional noise filtering methods are only based on labeled samples. Our contribution is to utilize the unlabeled samples to boost the performance of noise filtering in order to achieve higher recognition accuracy.

The experimental results show that noise filtering can improve the activity recognition performance. Moreover, the performance is further improved with the aid of unlabeled data.

The rest of paper is organized as follows: Section 2 presents some related work on activity recognition. Section 3 presents noise filtering. Then experimental results are given in Section 4. Section 5 includes conclusions and future work.

2. RELATED WORK
A typical workflow for activity recognition mainly includes activity samples collection, samples labeling and classifier training (activity modeling).

The first step is to collect activity samples. Samples collection could be characterized by the different usages of sensors. These usages include, (1) remotely observe the scene using audio, visual, electromagnetic field, or other sensors and interpret the signal readings [5][6][7], (2) attach sensors to the body and interpret the signal readings [8][9][10], (3) attach sensors to objects and devices in the environment and interpret the sensor readings [11][12].

After data collection, some samples are randomly selected and labeled by human. These samples will be used by the classifiers

ICUIMC-09, January 15-16, 2009, Suwon, S. Korea

for training purpose. Currently a variety of classifiers have been proposed for activity recognition, such as neural networks [13], dynamic Bayesian networks [14], naïve Bayesian networks [15], hierarchical hidden semi-Markov models [16], nearest neighbors [10], decision tree [10] and so on.

Activity models are generated through training classifiers. While the activity is being carried out, data is gathered from sensors. Then the activity data is compared to a set of activity models and inferred which model is the best match. The quality of activity models determines the performance of activity recognition.

Existing methods introduced above can only utilize labeled samples for learning. Considering that unlabeled samples are easy to obtain, it is important to utilize these cheap data to improve the learning performance. In our previous paper [3], we have proposed a semi-supervised learning method for activity recognition. Although it can improve the recognition accuracy, meanwhile it results in the augmentation of training data number, which

3. NOISE FILTERING

We think noise filtering is a suitable technique which can meet our requirements as it improves the classification accuracy by removing noises in the training data which leads to the reduction of training data. Although noise filtering is expected to improve the performance for activity recognition, most existing noise filtering methods cannot utilize unlabeled samples. We argue that approximately using unlabeled samples can aid noise filtering and then better recognition accuracy can be achieved.

Normally different noise filtering techniques are proposed for different machine learning methods. Therefore, among exiting machine learning methods, we have to choose one to present our idea. In this paper, we select K nearest neighbor (KNN) based on the following reasons: 1) KNN is highly susceptible to noise in the training samples as its high degree of local sensitivity. 2) KNN prefers to small number of training data because it is a type of lazy learning where all computation is deferred until classification. It works slower when the number of training data is increased.

In Section 3.1, the existing noise filtering technique in KNN is introduced. In Section 3.2, we present our idea which utilizes unlabeled samples to boost the noise filtering performance for KNN.

3.1 Noise Filtering Techniques in KNN

There are many noise removing techniques in KNN. Herein, we consider the edited nearest neighbor (ENN) [17] due to its popular use.

It removes all instances that have been misclassified by the KNN rule from the training set.

Figure 1 shows the effect of ENN. In this figure, the hollow rounds and the solid rounds represent the instances which belong to two different classes. The left part shows a hypothesis training set where misclassified instances using the 1-NN rule are marked with dotted circles around them. The right part of Fig. 1 shows the reduced dataset after applying ENN. The algorithm of ENN is given in Algorithm 1.

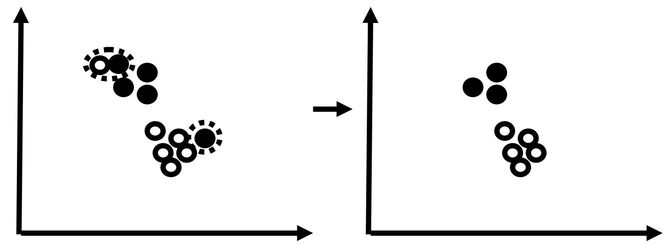

Figure 1. Original training set and the reduced training set by edited nearest neighbor.

Algorithm 1
The algorithm of edited nearest neighbor (ENN)

---

1. Let $T_e = T$ // T is the original training set, and T_e is the edited set
2. For each $x_i \in T_e$, do:
Discard x_i from T_e if it is misclassified using the k-NN rule with prototypes in $T_e / \{x_i\}$

3.2 Noise Filtering with the Aid of Unlabeled Samples

ENN detects the noises based on the labeled samples. Algorithm 1 shows that each labeled sample is checked (determining it is normal or noisy sample) through the voting of its nearest neighbors in the training set. This editing mechanism relies on the fact that the similar instances tend to be with the same class. Therefore, the instance is regarded as a noisy instance when it has a majority of its neighbors with the different class. Heuristically, the editing performance depends on the similarity degree between the instance and its nearest neighbors. For a given instance, its editing performance is expected to be better if the algorithm can find its neighbors which have higher similarity degree with it. Here, the intuitive idea is to extend the searching scope of neighbors from the training set to the whole data set which includes both the training set and unlabeled set. Considering that the labels of unlabeled data are not available, in order to utilize unlabeled data, the first phase therefore is to predict the labels for them. Then the second phase is to utilize this augmented set in data editing. Its two phases are briefly introduced below.

Phase 1: Predicting labels for the unlabeled data

In machine learning area, semi-supervised classification can provide a solution for this purpose. Traditional classifiers use only labeled data to train. However, semi-supervised classification uses unlabeled data together with the labeled data to train better classifiers. Semi-supervised classification use unlabeled data to either modify or reprioritize hypotheses obtained from labeled data alone. Semi-supervised classification can utilize unlabeled data in a variety of ways, however, we are only interested in the way which explicitly predicts labels for unlabeled data so that the training set is enlarged.

In general, there are two straightforward and popular semi-supervised methods following this way. They are self-training and co-training [4]. In self-training, a base learner is firstly trained on labeled set. Then, iteratively, it attempts to choose to label several examples that it is most confident of in the unlabeled set. After that it enlarges its labeled training set with these self-labeled examples. Co-training requires that features can be split into two sets; each sub-set is sufficient to train a good classifier; the two sets are conditionally independent given the class. Initially two separate classifiers are trained with the labeled data on the two sub-feature sets respectively. Each classifier then classifies the unlabeled data and aids the other classifier with the unlabeled samples they feel most confident. Each classifier is retrained with the additional training examples given by the other classifier. However, both of self-training and co-training present some drawbacks. Self-training only works for the classifiers which can measure the "confidence" of classification. Co-training requires the data which can be represented by two different set of features.

We have developed En-co-training algorithm [3] to deal with the problems.

Algorithm 2: The algorithm of ensemble-based co-training (En-co-training)

Given:

- L , A set of labeled training examples, consisting of M classes
- U , A set of unlabeled examples

Create a pool U' of examples by choosing u examples at random from U

Loop for k iterations:

(1) Use L to train a classifier h_1, h_2 and h_3 respectively

(2) For each class C , pick the n_c unlabeled data which classifier h_1, h_2 and h_3 agree with that its class label is C and add it to the collection of labeled examples

(3) Randomly choose $\sum_{c=1}^{M} n_c$ examples from U to replenish U'

As shown in Algorithm 2, it only augments the training data with the samples which have higher probability that their labels are correctly predicted. En-co-training is the combination of ensemble learning and co-training.

Phase 2: Utilizing the augmented data set in data editing

The nearest neighbors of a training instance obtained from a search of the training set and those from the whole data set may be different. The variation of nearest neighbors might lead to a change of editing result. Now the variant ENN in the case of using unlabeled data T_U to aid data editing on T are considered. It is shown in Algorithm 3.

Algorithm3:
The algorithm of ENN aided by unlabeled data

1. Let $T_e = T$

2. For each $x_i \in T_e$, do:

Discard x_i from T_e if it is misclassified using the k-NN rule with prototypes in $(T_e / \{x_i\}) \cup T_U$

4. EXPERIMENT

In our experiments, the activity data set published in [19] is used. This data set includes 25177 data samples and 9 activities. In our experiment, 4500 samples are used which includes the first 500 samples for each kind of activity.

This data set is divided into a training set and test set. ENN method works on the training set and generates the edited training set. Then, the test set is classified by the edited training set with the KNN algorithm. Classification accuracy is the measure to evaluate the performance of our method, where classification accuracy is:

$$\frac{\text{No. of correct classifications on testing instances}}{\text{No. of testing instances}}$$

To obtain the classification accuracy, the activity data set is processed as follows:

(1) Randomly partitioned into two parts: labeled set L and unlabeled set U .

(2) Ten trials derived from ten-fold cross-validation on L were used to evaluate the performance of data editing methods. At each trial, 90% of L , that is T , is used as training set. T was processed by the noise filtering method as mentioned above. The remaining 10% of L was used as test set to evaluate the performance of various processed sets of T .

(3) The average classification accuracy was obtained by averaging ten trial's accuracies.

(4) Considering that the partition of data set could influence this average classification accuracy, we execute the partition three times and get three classification accuracies (execute step 1-3 three times).

(5) Finally the reported accuracy is the further averaged value of these three values.

In this experiment, En-co-training's configuration is as follows. Three classifiers are generated by: 3-nearest neighbor, naive bayes and decision tree respectively. Initially, the size of U', u is equal to the size of training set, namely $u = |L|$. Iteration number k is 2. In noise filtering (ENN), 3NN is used and then 1NN is used to test the performance. An important parameter in the experiment is the ratio between labeled data to whole data, referred to {Labeled Ratio}. We adopt four different labeled ratios: 20%, 30%, 40% and 50%. The classification accuracy is shown in Table 1.

Table 1 shows that the classification accuracy of KNN is improved by using noise filtering technique. Based on our method, the performance is further improved. This observation shows that our method of using unlabeled data is effective.

Table 1. Activity recognition accuracies of each method under different labeled ratios

Labeled Ratio \ Method	KNN	ENN	Our Method
20%	83.2%	84.1%	84.9%
30%	85.1%	85.8%	87.2%
40%	87.6%	89.1%	90.0%
50%	88.9%	89.5%	91.7%

5. CONCLUSIONS AND FUTURE WORKS

We aim to utilize unlabeled data to improve the performance of activity recognition system. In our previous work, we have proposed a method which can improve the recognition accuracy through augmenting the number of training data by semi-supervised classification.

As an alternative method of previous one, we propose the other method to utilize unlabeled data for learning. Our method makes use of unlabeled data to remove the noises of labeled data. It can improve the recognition accuracy without increasing the number of training data. This merit makes it suitable for the algorithms (classifiers) which prefer small number of training data.

Experimental results show the effectiveness of our method. It should be noted although we present our method based on k nearest neighbor, it can also applied to other methods. Therefore, our future work is to evaluate our method through other popular machine learning methods.

6. ACKNOWLEDGEMENTS

This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Advancement) "(IITA-2008-(C1090-0801-0002))"

7. REFERENCES

[1] Stanford, V. 2002. Using pervasive computing to deliver elder care. IEEE Pervasive Computing 1, 1 (Jan-Mar. 2002), 10-13. DOI=http://doi.ieeecomputersociety.org/10.1109/MPRV.2002.993139

[2] Kidd, C. D., Orr, R., Abowd, G. D., Atkeson, C. G., Essa, I. A. 1999. The aware home: a living laboratory for ubiquitous computing research. In Proceedings of the Second International Workshop on Cooperative Buildings (October 01-02, 1999). 191-198.

[3] Guan, D. H., Yuan, W. W., Lee, Y. K., Lee, S. Y. 2007. Activity recognition based on semi-supervised learning. In Proceedings of 13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (Daegu, Korea, August 21-23, 2007). 469-475. DOI= http://doi.ieeecomputersociety.org/10.1109/RTCSA.2007.17

[4] Blum, A., Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In Proceedings of 11th Annual Conf. Computational Learning Theory (Madison, Wisconsin, July 24-26, 1998). 92-100. DOI= http://doi.acm.org/10.1145/279943.279962

[5] Intille, S. S., Davis, J., Bobick, A. 1997. Real-time closed-world tracking. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (San Juan, Puerto Rico, June 17-19, 1997). 697-703. DOI= http://doi.ieeecomputersociety.org/10.1109/CVPR.1997.609402

[6] Stillman, S., Tanawongsuwan, R., Essa, I. 1999. A system for tracking and recognizing multiple people with multiple cameras. In Proceedings of the Second International Conference on Audio-Vision-based Person Authentication (Washington, DC. April, 1999).

[7] Haritaoglu, I, Harwood, D., Davis, L. 1998. W4: Who, When, Where, What: A real time system for detecting and tracking people. In Proceedings of the Third International Conference on Automatic Face and Gesture (Nara, Japan, April 14-16, 1998). 222-227. DOI= http://doi.ieeecomputersociety.org/10.1109/AFGR.1998.670952

[8] Makikawa, M., Iizumi, H. 1995. Development of an ambulatory physical activity monitoring device and its application for categorization of actions in daily life. In Proceedings of the 8th World Congress on Medical Informatics: MEDINFO 95 (North-Holland, Amsterdam, 1995) 747-750.

[9] Aminian, K., Robert, P., Jequier, E., Schutz, Y. 1995. Estimation of speed and incline of walking using neural network. IEEE Transactions on Instrumentation and Measurement, 44, 3 (Jun 1995), 743-746. DOI=http://doi.ieeecomputersociety.org/10.1109/19.387322

[10] Bao, L. 2003. Physical activity recognition from acceleration data under seminaturalistic conditions. M.Eng thesis, EECS, Massachusetts Institute of Technology, 2003.

[11] Abowd, G. D. 2002. Director of the AwareHome initiative. Georgia Insitute of Technology, 2002.

[12] Barger, T., Alwan, M., Kell, S., Turner, B., Wood, S., Naidu, A. 2002. Objective remote assessment of activities of daily living: Analysis of meal preparation patterns. Poster presentation, Medical Automation Research Center, University of Virginia Health System, 2002.

[13] Mozer, M. 1998. The neural network house: an environment that adapts to its inhabitants. In Proceedings of the AAAI Spring Symposium on Intelligent Environments, Technical Report SS-98-02, AAAI Press, Menlo Park, CA, 1998 (110–114).

[14] Murphy, K. P. 2002. Dynamic Bayesian Networks: Representation, Inference and Learning. PhD thesis, University of California, Berkeley, 2002.

[15] Intille, S. S., Bobick, A. F. 2001. Recognizing planned, multi-person action. Computer Vision and Image Understanding, 81, 3, 414–445.

[16] Kautz, H., Etziono, O,. Fox, D., Weld, D. 2003. Foundations of assisted cognition systems. Technical report cse-02-ac-01, University of Washington, Department of Computer Science and Engineering, 2003.

[17] Wilson, D. L. 1972. Asymptotic properties of nearest neighbor rules using edited data, IEEE Transaction on Systems, Man, and Cybernetics, 2, 3 431-433.

[18] Guan, D. H., Yuan, W. W., Lee, Y. K., Lee, S. Y. 2008. In Proceedings of IEEE International Joint Conference on Neural Networks (HongKong, 2008). 1183-1187.

[19] K. Van Laerhoven and H.-W. Gellersen. 2004. Spine versus porcupine: a study in distributed wearable activity recognition. In Proceedings of the eighth International Symposium on Wearable Computers. 142-149.