

# Activity Recognition Based on Semi-supervised Learning

Donghai Guan, Weiwei Yuan, Young-Koo Lee\*, Andrey Gavrilov and Sungyoung Lee  
Department of Computer Engineering, Kyung Hee University, Korea  
{donghai, weiwei, avg, sylee}@oslab.khu.ac.kr; yklee@khu.ac.kr

## Abstract

*Activity recognition is a hot topic in context-aware computing. In activity recognition, machine learning techniques have been widely applied to learn the activity models from labeled activity samples. Since labeling samples requires human's efforts, most existing research in activity recognition focus on refining learning techniques to utilize the costly labeled samples as effectively as possible. However, few of them consider using the costless unlabeled samples to boost learning performance. In this work, we propose a novel semi-supervised learning algorithm named En-Co-training to make use of the unlabeled samples. Our algorithm extends the co-training paradigm by using ensemble method. Experimental results show that En-Co-training is able to utilize the available unlabeled samples to enhance the performance of activity learning with a limited number of labeled samples.*

## 1. Introduction

Activity recognition has gained a lot of interest in recent years due to its potential and usefulness for context-aware computing such as aged care monitoring [1] and smart homes [2]. Basically, 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 instance, in smart home environments for aged care monitoring [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.

There are several ways to acquire human' activity using sensor systems. These methods include, but are not limited to: (1) remotely observe the scene using audio, visual, electromagnetic field, or other sensors

and interpret the signal readings [3][4][5], (2) attach sensors to the body and interpret the signal readings [6][7][8], (3) attach sensors to objects and devices in the environment and interpret the sensor readings [9][10].

For different activity recognition systems, they may use various approaches to acquire activity information such as those methods mentioned above. However, machine learning is always a key aspect of them. For a system to automatically infer what activity is being performed, it must have a detailed model of the activity. Currently a variety of machine learning methods have been proposed for activity recognition, such as neural networks [11], dynamic Bayesian networks [12], naïve Bayesian networks [13], hierarchical hidden semi-Markov models [14], nearest neighbors [8], decision tree [8] and so on. One common characteristic of these methods is the requirement of labeled activity samples for training purpose. In addition, to achieve good recognition performance, a large amount of labeled samples are needed. However, in real activity recognition systems, labeled samples are usually difficult or expensive to obtain as they require the efforts of human annotators. In such cases, how to achieve a good learning model as best as possible is a crucial issue.

Semi-supervised learning [15][16][17][18] is a hot topic aiming to address this issue. In addition to labeled samples, it exploits unlabeled ones to improve learning performance. It should be noted that unlabeled activity samples are easy to obtain from routine experiments since they do not require human's annotation efforts. Co-training [19] is an attractive semi-supervised learning paradigm, which trains two classifiers through letting them label the unlabeled examples for each other. In co-training the data should be described by two sufficient and redundant attribute subsets, each of which is sufficient for learning and independent to the other given class label.

Although co-training has already been successfully applied to some fields [20][21][22], the requirement on

---

\* Corresponding author.

two sufficient and redundant attribute subsets might be too strong to be met in many activity recognition systems.

In this work, a new co-training style algorithm named En-Co-training, i.e. ensemble method based co-training, is proposed. It extends the co-training paradigm by incorporating ensemble method. Superior to co-training, En-Co-training has no requirement that the data should be described by two sufficient and redundant attribute subsets. Experimental results show that when activity data do not meet the requirement of Co-training, our proposed En-Co-training algorithm is able to utilize the available unlabeled samples to enhance the performance of activity learning on a limited number of labeled samples.

The rest of the paper is organized as follows. Section 2 describes semi-supervised learning, especially co-training approach. Section 3 introduces our proposed En-Co-training algorithm. Section 4 is the experiment part. Finally, conclusions are presented in Section 5.

## 2. Semi-supervised learning

Supervised learning utilizes training data to learn the hypothesis or models. Usually, all training data should be labeled before learning. Recently, much researches show that unlabeled data are also useful for learning. Semi-supervised learning is such a method which could make use of unlabeled data to improve the performance of learning. It should be noted that in real applications, labeled data are usually difficult or expensive to obtain as they require the efforts of human annotators. However, it is much easier to get unlabeled data since they do not need human's labeling effort. The ability to utilize the costless unlabeled data makes semi-supervised learning to be one of the hot topics in machine learning area.

Many current semi-supervised learning algorithms use a generative model for the classifier and employ Expectation-Maximization (EM) to model the label estimation or parameter estimation process [15]. For example, mixtures of Gaussians [16], mixture of experts [17], and naïve Bayes [18] have been respectively used as the generative model, while EM is used to combine labeled and unlabeled data for classification.

A preeminent work in semi-supervised learning methods is the co-training paradigm proposed by Blum and Mitchell [19]. In co-training, two classifiers are trained on two sufficient and redundant sets of attributes respectively. Each classifier labels several unlabeled examples whose labels are most confidently predicted from its point of view. These newly labeled

examples are used to augment the labeled training set of the other classifier. Then each classifier is refined with its augmented labeled training set. They showed that any weak hypothesis could be boosted from the unlabeled data if the data meet the class-conditional independent requirement and the target concept is learnable with random classification noise. Dasgupta [23] derived a generalization error bound for the co-trained classifier, which indicates that when the requirement on the existence of sufficient and redundant attribute is met, the co-trained classifiers can make fewer generalization errors by maximizing their agreements over the unlabeled data. Co-training algorithm is given in Table 1.

**Table 1.** Co-training algorithm.

<p>Given:</p> <ul style="list-style-type: none"> <li>● <math>L</math>, A set of labeled training examples <math>x</math>, consists of <math>M</math> classes</li> <li>● <math>U</math>, A set of unlabeled examples</li> <li>● <math>x</math> is described by two different views (<math>x_1</math> and <math>x_2</math>)</li> </ul> <p>Create a pool <math>U'</math> of examples by choosing <math>u</math> examples at random from <math>U</math></p> <p>Loop for <math>k</math> iterations:</p> <ol style="list-style-type: none"> <li>(1) Use <math>L</math> to train a classifier <math>h_1</math> that considers only the <math>x_1</math> portion of <math>x</math></li> <li>(2) Use <math>L</math> to train a classifier <math>h_2</math> that considers only the <math>x_2</math> portion of <math>x</math></li> <li>(3) For each class <math>C</math>, pick the <math>n_c</math> unlabeled data about which classifier <math>h_1</math> is most confident that its class label is <math>C</math> and add it to the collection of labeled examples</li> <li>(4) For each class <math>C</math>, pick the <math>n_c</math> unlabeled data about which classifier <math>h_2</math> is most confident that its class label is <math>C</math> and add it to the collection of labeled examples</li> <li>(5) Randomly choose <math>2 \times \sum_{c=1}^M n_c</math> examples from <math>U</math> to replenish <math>U'</math></li> </ol> <ul style="list-style-type: none"> <li>● Output: Two classifiers, <math>h_1</math> and <math>h_2</math>, that predict class labels for new examples</li> </ul>
--

### 3. En-Co-training

Co-training is a good approach to utilize unlabeled data to improve the learning performance. However, it has some intrinsic limitations:

1) Requirement on the data samples. As we present in Section 2, the requirement to use Co-training is that data should be described by two sufficient and redundant attribute subsets, each of which is sufficient for learning and independent to the other given class label. Due to this requirement, Co-training is only successfully used in few applications, such as text classification and most other applications could not meet this requirement.

2) Let us go to the 2<sup>nd</sup> and 3<sup>rd</sup> steps of Co-training algorithm shown in Table 1. In order to determine which example in  $U$  should be labeled, the confidence of the labeling of each classifier must be explicitly measured. Sometimes, such a measuring process is quite time-consuming [24]. In fact, the confidence of the labeling of each classifier is also needed to combine the results of two classifiers.

As explained later, our activity data can not meet the requirement of Co-training since they cannot be described by two sufficient and redundant attribute subsets. Hence, in order to utilize the theory of Co-training, a new type of Co-training algorithm is needed. In this work, we propose a novel Co-training algorithm, named En-Co-training, which extends the basic Co-training algorithm by using ensemble method. Our En-Co-training algorithm is shown in Table 2.

We propose En-Co-training to deal with the limitations of Co-training mentioned above.

First of all, at the first step of Co-training, two different portions are used to train different classifiers. To guarantee the performance of these two classifiers, each portion must be good enough for training. This is the reason why the data used for Co-training need to be described by two sufficient and redundant attribute subsets, each of which is sufficient for learning and independent to the other. Different with that in Co-training, the first step of En-Co-training uses whole data  $L$  to train different classifiers. Hence, the requirement on the data is avoided. It should be noted that Co-training requires two diverse classifiers. Standard Co-training utilizes different portions of data sample to train the same learning method to get different classifiers. However, in En-Co-training, the whole data is used for all the classifiers to avoid the requirement on data. In order to guarantee the diversity of these classifiers, we use different learning methods. For example, if one learning method is decision tree,

then another classifier must use different methods, such as neural network or naïve Bayes.

Secondly, different with Co-training, there are three classifiers in En-Co-training. Actually, three classifiers are used to deal with the second limitation of Co-training. Through using three classifiers, the explicit measure on the confidence of the labeling of each classifier is not required. In detail, as shown in the 2<sup>nd</sup> step of En-Co-training, we regard the label for the unlabeled data is confident if the three classifiers give the same classification result. Also, through using three classifiers, the predictions of them can be easily combined through ensemble method, such as majority voting.

**Table 2.** En-Co-training algorithm.

Given:
● $L$ , A set of labeled training examples $x$ , consists 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 $U'$
● Output: Three classifiers, $h_1$ , $h_2$ and $h_3$ that predict class labels for new examples. These predictions can be combined by majority voting.

### 4. En-Co-training on activity recognition

For our experiments we use a dataset published on [25]. It consists of 10 basic activities, namely Lying, Kneeling, Sitting, Standing, Walking, Running, Climbing Stairs, Descending Stairs, Bicycling and Jumping. The activities were recorded by 40 accelerometers strapped loosely to common trousers, 20 sensors per leg, starting from the ankle to the hip.

The original data set includes 25177 data samples and 9 activities (1 activity is missing in the dataset). For each kind of activity, we choose the first 500

samples. Hence totally 4500 data samples are used in our experiment.

#### 4.1. Test whether co-training can be used

In this part, we will show whether Co-training can be used on our activity data set. As we mentioned above, the requirement to use Co-training is that data should be described by two sufficient and redundant attribute subsets, each of which is sufficient for learning and independent to the other.

Three various supervised learning without co-training were performed with aim of determining how redundantly sufficient of our activity data set. Considering that the activity information is gathered from 40 sensors (20 sensors per leg). Hence an intuitive split of the data features is one portion with first twenty features (one leg) and the other portion with the other twenty features (the other leg). In addition, we also try random separation of features to check the redundancy.

Three various supervised learning methods are used. They are decision tree, Naïve Bayes and K-nearest neighbors (K=3). For each classifier, this test runs ten times. At each time, 20 percent of whole data were randomly selected as test data. For the other 80 percent, 5 percent were randomly selected as training data. Obviously, there is no overlap between test data and training data. The test results are shown in Table 3.

Based on the experimental results shown in Table 3, we can see the activity data used in our work can not be separated into two sufficient and redundant attribute subsets. Hence Co-training cannot be adopted to utilize the unlabeled data.

**Table 3.** Using decision tree (DT), k-nearest neighbors (KNN) and naïve bayes (NB) to test features' redundancy of activity samples.

DT	ALL	Natural Feature Split		Random Separation	
		1-20	21-40	Halves1	Halves2
1	0.779	0.769	0.143	0.222	0.092
2	0.816	0.778	0.177	0.134	0.133
3	0.806	0.811	0.181	0.104	0.196
4	0.766	0.782	0.206	0.112	0.116
5	0.793	0.751	0.328	0.017	0.179
6	0.748	0.781	0.132	0.128	0.183
7	0.778	0.789	0.109	0.137	0.110
8	0.768	0.803	0.121	0.117	0.181
9	0.797	0.784	0.098	0.276	0.081
10	0.751	0.766	0.137	0.080	0.170
Ave.	0.780	0.781	0.163	0.133	0.144

KNN	ALL	Natural Feature Split		Random Separation	
		1-20	21-40	Halves1	Halves2
1	0.777	0.787	0.264	0.106	0.112
2	0.804	0.826	0.286	0.342	0.188
3	0.781	0.798	0.272	0.112	0.218
4	0.783	0.789	0.253	0.073	0.208
5	0.801	0.817	0.287	0.119	0.343
6	0.802	0.831	0.272	0.211	0.107
7	0.793	0.818	0.277	0.160	0.136
8	0.821	0.807	0.268	0.196	0.201
9	0.809	0.823	0.229	0.134	0.230
10	0.788	0.839	0.230	0.189	0.159
Ave.	0.796	0.813	0.264	0.164	0.190

NB	ALL	Natural Feature Split		Random Separation	
		1-20	21-40	Halves1	Halves2
1	0.797	0.790	0.102	0.102	0.117
2	0.851	0.824	0.108	0.119	0.107
3	0.829	0.818	0.137	0.121	0.124
4	0.810	0.782	0.129	0.127	0.116
5	0.822	0.822	0.104	0.098	0.098
6	0.854	0.824	0.114	0.119	0.101
7	0.839	0.818	0.151	0.112	0.114
8	0.822	0.811	0.113	0.140	0.126
9	0.844	0.842	0.108	0.114	0.128
10	0.841	0.837	0.127	0.117	0.131
Ave.	0.831	0.817	0.119	0.117	0.116

#### 4.1. Performance of En-Co-training

We propose En-Co-training to deal with the limitation of Co-training. To test whether En-Co-training could make use of unlabeled data to improve the performance of learning, the experiments are executed as follows:

For the activity data set, 70 percent are kept as test examples while the rest are used as the pool of training examples, i.e.,  $L \cup U$ . In each pool,  $L$  and  $U$  are partitioned under different unlabeled rates including 90 percent, 80 percent, 70 percent, 60 percent and 50 percent. For instance, assuming a pool contains 1000 examples, when the unlabeled rate is 80 percent, 200 examples are put into  $L$  with their labels while the remaining 800 examples are put into  $U$  without their labels.

J4.8 decision trees, Naïve Bayes classifier and K-nearest neighbors (K=3) are used in the experiments. Under each unlabeled rate, ten independent runs with different random partitions of  $L$  and  $U$  are performed. The performance of En-Co-training is compared with these three classifiers and their ensemble. Here ensemble method is majority voting. The parameters

used in the experiments are: iteration number  $k$  is 20,  $n_c$  is 4,  $u$  is 270. The results are shown in Table 4,5,6,7 and 8. In these tables, “Last” in En-Co-training part represents the error rate at the 20<sup>th</sup> iteration. At each run, the best result among three various classifiers and their ensemble is regarded as baseline, which is shown in bold face. We will compare the performance of “Last” with the baseline value. Improvement is also shown in the tables.

**Table 4.** Error rates when unlabel ratio is 90 percent.

Runs	DT	KNN	NB	Ensem.	En-Co-training	
					Last	Imp.
1	0.242	0.241	<b>0.175</b>	0.193	0.157	10.3%
2	0.211	0.231	0.162	<b>0.160</b>	0.132	17.5%
3	0.220	0.208	0.188	<b>0.175</b>	0.137	21.7%
4	0.241	0.223	0.163	<b>0.162</b>	0.136	16.0%
5	0.185	0.207	0.159	<b>0.141</b>	0.118	16.3%
6	0.165	0.208	0.160	<b>0.137</b>	0.135	1.46%
7	0.229	0.199	0.204	<b>0.186</b>	0.142	23.7%
8	0.259	0.239	<b>0.170</b>	0.176	0.157	7.65%
9	0.255	0.216	<b>0.166</b>	0.204	0.121	27.1%
10	0.206	0.201	0.194	<b>0.172</b>	0.167	2.91%
Ave.	0.221	0.217	0.174	0.170	0.140	14.5%

**Table 5.** Error rates when unlabel ratio is 80 percent.

Runs	DT	KNN	NB	Ensem.	En-Co-Training	
					Last	Imp.
1	0.159	0.163	0.162	<b>0.126</b>	0.117	7.14%
2	0.193	0.180	0.161	<b>0.146</b>	0.138	5.48%
3	0.203	0.176	0.163	<b>0.144</b>	0.129	10.4%
4	0.220	0.177	0.170	<b>0.163</b>	0.153	6.13%
5	0.189	0.178	0.174	<b>0.148</b>	0.125	15.5%
6	0.198	0.156	0.157	<b>0.122</b>	0.105	13.9%
7	0.158	0.169	0.168	<b>0.126</b>	0.113	10.3%
8	0.159	0.163	0.162	<b>0.126</b>	0.117	7.14%
9	0.193	0.180	0.161	<b>0.146</b>	0.138	5.48%
10	0.203	0.176	0.163	<b>0.144</b>	0.129	10.4%
Ave.	0.188	0.172	0.164	0.139	0.126	9.19%

**Table 6.** Error rates when unlabel ratio is 70 percent.

Runs	DT	KNN	NB	Ensem.	En-Co-Training	
					Last	Imp.
1	0.176	0.157	0.168	<b>0.136</b>	0.131	3.68%
2	0.153	0.139	0.136	<b>0.122</b>	0.105	13.9%
3	0.183	0.156	0.165	<b>0.139</b>	0.127	8.63%
4	0.158	0.148	0.152	<b>0.119</b>	0.123	-3.36%
5	0.156	0.157	0.163	<b>0.129</b>	0.122	5.43%
6	0.148	0.152	0.148	<b>0.115</b>	0.113	1.74%
7	0.160	0.146	0.137	<b>0.135</b>	0.118	12.6%
8	0.160	0.156	0.150	<b>0.121</b>	0.117	3.31%
9	0.185	0.145	0.142	<b>0.116</b>	0.099	14.7%
10	0.157	0.138	0.152	<b>0.114</b>	0.113	0.87%
Ave.	0.164	0.149	0.151	0.125	0.117	6.40%

**Table 7.** Error rates when unlabel ratio is 60 percent.

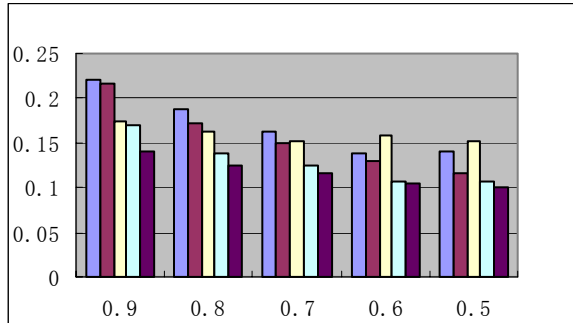
Run	DT	KNN	NB	Ensem.	En-Co-Training	
					Last	Imp.
1	0.138	0.126	0.163	<b>0.111</b>	0.107	3.60%
2	0.140	0.127	0.140	<b>0.103</b>	0.100	2.91%
3	0.137	0.118	0.171	<b>0.103</b>	0.102	0.97%
4	0.145	0.135	0.166	<b>0.114</b>	0.121	-6.14%
5	0.143	0.118	0.167	<b>0.095</b>	0.094	0.09%
6	0.128	0.157	0.167	<b>0.118</b>	0.103	12.7%
7	0.141	0.126	0.143	<b>0.106</b>	0.106	0
8	0.149	0.129	0.156	<b>0.111</b>	0.102	8.11%
9	0.141	0.135	0.157	<b>0.112</b>	0.107	4.46%
10	0.122	0.128	0.148	<b>0.097</b>	0.093	4.12%
Ave.	0.138	0.130	0.158	0.107	0.104	3.08%

**Table 8.** Error rates when unlabel ratio is 50 percent.

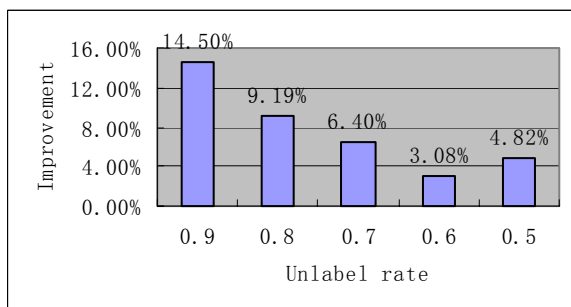
Runs	DT	KNN	NB	Ensem.	En-Co-Training	
					Last	Imp.
1	0.133	0.107	0.130	<b>0.090</b>	0.085	5.55%
2	0.119	0.122	0.166	<b>0.104</b>	0.102	1.92%
3	0.156	0.117	0.159	<b>0.114</b>	0.102	10.5%
4	0.141	0.118	0.150	<b>0.116</b>	0.110	5.17%
5	0.149	0.123	0.153	<b>0.117</b>	0.112	4.27%
6	0.149	0.117	0.154	<b>0.100</b>	0.099	1.00%
7	0.156	0.117	0.159	<b>0.114</b>	0.102	10.5%
8	0.134	0.114	0.156	<b>0.104</b>	0.102	1.92%
9	0.120	0.103	0.134	<b>0.092</b>	0.090	2.17%
10	0.141	0.118	0.150	<b>0.116</b>	0.110	5.17%
Ave.	0.140	0.116	0.151	0.107	0.101	4.82%

Above experiment results show that our propose En-Co-training could utilize unlabeled data to improve the learning performance with different unlabel rate. In detail, when unlabel rate is 90 percent (135 samples were used as labeled data), average improvement is 14.5 percent. When unlabel rate is 80 percent (270 samples were used as labeled data), average improvement is 9.19 percent. When unlabel rate is 70 percent (405 samples were used as labeled data), average improvement is 6.4 percent. When unlabel rate is 60 percent (540 samples were used as labeled data), average improvement is 3.08 percent. When unlabel rate is 50 percent (675 samples were used as labeled data), average improvement is 4.82 percent. Fig. 1 shows the average classification error rate for three separate classifiers, ensemble and En-Co-training at different unlabel rate.

Performance improvement through using En-Co-training is shown in Fig. 2. We can see that improvement is greatest when unlabel rate is 0.9. It means when the number labeled data is smaller, our En-Co-training could make more improvement.



**Fig. 1.** Average classification error rate at different unlabel rates (Different colors represent decision tree, knn, naïve bayes, ensemble and En-Co-Training respectively).



**Fig. 2.** Average performance improvement at different unlabel rates.

## 5. Conclusions

In this work, we propose a novel semi-supervised learning algorithm for activity recognition. Different with most existing work, our proposed method could improve the learning performance through make use of unlabeled data. It should be noted that in activity recognition systems, usually the number of labeled samples is limited because they require the efforts of human annotators. Different with labeled data, the number of unlabelled data is huge because they are easy to get without human's labeling effort. Hence, our work is very useful for real activity recognition applications. Experiment results show that our proposed method could effectively improve learning performance.

## 6. Acknowledgements

This research was supported by the MIC (Ministry of Information and Communication), Korea, Under the ITFSIP (IT Foreign Specialist Inviting Program) supervised by the IITA (Institute of Information Technology Advancement).

## 7. References

- [1] V. Stanford, "Using pervasive computing to deliver elder care", *IEEE Pervasive Computing*, 2002, pp. 10-13.
- [2] C.D. Kidd, R. Orr, G.D. Abowd, C.G. Atkeson and I.A. Essa, "The aware home: A living laboratory for ubiquitous computing research", *Proceedings of the Second International Workshop on Cooperative Buildings*, 1999, pp. 191-198.
- [3] S.S. Intille, J. Davis, and A. Bobick, "Real-time closed-world tracking", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1997, pp. 697-703.
- [4] S. Stillman, R. Tanawongsuwan, and I. Essa, "A system for tracking and recognizing multiple people with multiple cameras", *Proceedings of the Second International Conference on Audio-Vision-based Person Authentication*, 1999.
- [5] I. Haritaoglu, D. Harwood, and L. Davis, "W4: Who, When, Where, What: A real time system for detecting and tracking people", *Proceedings of the Third International Conference on Automatic Face and Gesture*, Nara, 1998, pp. 222-227.
- [6] M. Makikawa and H. Iizumi, "Development of an ambulatory physical activity monitoring device and its application for categorization of actions in daily life. *MEDINFO*, 1995, pp. 747-750.
- [7] K. Aminian, P. Robert, E. Jequier, and Y. Schutz, "Estimation of speed and incline of walking using neural network", *IEEE Transactions on Instrumentation and Measurement*, 1995, pp. 743-746.
- [8] L. Bao, "Physical activity recognition from acceleration data under seminaturalistic conditions", *M.Eng thesis*, EECS, Massachusetts Institute of Technology, 2003.
- [9] G.D. Abowd, "Director of the AwareHome initiative", Georgia Institute of Technology, November 2002.
- [10] T. Barger, M. Alwan, S. Kell, B. Turner, S. Wood, and A. Naidu, "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.
- [11] M. Mozer, "The neural network house: an environment that adapts to its inhabitants", *Proceedings of the AAAI Spring Symposium on Intelligent Environments*, Technical Report SS-98-02, AAAI Press, Menlo Park, CA, 1998, pp. 110-114.
- [12] K. P. Murphy, "Dynamic Bayesian Networks: Representation, Inference and Learning", PhD thesis, University of California, Berkeley, 2002.

- [13] S.S. Intille and A.F. Bobick, "Recognizing planned, multi-person action", *Computer Vision and Image Understanding*, 2001, pp. 414-445.
- [14] H. Kautz, O. Etzioni, D. Fox, and D. Weld, "Foundations of assisted cognition systems", technical report cse-02-ac-01, University of Washington, Department of Computer Science and Engineering, 2003.
- [15] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood for incomplete data via the EM algorithm," *J. Royal Statistical Soc.*, Series B. 1977, pp. 1-38.
- [16] B. Shahshahani and D. Landgrebe, "The Effect of Unlabeled Samples in Reducing the Small Sample Size Problem and Mitigating the Hughes Phenomenon," *IEEE Trans. Geoscience and Remote Sensing*, 1994, pp. 1087-1095.
- [17] D.J. Miller and H.S. Uyar, "A Mixture of Experts Classifier with Learning Based on Both Labelled and Unlabelled Data," *Advances in Neural Information Processing Systems*, 1997, pp. 571-577.
- [18] K. Nigam, A.K. McCallum, S. Thrun, and T. Mitchell, "Text Classification from Labeled and Unlabeled Documents Using EM," *Machine Learning*, 2000, pp. 103-134.
- [19] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," *Proc. 11<sup>th</sup> Ann. Conf. Computational Learning Theory*, 1998, pp. 92-100.
- [20] R. Hwa, M. Osborne, A. Sarkar, and M. Steedman, "Corrected Co- Training for Statistical Parsers," *Working Notes of the ICML'03 Workshop Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, 2003.
- [21] D. Pierce and C. Cardie, "Limitations of Co-Training for Natural Language Learning from Large Data Sets," *Proc. 2001 Conf. Empirical Methods in Natural Language Processing*, 2001, pp. 1-9.
- [22] A. Sarkar, "Applying Co-Training Methods to Statistical Parsing," *Proc. Second Ann. Meeting of the North Am. Chapter of the Assoc. for Computational Linguistics*, 2001, pp. 95-102.
- [23] S. Dasgupta, M. Littman, and D. McAllester, "PAC Generalization Bounds for Co-Training," *Advances in Neural Information Processing Systems*, 2002, pp. 375-382.
- [24] S. Goldman and Y. Zhou, "Enhancing Supervised Learning with Unlabeled Data," *Proc. 17<sup>th</sup> Int'l Conf. Machine Learning*, 2000, pp. 327-334.
- [25] K. Van Laerhoven and H.-W. Gellersen, "Spine versus Porcupine: a Study in Distributed Wearable Activity Recognition", *Proceedings of the eighth International Symposium on Wearable Computers*, 2004, 142-149.