Cross-People Mobile-Phone Based Activity Recognition

Zhongtang ZHAO[1,3],Yiqiang CHEN[1,2],Junfa LIU[1,2] , Zhiqi SHEN[2] , Mingjie LIU[1,3]
[1]Pervasive Computing Center, Institute of Computing Technology, CAS, Beijing 100190,China
[2]Nanyang Technological University, Singapore 088891
[3]Graduate University of the Chinese Academy of Sciences, Beijing 100049, China
{zhaozhongtang,yqchen,liujunfa,liumingjie}@ict.ac.cn ; {zqshen}@ntu.edu.sg

Abstract

Activity recognition using mobile phones has great potential in many applications including mobile healthcare. In order to let a person easily know whether he is in strict compliance with the doctor's exercise prescription and adjust his exercise amount accordingly, we can use a smart-phone based activity reporting system to accurately recognize a range of daily activities and report the duration of each activity. A triaxial accelerometer embedded in the smart phone is used for the classification of several activities, such as staying still, walking, running, and going upstairs and downstairs. The model learnt from a specific person often cannot yield accurate results when used on a different person. To solve the cross-people activity recognition problem, we propose an algorithm known as TransEMDT (Transfer learning EMbedded Decision Tree) that integrates a decision tree and the k-means clustering algorithm for personalized activity-recognition model adaptation. Tested on a real-world data set, the results show that our algorithm outperforms several traditional baseline algorithms.

1 Introduction

1.1 Background

Exercising is important for keeping fit and the amount of exercise required by each person can be quite different. To help people keep fit and improve health, doctors often prescribe exercises, such as walking, running, climbing upstairs, going downstairs, etc., to patients. The most important metric that measures the amount of exercise is time. In order to let a person know if he is in strict compliance with the doctor's prescription and adjust his exercise amount accordingly, the time he has spent in an exercise should be made easily accessible to him. This is a crucial factor to enable people to improve their quality of exercise. A population of people with good health can save the government's cost in medical care.

There are some devices that can evaluate the amount of exercises, such as the one presented in [Consolvo *et al.*, 2006] that can quantify a user's activity in terms of "number of steps", which is ideal for measuring the amount of exercise in walking and running. Polar [Nachman *et al.*, 2010] uses heart rates to quantify activities in terms of calories, which is suitable for cardio training applications. However, these solutions only give some statistical figures to users and do not give the details on each kind of exercises. A heart rate belt has to be put on a user's chest when the Polar system is used, which is not convenient to use for all users.

In this paper, we present a smart-phone based portable activity reporting system that uses a triaxial accelerometer to accurately recognize a range of daily activities, such as staying still, walking, running, and going upstairs and downstairs, and report the duration of each activity. To solve the cross-people activity recognition problem, that is, the model learnt from a specific person often can not yield accurate results when used on a different person, a transfer-learning technique that integrates decision trees and k-means clustering algorithm is embedded in a phone-based activity reporting system. An activity recognition model is trained off-line and is installed on a smart phone for one user. When another person uses the same phone, the system can collect his or her activity samples and need not label the data for the new user, which avoids expensive data labeling costs. The TransEMDT system uses the unlabeled samples to adapt the activity recognition model and construct a personalized model for the new user.

The contributions of our work can be described as follows: 1) We propose a novel fast and simple adaptive model to be *embedded* in mobile phones that have the limited computing resource, storage and power. 2) We experimentally evaluate the effectiveness of the model by testing how to minimize the labeled training sample set and reduce the iteration times, while maintaining accuracy.

1.2 Activity Reporting System

Overall System

Figure 1: Activity Reporting System

Previous research has shown that even a limited amount of physical activity can be beneficial for people's health, if done regularly and over a long time. Especially for the elderly, a proper amount of exercise is very important to prevent and control diseases. To help monitor the daily activities of the elderly, we have built a motion recognition system named *Activity Reporting System*, which is a software system that embeds on a smart phone. As shown in Figure 1, with the smart phone, the activities of a user change the readings of an accelerometer. The artificial intelligence (AI) module of the activity reporting system collects the readings of the accelerometer, and recognizes the pattern of each activity using a classifier that has been trained off-line and measures the total time of each activity. Then, a user such as an elderly person can easily know whether he reaches his prescribed amount of exercise.

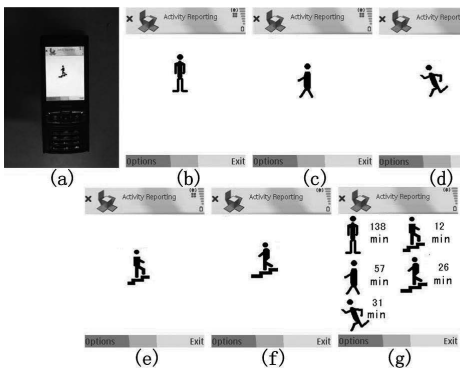

Figure 2: the screenshots of activity reporting system. (a)the Activity Reporting System is running on the phone.(b)$\sim$(f) the activities of the user, they are stationary, walking, running, going upstairs, going downstairs respectively.(g)the accumulated duration of every activity.

The activity reporting system can be installed on a smart phone, which the user can hold in hand, put in a pocket, wear on the waistband, or hang in front of the chest. As shown in Figure 2, while the user is doing exercise, the activity states can be displayed on the screen. When the user wants to know the statistical information of each activity, he can change to the corresponding interface via shortcuts on the screen.

Embedded AI Module

It is hard to generate an one-size-fits-all model that can classify all users' motion activities with high accuracy. When a training sample set contains data from only a single person, the model learnt from them often performs well for this person. However, often, such a model does not yield accurate results when used on a different person due to data distribution changes.

In this paper, to solve the above problem, we propose an embedded transfer learning technique [Pan and Yang, 2010] for the problem. Our solution integrates the k-means clustering algorithm [Hamerly and Elkan, 2002] and the decision tree algorithm [Ravi *et al.*, 2005] together. We call our system TransEMDT, which stands for *transfer learning embedding using decision trees*. As shown in Figure 3, firstly,

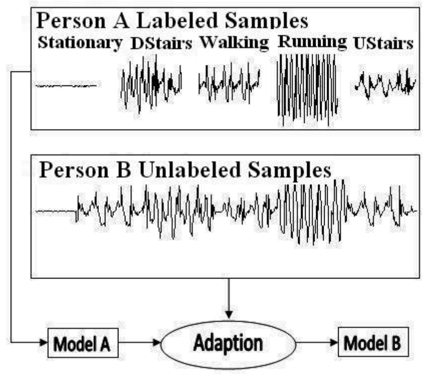

Figure 3: Cross-People Motion Activity Recognition

we build Model A with the labeled samples of Person A. Secondly, we classify the unlabeled samples of Person B to Model A. These data are used to adapt Model A to yield a new personalized model, Model B, for person B. Experiments using real-world samples show that the TransEMDT algorithm performed well using the generated personalized model.

The rest of the paper is as follows. In Section 2, related works are reviewed. In Section 3 TransEMDT is presented in detail. In Section 4, experiments on TransEMDT is given. Section 5 concludes the paper.

2 Related Work

Some of the existing literature has explored activity recognition based on accelerometers. In the work of [Ravi *et al.*, 2005], [Ward *et al.*, 2011], the authors used multiple accelerometers to classify different activities. Chen [Chen *et al.*, 2010] used a smart phone to detect six activities in order to find the state change point. These models can achieve high recognition accuracies because their testing and training samples are from the same batch of samples and follow the same data distribution. All these above literature did not consider the cross-people activity recognition problem.

Some literature has mentioned the cross-people problem. For example, in the work of [Albinali *et al.*, 2010], the authors used multiple sensors to detect activity types. When subject-specific training data was used, the average accuracy is 49%. When the leave-one-subject-out validation was used, the average accuracy dropped 26%.

Researchers have applied transfer learning to activity recognition when the activity labels are different. For example, Zheng et al. [Zheng *et al.*, 2009] learnt a similarity function between the activities in the source domain and the target domain via Web search. Based on the learnt similarity measures, they transferred the data from the source domain to the target domain that have different label space. As for our case, due to the source domain has the same label space as the target domain, we focus on quick adaptation and power efficient algorithm which can be implemented on mobile phone.

In literature [Stikic *et al.*, 2008], the authors used collaborative learning to introduce the unlabeled samples into the training data set and achieved better models. But their algorithm employs two or more learners that are learnt from different feature sets separately. They used voting to select labels for the new samples and then put the labeled samples to the training set to train new learners, which is complicated and not power efficient, which may not be suitable for smart phone based applications.

Lee [Lee and Giraud-Carrier, 2007] proposed a Transfer in Decision Trees (TDT) algorithm. TDT assumes that the set of attributes of the source task is a proper subset of the set of attributes of the target task. It learns a partial decision tree model from the source task and then transformed it as required by the training data in the target task. Torrey et al. [Torrey *et al.*, 2005] presented a method for transferring knowledge learnt in one task to a related task by reinforcement learning. They needed a human teacher in order to provide a mapping from the source task to the target task to guide this knowledge transfer. In contrast, our method is an automatic method without any manual input.

3 The TransEMDT Algorithm

In this section, we present the embedded AI component in detail. Our algorithm runs in the following high-level steps. We first generate a decision tree DT for person A. This decision tree is binary, in that every non-terminal node A corresponds to a binary split on a continuous value θ_A. We then use the decision tree DT to classify data for person B. We then use the classified data as the initial cluster centers for person B and start an one-step K-means algorithm. For this case, each leaf node is a class. This step redistributes the target samples in the leaf nodes of the decision tree DT. We update the parameters of the DT using the high confident samples on each leaf node through a bottom-up process, we select k-nearest samples to the centers as the high confident samples. Finally, we repeat Step 2 to Step 4 in a loop, until the algorithm converges. Then, the result is a DT' for person B.

3.1 Overview of TransEMDT

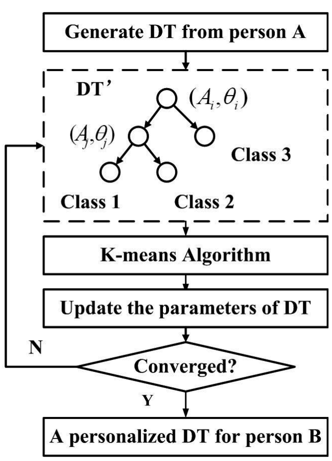

Figure 4: the TransEMDT Framework

As shown in Figure 4, the TransEMDT model consists of three layers, the DT layer (Decision Tree), the k-means clustering algorithm layer and the layer to updating the binary-split parameters of DT. In the learning process for person B, it is the binary-split parameters that are to be adjusted according to the target-domain data.

3.2 The DT Layer

The DT is a binary decision tree. Each non-terminal node has its own decision conditions (A_i,θ_i), A_i represents the decision attribute, θ_i represents the decision threshold, such that if the value of A_i is less than this value, the data route to the left subtree; otherwise, the data route to the right subtree. The initial decision tree DT is learnt off-line using person A's data. In this layer, we can not only construct a classifier, but also find the attributes that can distinguish one class from another.

Finding the Distinguishing Attributes

It can be seen from Figure 4 that the unlabeled samples will be assigned corresponding labels after the DT model classifies them. At the same time, the attributes on the path from leaf node to root node are the best attributes to distinguish one class from another. For example, Class 1 has the different attributes set as compared to Class 3. Although Class 1 and Class 2 have the same attributes set, they can be distinguished from each other using the attribute nearest to the leaf node.

For a sample x=$\{x_1, x_2, \ldots\}$, A_j is the j^{th} attribute of *x*, x_j is its value. For the i^{th} leaf node, a vector ω_i was set and its j^{th} component can be determined as follows:

$$\omega_{ij} = \begin{cases} 1, & A_j \in P_i \\ 0, & A_j \notin P_i \end{cases} \tag{1}$$

where P_i is the path from the j^{th} leaf node to the root node.

We take the samples in each terminal node and find its cluster center. We then find the distance from each instance x to the i^{th} cluster center μ_i using a distance function D, as follows:

$$D(x, \mu_i, \omega_i) = \frac{|\omega_i \cdot x - \mu_i|^2}{\sum_{j=1}^{|\omega_i|} \omega_{ij}}$$

where $\omega_i \cdot x$=($\omega_{i1}x_1$, $\omega_{i2}x_2, \ldots$). Then, the index of the label of sample *x* is *j*:

$$j = argmin_i D(x, \mu_i, \omega_i)$$

$Label_j$ then is the label of the samples in the j^{th} leaf node of the decision tree. The distance will be calculated when distributing the samples to the corresponding leaf nodes as described in the next section.

3.3 The One-Step K-means Algorithm Layer

On one hand, the DT model can be seen as a weak classifier which can classify a new user's samples with the accuracy higher than random guess. On the other hand, k-means algorithm can cluster samples into several classes, but its clustering quality and iteration number are related to the initial *K* cluster centers and the algorithm can be very slow to converge with a bad initialization. In our solution, we combine the merits of DT model and the k-means algorithm to realize cross-people activity recognition.

Firstly, the target user's samples can be classified by the DT: $D_{tar} = \{(x_{tar}^{(i)})\}_{i=1}^{N_2}$, into classes, $V_{tar} = \{(Label_j, \omega_j, \mu_j, V_j)\}_{j=1}^{m}$, where *m* is the number of leaf nodes, $Label_j$ is the label of the samples inside the j^{th} leaf node, ω_j is calculated by following Formula (1), μ_j is the center of the samples in the j^{th} leaf node and V_j is used to store the samples in the j^{th} leaf node.

Secondly, we prepare the initial conditions of K-means clustering algorithm. We wish to run this algorithm once in each run of the outer-loop, so that all sample will look for their closest leaf node centers just as in a K-means algorithm.

Let V_{tar} be the set of target data. We first compute the initial class centers for each leaf node j: $\mu_j = \frac{\omega_j \cdot \sum_{i=1}^{|V_j|} x_i^j}{|V_j|}$, where x_i^j is the i^{th} sample of class *j*. Then, we carry out a re-assignment step based on the distance of each sample to a center. This step allows each sample *x* to be labeled by the leaf node closest to it by the distance metric.

$$j = argmin_i D(x, \mu_i, \omega_i)$$

Then, sample *x* is re-assigned to the leaf node V_j. This completes the one-step K-means algorithm.

After the above one step K-means method, we can get the labeled samples.

3.4 Update the DT Model

With the target samples of a new user re-assigned to various leaf nodes, we can update the threshold values of all non-terminal nodes in a bottom-up fashion. For a non-terminal node with attribute A_i in DT, we collect the sample set on the left subtree in the set LSamples, and all samples on the right subtree in RSamples. Since θ_i, the value of A_i, separates the two sample sets into two extremities, we can find a value between the maximum of the LSamples set and the minimum of the RSamples set. This new threshold value can be calculated as follows:

$$\theta_i = \frac{argmax_{x \in LSamples} x_i + argmin_{x \in RSamples} x_i}{2}$$

where x is a sample and x_i is the value of the j^{th} attribute of it.

After all the non-terminal nodes are updated, the updated DT model, DT', can be generated. The process repeats until convergence.

3.5 Formal Description of the TransEMDT Algorithm

To summarize, our TransEMDT algorithm is listed in Algorithm 1. At step 1, the DT model is learnt from the labeled samples. From step 2 to step 4, the DT model is transferred to a new user and the initial conditions for the k-means algorithm are prepared. At step 5, the k-means algorithm repeated only once. At step 6, the non-terminal nodes of DT are adapted. At step 7, The personalized model is returned.

Algorithm 1: TransEMDT Algorithm

Input: Source domain $D_{src} = \{(x_{src}^{(i)}, y_{src}^{(i)})\}_{i=1}^{N_1}$, where $y_{src}^{(i)}$ is the label of $x_{src}^{(i)}$, has N_1 labeled samples and all the samples are from one person. Target domain $D_{tar} = \{(x_{tar}^{(i)})\}_{i=1}^{N_2}$ does not have any labeled samples and they are all from another person. *Thd* is a threshold that determine the number of the iterations to run the integrated one-step K-means algorithm and updating the DT. K is the number of high confidence samples selected for each activity class in each iteration. *Times* is an upperbound on the number of iterations.

Output: Personalized Model

Begin:

1. Learn a DT model from the labeled samples D_{src}.
2. For every leaf node of DT, find the related attributes and form the vector ω_i which indicates the importance of every attribute. This vector is used for calculating the centers of the leaf nodes.

   t=0;

   While $\sum_{j=1}^{m} \sum_{i=1}^{|V_j|} D(x_i^j, \mu_j, \omega_j) > Thd$ or t<*Times*
3. Classify each sample in D_{tar} with DT, with the result as $V_{tar} = \{(Label_j, \omega_j, \mu_j, V_j)\}_{j=1}^{m}$.
4. Set initial seeds for the one-step K-means algorithm. $\mu_j = \frac{\omega_j \cdot \sum_{i=1}^{|V_j|} x_i^j}{|V_j|}$
5. For each sample x, find the leaf node closest to it by the distance $D(x, \mu_i, \omega_i)$, and change its membership to that leaf node.
6. Update all non-terminal node of DT in a bottom-up process: for each leaf node, select K high confidence samples nearest to their leaf-node center, in order to participate in the adjustment of the thresholds.

   For each non-terminal node A_i in DT, let LSamples be the samples on the left subtree, the RSamples for the right subtree. We update the threshold as $\theta_i = \frac{argmax_{x \in LSamples} x_i + argmin_{x \in RSamples} x_i}{2}$
   End For

   t=t+1;
   End While
7. Output the personalized model DT

End Begin

---

4 Experiment

4.1 Data Collection

In our experiments, we used Nokia N95 8GB mobile phones to collect the accelerometer data. An activity database is constructed from the data collected from these devices. In this database, there are 10 participants and five activities. The sliding window method is used to extract the features. The sampling frequency of N95 accelerometer sensor has been reduced to approximately 32 Hz by calling the Nokia Accelerometers plug-in API. Our chosen window size is two seconds and the overlap time is one second. Thus a complete action can be included in the window. Feature extraction on windows with 50% overlap has demonstrated successful in previous work [Bao and Intille, 2004]. In each window, features are extracted from each of the three axes of the accelerometer. The features are: Mean, Standard Variance, the-Fast Fourier Transform energy and the correlations between every pair of axes. So there are a total of 12 features extracted from one window and these features are processed into a single sample. The number of samples of every activity are listed in Table 1.

Activity Name	Label	*Number of sample*
stationary	1	4520
downstairs	2	4293
walking	3	4327
Running	4	4245
upstairs	5	4369

Table 1: Activity Sample Information

4.2 Algorithm Performance

To illustrate the performance of our algorithm in detail, we have verified the TransEMDT algorithm with a full data set, as listed in Table 2. A decision tree named Model A was learnt from 100 samples of Person A, 20 samples per class. Then, we classified samples of Person B, clustered them with k-means algorithm and selected 20 samples per class to update Model A. This step was repeated 5 times. Lastly, the remaining samples of Person B were classified with the updated model. The rationale for choosing these parameters (100, 20 and 5) will be discussed in the following experiments.

Pairs	Before TransEMDT	After TransEMDT
45	61.25%±5.35%	82.16%±6.24%

Table 2: Algorithm Performance

As can be seen from table 2, there are 45 permutations of selecting 2 persons from 10 persons. When applying the TransEMDT algorithm, the average performance of the model can be improved by 20%.

4.3 Comparison of Different Classifier

The Decision Tree (DT) method has been compared with existing methods including Naïve Bayes Network (NBN) and Support Vector Machine (SVM). For the DT, NBN implementation, we use the Weka Machine Learning Algorithms Toolkit and for the SVM implementation, we use the LIBSVM package. Before learning and testing the classifiers, we have scaled the data to [0, 1]. When SVM is used, we choose the optimal parameter c and g by traversing all the values. And for each classifier, 10-fold cross-validation is used. Each participant's sample set is equally divided into two parts (this is done not only for the entire dataset, but also for every class). The classifiers are trained on one half of the dataset and tested on the other half and 10-fold validation is employed. This process repeats for all ten subjects and the mean accuracy is reported. Detail results are shown in Table 3 below.

	DT	SVM	NBN
Mean Accuracy	85.31%	83.99%	80.67%

Table 3: Comparison of Classifiers

As we can learn from Table 3, for our database, Decision Tree indeed outperforms other classifiers.

4.4 DT Model vs Training Samples

In this experiment, firstly, we have learnt Model A from K samples of Person A. As described in our algorithm, it is called DT model. Secondly, we test the DT model with the remaining of samples of Person A. Thirdly, we test the DT model with all the samples of Person B. Person A and Person B are randomly selected from the ten participants. The process of selecting the participants, learning the model and testing repeate ten times and the average accuracy is shown in Figure 5. The K value has been varied from 15 to 330 to study the relationship between the accuracy of the model and the number of the training samples.

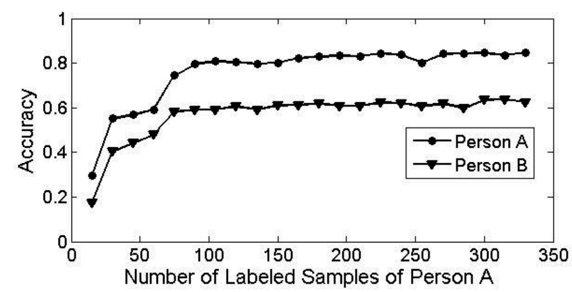

Figure 5: DT Model vs Training Samples

As shown in Figure 5, the accuracy of the DT model increases significantly when the number of training samples varies from 15 to 100 and remains almost the same when the number is larger than 100. In other words, with 100 training samples, we can learn a well built model. The DT model has a moderate accuracy to classify the new user's data.

4.5 Personalized Model vs New User Samples

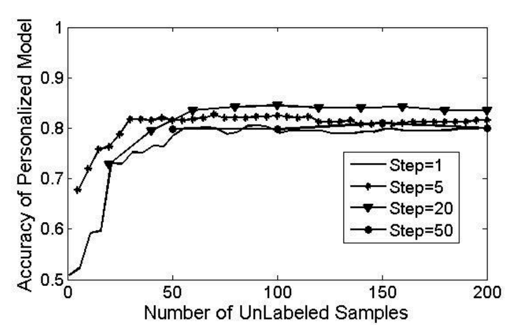

Figure 6: Personalized Model vs New User Samples

In this section, we study how well the DT model transfers to a new user and test the TransEMDT algorithm. To address the power efficiency problem, our experiment attempts to find a trade-off between the accuracy and the amount of computation involved in the algorithm. Our goal is to find the number of unlabeled samples and the number of iterations that minimizes energy consumption while still produces reasonably accurate classification results. A variable named "Step" is used to represent the number of unlabeled samples that will be used to update the DT model in each iteration. The accuracy data produced by using different "Step" values are shown in Figure 6. It can be observed that:

1) The TransEMDT algorithm can converge regardless of whether the "step" is too small or too large in our tests.

2) A value of about 20 for the "Step" variable would yield the optimal results. This can be explained by the fact that, when the value of"Step" is too small, the useful information of the new user is comparatively little in each iteration . When "Step" is too large, it may import noise into the personalized model.

5 Conclusion and Future Work

In this paper, we have proposed the TransEMDT algorithm to solve the cross-people mobile-based activity recognition problem. Empirical evaluations have shown that:

1) The TransEMDT model can be used to transfer the knowledge on user activities.

2) The number of training samples can affect the performance of the classifier. When the number is larger than 100 (i.e., 20 samples per class), the classifier is close to the optimal result. It is very important to reduce the training samples to construct a power efficient application.

3) When transferring the DT model to a new user, if we add 20 unlabeled samples each time, after five iterations, the best personalized model can be obtained. This finding tells us that there is no need to iterate too many times to update the DT model for a new person, which is important for reducing the power consumption significantly, an issue which is especially important and useful for resource constrained devices.

Although we have investigated several important aspects of cross-people mobile-phone based activity recognition, there are still lots of open research problems worth further consideration.

1) Applying this solution to non-binary decision trees: This problem is more challenging than the binary decision tree as the average value between the maximum and minimum may not be correct for computing the new threshold any longer.

2) Considering larger data set and complicated scenarios: In the future, we will employ 50 persons to collect data. They should have different physical conditions such as gender, age, height, et al. We will collect the realistic data as the participants go about their normal activities.

6 Acknowledgments

This work was supported in part by the Natural Science Foundation of China (61070110, 90820303) , and Beijing Municipal Natural Science Foundation (4112056).

References

[Albinali *et al.*, 2010] F. Albinali, S. S. Intille, W. Haskell, and M. Rosenberger. Using wearable activity type detection to improve physical activity energy expenditure estimation. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, 2010.

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

[Chen *et al.*, 2010] Yiqiang Chen, Juan Qi, Zhuo Sun, and Qiong Ning. Mining user goals for indoor location based services with low energy and high qos. *Computational Intelligence*, 26(3):318–336, 2010.

[Consolvo *et al.*, 2006] S. Consolvo, K. Everitt, I. Smith, and J.A. Landay. Design requirements for technologies that encourage physical activity. In *Proceedings of the CHI*, pages 457–466, Montreal, Canada, April 2006.

[Hamerly and Elkan, 2002] Greg Hamerly and Charles Elkan. Alternatives to the k-means algorithm that find better clusterings. In *Proceedings of the eleventh international conference on Information and knowledge management*, CIKM '02, pages 600–607, New York, NY, USA, 2002. ACM.

[Lee and Giraud-Carrier, 2007] J. W. Lee and C. Giraud-Carrier. Transfer learning in decision trees. In *In Proceedings of the International Joint Conference on Neural Networks*, 2007.

[Nachman *et al.*, 2010] L. Nachman, A. Baxi, S. Bhatta-charya, V. Darera, P. Deshpande, N. Kodalapura, V. Mageshkumar, S. Rath, J. Shahabdeen, and R. Acharya. Jog falls: A pervasive healthcare platform for diabetes management. In *Proceedings of the Pervasive*, pages 94–111, 2010.

[Pan and Yang, 2010] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[Ravi *et al.*, 2005] N. Ravi, N. Dandekar, P. Mysore, and M.L. Littman. Activity recognition from accelerometer data. In *Proceedings of AAAI*, pages 1541–1546, 2005.

[Stikic *et al.*, 2008] M. Stikic, K.V. Laerhoven, and B. Schiele. Exploring semi-supervised and active learning for activity recognition. In *Proceedings of the IEEE Int. Symp. on Wearable Computers*, 2008.

[Torrey *et al.*, 2005] L. Torrey, T. Walker, J. Shavlik, and R. Maclin. Using advice to transfer knowledge acquired in one reinforcement learning task to another. In *Proceedings of the Sixteenth European Conference on Machine Learning*, pages 412–424, 2005.

[Ward *et al.*, 2011] Jamie A. Ward, Paul Lukowicz, and Hans W. Gellersen. Performance metrics for activity recognition. *ACM Trans. Intell. Syst. Technol.*, 2, January 2011.

[Zheng *et al.*, 2009] V. W. Zheng, D. H. Hu, and Q. Yang. Cross-domain activity recognition. In *Ubiquitous computing*, pages 61–70. ACM, 2009.