Active Learning Enabled Activity Recognition

H M Sajjad Hossain, Nirmalya Roy, Md Abdullah Al Hafiz Khan
Department of Information Systems, University of Maryland Baltimore County
riaj.sajjad@umbc.edu, nroy@umbc.edu, mdkhan1@umbc.edu

Abstract—**Activity recognition in smart environment has been investigated rigorously in recent years. Researchers are enhancing the underlying activity discovery and recognition process by adding various dimensions and functionalities. But one significant barrier still persists which is collecting the ground truth information. Ground truth is very important to initialize a supervised learning of activities. Due to a large variety in number of Activities of Daily Living (ADLs), acknowledging them in a supervised way is a non-trivial research problem. Most of the previous researches have referenced a subset of ADLs and to initialize their model, they acquire a vast amount of informative labeled training data. On the other hand to collect ground truth and differentiate ADLs, human intervention is indispensable. As a result it takes an immense effort and raises privacy concerns to collect a reasonable amount of labeled data. In this paper, we propose to use active learning to alleviate the labeling effort and ground truth data collection in activity recognition pipeline. We investigate and analyze different active learning strategies to scale activity recognition and propose a dynamic k-means clustering based active learning approach. Experimental results on real data traces from a retirement community-(IRB #HP-00064387) help validate the early promise of our approach.**

I. INTRODUCTION

Recently we have been experiencing the rapid development in using ambient technologies and smart devices which has been triggered by the adaptable sensor technologies. These advancements in technologies have broadened the functional domain of mobile and sensor computing. As a result Internet of Things (IoTs) where multimodal sensor setup is combined to perform context aware actions, is becoming an integral component. Wide variety of context aware applications like occupancy detection [1], HVAC control [2], localization, health monitoring [3] etc. , have been introduced. Combining with human activity recognition model, an IoT inspired domain "*Smart Home Technologies*" have been gaining more attention for health monitoring, independent living for senior citizens and making our life comfortable. Smartthings, Vera, Microsoft Lab of Things, openHAB, Ninjablocks, Twine, CASAS:Smart Home in a Box [4] and other automation systems are already in the market for the users. Researchers are extending the capabilities of these automation systems by augmenting different functionalities. One major driving component for these context-aware smart home technologies is to reliably *learn* the human activities using the observable states of ambient and wearable sensor data. For *learning* and *recognizing* activities of daily living (ADLs) machine learning or rule based algorithms are being exploited extensively. These trained models are then employed to make decisions according to the domain of the application even in unknown situations. Traditional passive learning approaches are only consistent with the existing class labels. The training for passive learning is carried out in specific experimental setup which is not ideal for real life

applications due to a large variety of human activities and underpinning uncertainty with the data capturing. In order to build a robust and stable model, we have to provide vast amount of labeled data- *ground truth information* which is cumbersome and not always feasible. Building adaptive and personalized model for individual users has become a crucial obligation because of the diversity of a same activity across different individuals. For example, the differences in speed of walking, gestures, sleep habits etc., are ambiguous to a general passive learning model. *Active learning* [5] helps us to determine the most informative data points which is a semi-supervised approach. We can actively query the users for labeling informative data points and mitigate the necessity for acquiring a large amount of labeled data points by applying active learning. In activity recognition tasks using ambient or wearable sensors, we have abundant unlabeled data instances which makes active learning as an ideal solution for building an efficient classification model. Although crowdsourcing provides the platform for labeling large amount of data, the fundamental question is how much of these labeled data will be statistically significant, reliable, noise free and just-in-time. Also from a requester perspective increase demand in data labeling in turn increases the cost of crowdsourcing. Therefore, a cost-effective crowdsourcing model needs to be designed. In this case, active learning can potentially help us to identify potential important data instances that can balance out the trade-off between cost and model performance.

In general, a subset of ADLs are dealt with and most of the proposed activity recognition models are trained with a handful number of activities. On the other hand, it is quite impossible to train the classifier with as many activities as possible. Thus we need an activity recognition system which can dynamically discover unseen activities. Recognition model which can discover unseen activities has been investigated in some of the previous works. [6] proposed a semantic attribute based activity recognition model by creating an activity-attribute matrix for learning unseen activities while [7] proposed a metric learning based approach to reject unseen activities. An activity recognition model using finite state machines which can differentiate unimportant instances of activities for learning a generalized model has been presented in [8]. The use of active learning for activity recognition systems has been investigated by very few researchers. The authors of [9] have proposed a bayesian active learning approach for labeling data in smart homes. Uncertainty based active learning for activity recognition has been employed in [10] [11]. The authors of [12] used entropy based measure to calculate the informativeness of activity data instances. An activity recognition framework called Legion:AR [13] uses active learning and seeks for labels from crowd on demand. AALO [14], a single habitant activity recognition model for smart homes uses active

learning for labeling overlapped activities. In this paper, we propose a model for active learning which can help identify latent informative and as well as unseen data instances and validate the annotators in the labeling process.

One of the other major challenges in crowdsourcing or collaborative systems is managing the expertise of the crowd. The crowd have different backgrounds and intellectual ability which make the feedbacks noise prone. One way to tackle this problem is to rank the annotators based on their reliability and distribute the labeling process appropriately. One other way is to train the crowd with appropriate domain specific examples and increase their competence. This requires a proper representation of the crowdsourcing scenario. Crowdsourcing raw sensor data is challenging as it is very difficult to analyze and find meaningful patterns in the data to associate with a class for a general worker. We have to provide some sort of semantic information to the worker to help them understand the scope of the problem domain. In this work, we discuss about data representation ways for active learning and crowdsourcing sensor data to scale and deploy practical activity recognition applications among a community of individuals.

A. Our Contributions

Most of the proposed active learning model focused on finding the most informative point using uncertainty measurement or maximizing error reduction. For some classifiers a basic intuition is followed - *any data instance closest to the decision boundary is considered important and thus queried*. Some approaches are strictly for SVM classifiers where the intention is to identify the instances from feature space which will maximize the hyperplane margin. These approaches can yield good results and improve the overall performance, however they tend to ignore the prior distribution of the feature space. Prior distribution can be useful for active learning and ignoring them can create sampling bias in the overall system. In this paper we propose a cluster based active learning model for activity recognition. Combining clustering with active learning has been proposed in [15] [16] [17]. But postulating them for practical human-in-the-loop applications are still in its infancy.

In our proposed model, we first create hard clusters without explicitly differentiating the number of clusters, rather focusing the minimum number of clusters in association with the existing number of labels in the label space. We posit the most informative instances in a cluster and, subsequently acquire label for them using our novel objective function and finally reinstate the cluster label through empirical validation. In summary our proposed active learning enabled activity recognition model contains following salient contributions:

- We propose a dynamic k-means clustering algorithm for creating clusters of unlabeled data.

- We propose an objective function to find the most informative data instance in the cluster.

- We validate our model using real life data traces and compare our model with other viable active learning approaches such as disagreement based approach.

- We present a data representation technique for crowdsourcing smart home data and discuss its implication on active learning assisted activity recognition.

The paper is organized as follows. Section II presents the active learning methodologies. The design architecture of our proposed model is presented in Section III. K-means clustering and optimizing heuristic are illustrated Sections IV and V.The objective function for filtering informative instances is discussed in Section VI. In Section VII, we propose a worker selection model for effective active learning. Experimental results and comparison of our proposed model with existing strategies are presented in Section VIII. In Section IX, we discuss data representation techniques for making active learning and crowdsourcing applicable in large scale activity recognition domain. We articulate the challenges in Section X and conclude with future research directions in Section XI.

II. RELATED WORKS

Active learning algorithms aim to ease the learning complexity and cost by sequentially selecting optimal number of informative unlabeled data instances to query for their label in order to minimize the prediction error of the classification model. Studies have shown that active learning can help reduce the labeling effort in different domains [18] [15]. The most important step for active learning is to define the informativeness measurement for data instances. In a smart home setting a wearable sensor reading may belong to *watching television*, a high level activity class and *sitting* micro activity class. In such cases the learner can consolidate both feature and label space (feature-label pair) instead of relying on the distinct features for evaluating uncertainty. Evaluating instance uncertainty is a common approach for measuring the informativeness of a data instance. The uncertainty is measured with respect to the feature space. In this approach the learner can focus more on the data instances which are confusing based on the uncertainty score. Least Confident, Margin Sampling and Entropy learning are the most popular informativeness measures [19]. Another uncertainty resolving approach is searching through hypothesis space where the associated classifier maintains a set of candidate hypothesis space known as version space. The goal of the active learning algorithm is to minimize the cardinality of the version space which depicts maximal change to the current classification model. One popular and significant contribution in this approach is disagreement based active learning [20]. Another approach *Variance Reduction* chooses the instances that minimize the square loss of a learner [21].

Some active learning works have proposed to augment instance correlations where an utility metric from a sample or sample-label space has been defined as a combination of an uncertainty function and a correlation function. In feature based correlation, a similarity measurement [22] or a correlation matrix [23] on features has been utilized to compare pairwise similarities of instances, so the informativeness of an instance is weighted by average similarity on its neighbors. *Label correlation* is widely used for multi label learning. Conventional active learning algorithms considers an oracle to provide the correct label for each query always which is not ideal. Many of the existing active learning strategies are prone to introduce noise on learning models, as the process of finding an optimal boundary between two classes involves label queries that have lower proximity to the decision boundary and usually these labels induce large noise. The authors of [24] have proposed

an algorithm A^2 Learning which works in presence of arbitrary forms of noise. [25] uses a randomized query mechanism and includes importance weights in the calculation of empirical error rates, to compensate for the bias in the sample so that it is possible to obtain rough estimates of the excess empirical error rates. we define the label complexity of A^2 learning [24] by using the disagreement coefficient has been described in [20]. The authors of [26] proposed a bayesian based active learning algorithm where queries are selected sequentially to reduce uncertainty. The authors demonstrate that instead of focusing on minimizing uncertainty, the aim is to drive uncertainty into a single decision region as quickly as possible. Next we discuss the usage of active learning for practical applications such as smart home activity recognition. In [16] the author used a hierarchical clustering for unlabeled instances and then a data instance is picked randomly from the cluster. [15] tries to reduce the classification variance by employing Query By Committee (QBC) method.

III. Overall Design & Challenges

Our proposed model is comprised of two significant steps. First formulate clusters for the unlabeled data instance pool (U) using K-Means clustering. The fundamental motivation behind using clustering over other active learning strategies is *computational complexity*. Though uncertainty sampling based strategies are computationally inexpensive, they tend to be biased and become over confident. Other popular disagreement based strategies like Query By Committee (QBC) uses the hypothesis space to form the committee, and maintaining a hypothesis space demands a lot of computation. For a practical and real life system, we need an efficient strategy with low computational complexity. Clustering based strategies reduces the overall complexity of the active learning algorithm. Dasgupta et al. [16] used hierarchical clustering which becomes computationally expensive for large data sets. K-Means clustering is a widely used partition clustering method and if employed with proper heuristic, K-means can achieve linear time complexity. The clustering process follows a simple and easy way to classify a given data set through a certain number of clusters (let us assume k clusters) fixed beforehand. The idea is to identify the center of the clusters - centroids, and then associate the data instances to the closest centroid and formulate clusters. However the computational complexity becomes NP Hard. One other challenge for applying K-Means clustering is that we have to postulate the number of clusters explicitly. For building an adaptive and spontaneous activity recognition model to discover unseen activities, it is not practical to posit the number of clusters beforehand. We iteratively run the clustering until we find a stable set of clusters with minimum clustering error. We decrease and control the computational complexity of K-Means clustering by using Elkan's heuristic [27]. We apply a pool based sampling strategy for constructing instance pool (U) as it has been used in practice [28] [29] [30]. We delve into the practical application of active learning and particularity for smart home activity recognition where a multitude of ambient and wearable sensor data streams are abundant. Thus we design a sampling approach based on a pool of unlabeled data instances and then pass it to our active learner.

After compiling the clusters out of unlabeled data instances pool, we find out the most informative data instances and
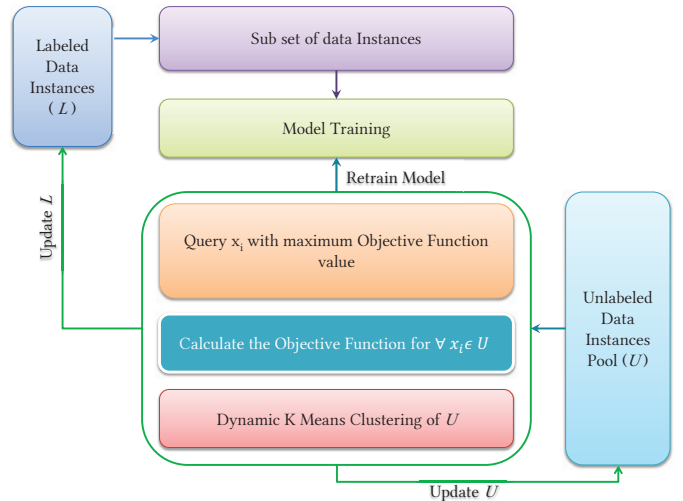

Fig. 1. Overall framework for active learning inspired activity recognition.

query them accordingly. We also incorporate the clusters with unseen activities in this underlying process. In order to find the most informative data instances, we formulate an objective function which is constructed using entropy measurement and a similarity coefficient. The similarity coefficient for any data instance is calculated by the distance measurement between the points in its cluster and other surrounding cluster centers. Fig. 1 depicts this entire active learning enabled activity recognition model.

IV. Background

Given n points $\{x_1, x_2....., x_n\} \in \mathbb{R}_d$ the goal of K-means is to find K cluster centers $\{c_1, c_2...., c_m\} \in \mathbb{R}_d$ and assignment $\{q_1, ..., q_n\}$ of the points to the centers. K-Means clustering tries to find the position of the cluster centers and minimize the distance of the data instances in Eqn 1. K-means is obtained for the case p=2 (l^2 norm), because in this case the optimal centers are the means of the input vectors assigned to them.

$$E(c_1, ..., c_k, q_1, ..., q_n) = \sum_{i=1}^{n} \|x_i - c_{q_i}\| \quad (1)$$

Minimizing the objective E is in general a difficult combinatorial problem, so locally optimal or approximated solutions are sought instead. E is also the average reconstruction error, if the original points are approximated with the cluster centers. Thus K-means is used not only to group the input points into cluster, but also to quantize their values. The basic K-means algorithm alternate between re-estimating the centers and the assignments. Combined with a good initialization strategy and potentially, by re-running the optimization from a number of randomized starting states, this algorithm helps reduce the complexity of handling exponential state-space of active learning and increase efficiency in practice. However, despite its simplicity, simple K-means is often too slow. Thus we consider Elkan's algorithm [27], which uses the triangular inequality to cut down significantly the cost of basic K-means.

V. Cluster Heuristics

Elkan's algorithm [27] is different than Lloyd alternate optimization algorithm (Lloyd's algorithm) which calculates

the triangular inequality to mitigate many distance calculations when assigning data instances to clusters. Although this heuristic is much faster than Lloyd, but it needs storage proportional to the number of clusters. which makes it difficult to operate in case of large number of clusters. The base of this algorithm is that, if a centroid update does not move data instances much, then most of the instance to center distance computations can be avoided when the assignments are recomputed. To distinguish which distances need evaluation, the algorithm bounds the distances by lower and upper bound using triangular inequality after a center update. Elkan algorithms uses two key observations. First, one has

$$\|x_i - c_{q_i}\| \le \|c - c_{q_i}\|/2 \quad \Rightarrow \quad \|x_i - c_{q_i}\| \le \|x_i - c\| \quad (2)$$

Thus if the distance between x_i and its current center c_{q_i} is less than half the distance of the center c_{q_i} to another center c, then c can be skipped when the new assignment for x_i is searched. Checking this requires keeping track of all the inter-center distances, but centers are typically a small fraction of the training data, so overall this can be a significant saving. In particular, if this condition is satisfied for all the centers $c \neq c_{q_i}$, the point x_i can be skipped completely. Furthermore, the condition can be tested also based on an upper bound $U(x)$ of $\|x_i - c_{q_i}\|$. Second, if a center c is updated to $\hat{c}$, then the new distance from x to $\hat{c}$ is bounded from below and above by

$$\|x - c\| - \|c - \hat{c}\| \le \|x - \hat{c}\| \le \|x - \hat{c}\| + \|c + \hat{c}\|. \quad (3)$$

This allows to maintain an upper bound on the distance of x_i to its current center c_{q_i} and a lower bound to any other center c.

$$U(x) \leftarrow U(x) + \|c_{q_i} - \hat{c}_{q_i}\| \quad (4)$$
$$L(x, c) \leftarrow L(x, c) - \|c - \hat{c}\|. \quad (5)$$

Traditional K-means clustering needs prior definition of number of clusters. But if data instances of unseen classes are present in the data pool, they will be considered as outliers in the clustering process. Having outliers in the feature space can significantly decrease the performance of the clustering algorithm if the number of clusters are not properly defined. From these motivations we choose to apply an incremental K-means clustering where at each iteration we increase K and record the overall error for our clustering using the error function in Eqn. 6. The minimum error depicts the best clustering for the data set and the corresponding number of k cluster.

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \|x^{(j)} - c_j\|^2 \quad (6)$$

VI. Measuring Informativeness

After clustering the unlabeled data instances, our task is to figure out the most informative data instance in different clusters. [16] proposed random sampling for labeling the data. Instead of using random sampling we propose to use uncertainty sampling to choose the most informative instance. One challenge is to differentiate between outliers and most informative data instances as outliers will have much higher uncertainty like the most informative instances. To represent whether a data instance with high uncertainty is an outlier

or not, we calculate the silhouette coefficient [31], $S_c^{(x_i)}$ for each data instances. For each data instance x_i, let $a(i)$ be the average dissimilarity of x_i with all other data within the same cluster. $a(i)$ interprets how well a data instance x_i is assigned to its own cluster. Where smaller value of $a(i)$ indicates better assignment of x_i. The average dissimilarity of point x_i to a cluster c is defined as the average of the distance from x_i to points in c. Let $b(i)$ be the lowest average dissimilarity of x_i to any other cluster, of which x_i is not a member. The cluster with this lowest average dissimilarity is said to be the "neighbouring cluster" of x_i because it is the next best fit cluster for point x_i. We now define a silhouette:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (7)$$

$$s(i) = \begin{cases} 1 - \frac{a(i)}{b(i)}, & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ \frac{b(i)}{a(i)} - 1, & \text{if } a(i) > b(i) \end{cases} \quad (8)$$

From Eqn. 8 it is evident that $-1 < s(i) < 1$. For $s(i)$ to be close to 1, we require $a(i) \ll b(i)$. As $a(i)$ is a measure of how dissimilar x_i is to its own cluster, a small value means it is well matched. Furthermore, a large $b(i)$ implies that x_i is poorly matched to its neighbouring cluster. Thus an $s(i)$ close to 1 means that the instance is appropriately clustered. If $s(i)$ is close to negative one, then by the same logic we note that x_i would be more appropriate if it was clustered in its neighbouring cluster. An $s(i)$ near zero means that the instance is on the border of two natural clusters. The informativeness is measured by the entropy of the instances. The entropy of an instance is defined by the following equation:

$$e_\theta(x) = \operatorname*{argmax}_x H_\theta(y|x)$$
$$= \operatorname*{argmax}_x (-\sum_y P_\theta(y|x) \log P_\theta(y|x)) \quad (9)$$

We combine this entropy measurement with the $S_c^{(x_i)}$ to filter out the most informative points. So the final objective function f_c for finding the most informative point becomes

$$f_c(x) = \operatorname*{argmax}_x \{e_\theta(x) . S_c^{(x_i)}\} \quad (10)$$

By using Eqn. 10, the points which are properly clustered and has higher entropy values are chosen. The points which has $S_c^{(x_i)} = 0$ or negative values might also be important. So instead of discarding those data instances, we randomly pick instances and query them. An instance with $S_c^{(x_i)}$ closer to zero is close to the boundary of two different clusters. If we have the same label as other data instances in that cluster, we do not have to change the cluster, but if different label is received we have to rearrange our cluster with respect to the received label. In Algorithm 1 we demonstrate the full active learner with K-means clustering.

VII. Selecting Annotator

We filtered out the most informative data instances in the previous step, but it is quite impractical to presume in a real life environment that the annotators are going to provide true and correct labels all the time. As the search for most informative data instances is important so is getting the true

Algorithm 1 Active Learner with K-Means Clustering

1: **Input:** $U = $ A pool of unlabeled instances $\{(x)^u\}_{u=1}^{U}$,
   $min_k = $ Minimum number of clusters,
   $\theta = $ An error threshold.
2: **Output:** Clustered classification of U and most informative data instances in each cluster.
3: **for** $K = min_k$ **do**
4:   **Initialization:** Compute $L(x^i, c) - \|c - \hat{c}\|$
5:   Find the current assignment q_i and bounds $U(x^i)$ by finding the closest centers to each point
6:   Estimate Center and Quantize the data instances in U to associate them with a cluster c.
7:   Compute the Error $J_{current}$ using Eqn. 6.
8:   **if** $K = min_k$ **then**
9:     Continue
10:   **else**
11:     **if** $J_{current} > \theta$ **then**
12:       $K = K + 1$
13:     **else**
14:       break
15:     **end if**
16:   **end if**
17: **end for**
18: **for** every $x_i \in U$ **do**
19:   Calculate $S_c^{(x_i)}$ using eqn 8.
20:   Calculate the objective function $f_c(x)$ using Eqn. 10.
21:   Choose the instance with highest $f_c(x)$ and query for label.
22: **end for**

---

label for them, otherwise unnecessary noise draws on our model. In order to build practical active learning enabled activity recognition model, choosing the right annotator is crucial and it becomes more challenging when the environment is cohabited by multiple inhabitants. As all annotators are not expert in annotating a specific activity class, we assign a sensitivity score to each annotator for each activity class. The sensitivity is defined by the number of labels correctly labeled by an individual annotator.

$$\mu_{sen}^c = \frac{\text{correctly labeled instances of class c}}{\text{number of points queried of class c}} \quad (11)$$

Initially when an annotator has not labeled any examples from class c_i, we assign a sensitivity score of $\mu_{sen}^{c_i} = 0.5$. Using the sensitivity score, we pinpoint the most efficient annotator. To further strengthen the querying process, we choose the annotator who has annotated a similar data instance with high sensitivity which is annotator's specificity δ_{spec}. For this we first monitor j neighboring data instances who are closer to x. For our experimental purpose we chose $j = 4$.

$$\delta_{spec_i} = \frac{\frac{1}{k} \times \sum_{i=1}^{j} \mu_{x_i}^{a_k}}{1 + \frac{1}{k} \times \sum_{i=1}^{j} |x - x_i|} \quad (12)$$

In Eqn. 12, $\mu_{x_i}^{a_k}$ is our sensitivity of annotator, a_k. $|x - x_i|$ is the euclidean distance between the neighboring data instances. Our active learner is set to select the annotator by the following Equation 13.

$$i = \text{argmax}(\delta_{spec_1}, \delta_{spec_2} \ldots, \delta_{spec_k}) \quad (13)$$

After selecting a subset of annotators we ranked accordingly based on their specificity score and query the data instance to top 4 annotators. If the confidence of the data instance is less than 80% then we keep on querying until it reaches confidence level of 80%. Here the confidence level for a data instance means the ratio of number of labels and total number of query. If x_i receives 6 labels of class c_j out of total 10 queries, then the confidence score is $100 \times \frac{6}{10} = 60\%$.

VIII. EXPERIMENTAL RESULTS

In this section we validate our active learning algorithm for activity recognition and compare the outcome with other popular strategies. We set up a smart home environment with PIR motion sensors and object sensors on different household appliances. The PIR motion sensors were mounted on three different locations of a single bedroom apartment (bedroom, living room and kitchen). The object sensors were mounted on the broom, trashcan, laundry basket, dustpan and phone. The object sensors have built in compass and accelerometer which provided the usage and orientation of the objects. Two door sensors were placed on the apartment door and on the closet door. In our experiment we considered only single inhabitant environments. We had the following activities in our dataset - 1) brooming, 2) cooking, 3) doing laundry (washing), 4) taking out the trash (cleaning), 5) eating, 6) sleeping and 7) using land line telephone (talking). We trained our passive learner with the first four activities and left the last three activities for our active learner to discover. We have used Decision Tree classifier (J48) as our passive learner. We extracted features from the ambient motion sensors which include the start and end of the sensor events, time span of the event within a k-event window, and count of events in the window. We collected data from 10 participants who reside in a retirement community [32] - IRB (#HP-00064387). Each participant provided around 24 hours of continuous sensor data. Average age of the participants was 85. We evaluated our cluster performance using normalized mutual information (NMI) using ground truth. Both the activity class label and clustering assignment are considered as random variables in NMI. It measures the mutual information between the two data instances, and normalizes it to a zero-to-one range. Let C be the data instance representing the cluster assignments of instances, and K be the random variable representing the class labels of the instances, the NMI is computed by the following equation:

$$NMI = \frac{2I(C; K)}{H(C) + H(K)} \quad (14)$$

Here $I(X; Y) = H(X) - H(X|Y)$ is the mutual information between random variables X and Y. For our experimental validation, we focus on the following specificities for our experiments: 1) Correctly classified instances by our active learner 2) Instance selection time 3) Mean Absolute Error 4) Number of average queries per data instance for gaining confidence level of 80% 5) Annotator selection time 6) Impact for introducing new unseen activities.

A. Smarthome System

We have used CloudEngines PogoPlug [33] as our central component which interfaces with the motion sensors and object sensor tags. We built a custom linux kernel for the PogoPlug
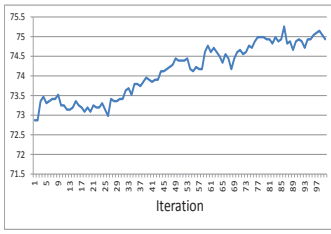
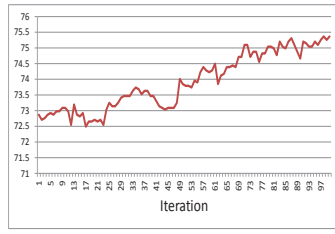Fig. 2. Correctly classified instances for our active learner.

Fig. 3. Correctly classified instances for maximum entropy sampling.
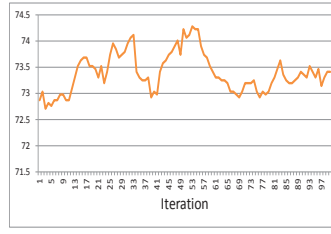

Fig. 4. Correctly classified instances for least confidence sampling.

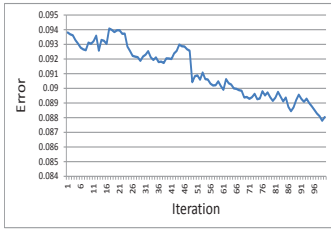Fig. 5. Correctly classified instances for Query By Committee (QBC).

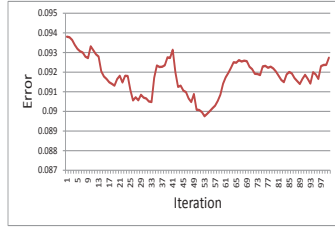Fig. 6. Mean absolute error for our active learner.

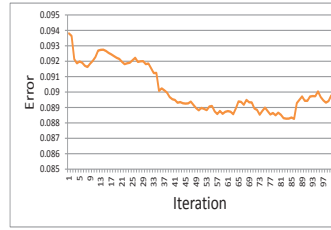Fig. 7. Mean absolute error for maximum entropy sampling.

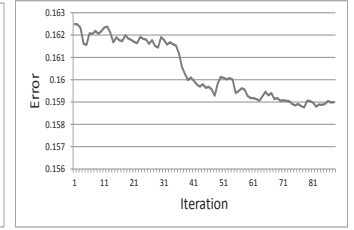Fig. 8. Mean absolute error for least confidence sampling.

Fig. 9. Mean absolute error for Query By Committee (QBC).

Fig. 10. Smarthome System Setup.

Fig. 11. Precision, Recall and F1 Measurement of each activity for Passive Leaner.

Fig. 12. Clustering Performance

and developed tools to communicate with the sensor tags. The PogoPlug works as a bridge between the Sensor Tags and a standard 802.11N network. Tests have indicated the device is stable in this capacity, and recovers from power loss and outages on the external networks without any issue. The PogoPlug is also successfully doing NAT translation in order to bridge additional Ethernet devices on to the network. We are streaming and storing the data in real time in our lab server. We have used Foscam IP camera for collecting ground truth. Due to bandwidth limitation it was difficult to stream the video remotely, so we recorded the video in SD card. Since video violates the privacy issue, each participant provided only two hours of video data. In these two hours the participants were asked to follow a script to perform several tasks related to our activity list. Rest of the 22 hours were not recorded which is our test data. Figure 10 demonstrates the components of our system setup.

B. Supervised Model

In figure 11 the precision, recall and F1 score for each activity is shown for our decision tree classifier. It is evident that the cooking activity has lower accuracy than other activit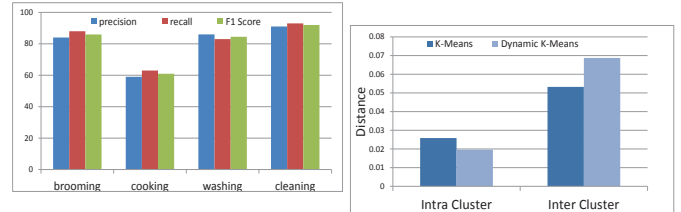ies. As in our experiments, we have placed object sensors on various appliances and equipments, so it was possible to pin point an activity by tracking the usage of the object sensors. For example if we detected any movement for the laundry basket, we labeled the data instance of the motion sensor as cleaning. As for cooking activity there was no object sensor, so it was difficult to track down the activity properly. In some cases the objects were just moved or handled, not used for the corresponding activities which imposed noise in the dataset. For this reason we filtered the dataset by taking an assumption for the duration of each activity. If the duration of the performed activity was less than the threshold then we discard the data instance for training. The thresholds were defined in an empirical manner. Also some of the participants had pets in their apartment which introduced more noises in the dataset as the pets move around the house abruptly.

C. Active Learning Experiments

We have mentioned the criteria for our active learning experiments beforehand. We applied the active learning algorithm in a 10-fold cross validation and pool based sampling manner. We initially started with a small labeled data set (5% of train data), and then made queries by using different active learning strategies. The results are shown for 100 iterations. We compare our algorithm with other popular query strategies - maximum entropy, least confidence and vote entropy or Query by Committee (QBC). QBC is a very effective alternative
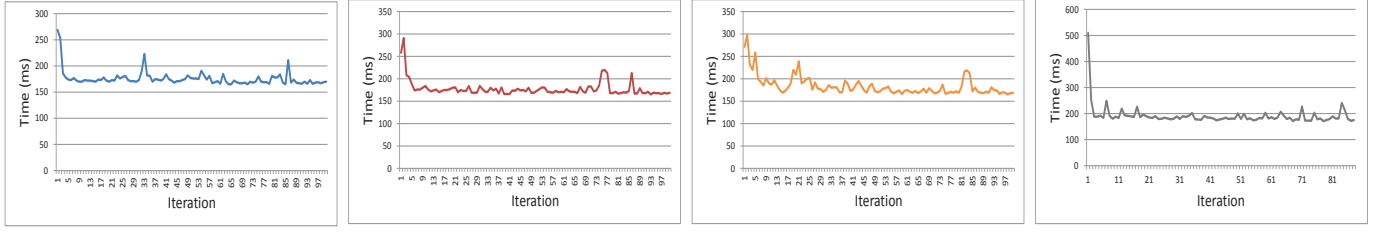
Fig. 13. Instance selection time for our active learner.

Fig. 14. Instance selection time for maximum entropy sampling.

Fig. 15. Instance selection time for least confidence sampling.

Fig. 16. Instance selection time for Query By Committee (QBC).

Algorithm 2 Query By Committee

1: **Input:** U = A pool of unlabeled instances $\{(x)^u\}_{u=1}^U$
   L = A pool of unlabeled instances $\{(x)^l\}_{l=1}^L$
   k = number of iterations
2: Repeat k times
3: Generate a Committee of Classifiers C^*
4: $\forall x^i \in U$, compute disagreement x_{VE}^i using 15 based on the current committee.
5: Select a subset S of instances from U that maximizes utility.
6: Query instances of S
7: Remove S from U
8: Update L by adding S
9: Return

---

approach to uncertainty sampling which has been applied in many classification problems. QBC manipulates the version space and at each iteration it maintains a committee - an effective set of hypotheses based on current training set. The committee evaluates the potential utility of the unlabeled data instance. This utility measure is also called *disagreement measure*. The disagreement measure for QBC was defined by the following equation where $P_c(y|x)$ is the average probability that y is the correct activity label to the committee. The steps of a generalized QBC is show in algorithm 2.

$$x_{VE} = \text{argmax} - \sum_y P_c(y|x) \log P_c(y|x) \qquad (15)$$

First we cluster the unlabeled data set using our dynamic k-means clustering algorithm. In figure 12 we show the intra and inter cluster distance for our dynamic k-means clustering algorithm comparing to simple k-means algorithm. We discuss and compare our active learning algorithm based on each analysis criterion in the following:

1) Correctly Classified Instances: One of the most important performance measurement for an active learning algorithm is how many data instances were correctly classified. In figure 2 correctly classified instance for our algorithm over 100 iterations is plotted. Most of the query strategies show similar results and almost 75% of the instances are correctly classified. However we monitor a lot of changes for least confident sampling as least confident sampling only considers the best prediction and eventually throws away other important information.

2) Instance Selection Time: Instance selection time depicts the speed of the active learning algorithm. Average instance

TABLE I. AVERAGE NUMBER OF QUERIES FOR EACH ACTIVITY FOR GAINING CONFIDENCE LEVEL 80%

Activity	Avg. Number of Queries
Brooming	2
Cooking	9
Washing	3
Cleaning	5
Eating	12
Sleeping	16
Talking	3

selection time for all of the strategies were close to 200 ms (figure 13, 14, 15, 16).

3) Mean Absolute Error: Mean absolute error expresses the difference between the predicted value and the actual value. In figure 6, 7, 8 and 9 the trend of mean absolute error at each iteration for the query strategies are plotted. For our active learner (figure 6), the algorithm converges to lower mean absolute error than other approaches. Mean absolute error for maximum entropy sampling was much higher. After investigating we found that maximum entropy sampling tends to be biased and over generalized. As a result first the mean absolute error started to decrease and in the end it started to increase.
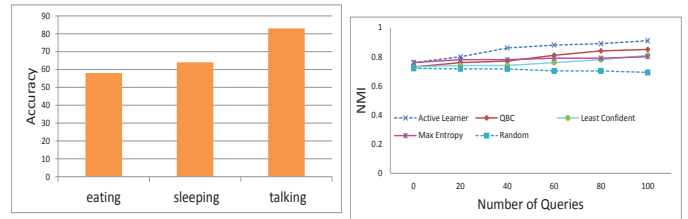

Fig. 17. Accuracy of unseen activities.

Fig. 18. NMI for 100 queries.

4) Average Number of Queries: In our experiments we did not train our supervised model with the acquired label using active learning till the label confidence is 80%. Average number of queries for 100 data instances was ≈ 7. Average number of queries for each activity is shown in table I. From the table it is visible that for Cooking, Eating and Sleeping activity the average number of queries were higher. As for these activities there was no object sensors involved, it was difficult for the annotators to take the decision. For sleeping activity, it was the highest as the movements detected while sleeping are mislabeled frequently by the annotators. Also mostly the eating activity was performed in the living room area, where the participants were frequently sitting and doing their chores. This created confusion among the annotators.

Cooking was another difficult high level activity to properly label as just being in the kitchen does not indicate that the participant was cooking. In figure 18 we show the change in NMI for our clustering algorithm for different active learning strategies. NMI close to 1 means better correlation. Our active learner converges to 1 faster than other query strategies.

5) Introducing Unseen Activities: We left out three activities (eating, sleeping and talking) while training our supervised learning model S. Using active learning, we acquired labels of these unseen activities. After querying them, we trained our supervised model with the received new activity labels. In figure 17 we show the accuracy of S for these new activities. Talking achieved the highest accuracy because of the object sensor attached to the phone. Whenever a movement of the phone is involved, the supervised learner predicts the motion sensor event as talking. For other two activities it was difficult as living room and bedroom involved so many movements. The accuracy of recognizing sleeping is better than eating because the participants performed more variety of activities in the living room than in the bedroom. And also sleeping during night was properly classified by our supervised learner than sleeping at daylight. After including the collected activity labels using our active learner, we retrained our supervised learning model. Initially we labeled our training data using a labeled data set L consisting of $21,014$ instances. On the other hand we had an unlabeled data set U consisting of $126,874$ samples. Initially the average accuracy of S was ≈ 81. If we consider only the four activities, after applying our active learning strategy the 87% accuracy was observed. If new unseen activities are introduced, 77% accuracy is achieved. The actual accuracy decreased, because initially the recognition capability of new unseen activities are low. The mentioned accuracy is reported after 100 iterations, so we increased our iteration to 300 and we achieved 79.5% accuracy. So if we query more data instances using our active learner we can achieve better accuracy.

IX. DATA REPRESENTATION FOR CROWDSOURCING AND ACTIVE LEARNING

Sensor data representation in a user friendly way is a huge challenge in crowdsourcing domain. Even just applying active learning for querying the label from the same user demands an effective and eloquent representation. One straight forward approach for querying the user can be to ask what the user was doing at a certain timestamp. But all human do not have the ability to precisely recall an event at a certain timestamp. Previous works have proposed to annotate image or video recordings by the crowd [34] [35]. Although using image or video based surveillance provides proper idea about the performed activities to the crowd but it violates the privacy of the users.

It is certainly difficult to disjointly represent all the human activities, as there are activities which are similar in functional sense. For example, watching television and studying both involves sitting. It is possible that they are performed at different locations in the apartment, so we can pair activities with different indoor locations. [36] used a visual representation of the motion sensor firing sequence and provided the apartment layout to assist the crowd. But motion sensor activation not always provides a straight forward indication of an activity.

For example, a person might be in the bathroom just to look at the mirror and the motion sensor will still be activated. In a multi-inhabitant environment this becomes more complex. Object sensors can help us greatly in these cases. With proper usage data of an object paired with motion sensor and location information can ideally pinpoint the activity. Human activities are performed in a pattern. While being at home, people are used to eat, cook, vacuum or perform other household activities in a certain pattern. For example, a person may eat his lunch at a certain time in most of the days. Knowing these activity patterns can also help a crowd to understand the functional and behavioral patterns of the user. So while crowdsourcing, it is important to provide a semantic representation of the user activity pattern so that we can collect noise free labels.

X. DISCUSSION AND FUTURE DIRECTIONS

The theoretical foundation of active learning is very rich and resourceful. Researchers have proposed many effective approaches for active learning. But the problem is these solutions are theoretically sound but computationally expensive. Simpler models like entropy based uncertainty calculation are effective but they imposes bias and runs the risk of becoming over confident on incorrect predictions. It is difficult to make computationally expensive approaches applicable. Also not all the approaches perform the same in a specific domain. Choosing the right approach with respect to the scope of the domain is very important for active learning to be effective. In smart home settings, multiple heterogeneous sensors impose various types of uncertainties like biased readings, failed sensors etc. In multi inhabitant environment the situation is much more complex. On the other hand in multi inhabitant settings, we can take advantage of the inhabitants by asking them to label each others data. But still it is extremely difficult to differentiate overlapped sensor reading and individual's activity.

Due to the dynamic nature and a huge variety of human activities it is difficult to collect the ground truth information for an activity learning model. We can record and annotate data sets for training the system from scratch for individual house and individual person but this will be extremely costly. So leveraging active learning can boost the ground truth collection process. As the basic movements and locations of two or more activities can be same, the probability that they will belong to a same cluster is higher. But automatic discovery of similar activities is not handled in our proposed solution. In such cases we have to consider overlapped clusters to properly separate two distinct activity classes. [14] demonstrated an approach for overlapped activities, but the informativeness measurement has not been discussed in their work.

One of the major challenges for making active learning and crowdsourcing practical is the expertise of the crowd. The crowd will not always provide the labels correctly. Also the crowd may not be able to provide labels for some instances at all. In some cases getting label for a data instance from only one annotator will not be enough. Relabeling the data instances by other crowd will further validate the consistency of the acquired label. For handling noisy input from crowd this clear approach may be helpful, but still there is a chance that noise will be introduced. As an example, we have to assume that the crowd is not expert and so it will be challenging for them to differentiate between similar activities. Also if

an unseen activity which is very much alike with an existing one is queried, the crowd may not provide new class label and annotate it with existing label. This impose both bias and noise in the classification model. So the crowdsourcing model should be able persist an agnostic environment. Researchers have proposed to incorporate reliability of the annotators as an important factor while querying the data. But opportunistically selecting reliable user compulsively is not a good approach as it may annoy the annotator. For this reason, cost-sensitive crowdsourcing has been gaining attention recently.

XI. CONCLUSIONS

In this paper we have proposed and validated an active learning inspired adaptive activity recognition model to deal with the difficulties of collecting large amount of ground truth information. Our proposed model can discover unseen new activities and include the new activity class in the supervised learning model reliably. Prior works only considered a set of pre-defined activities, whereas our model is dynamic and can discover unseen new activities spontaneously. We have also validated our model in a real life dynamic environment and the experimental results show that our model outperforms traditional active learning approaches. Our results also show that active learning with pre-clustering can accelerate the informativeness measurement and supports faster convergence to optimal accuracy in presence of our proposed model. In the current version of our work, we ran our experiments in a single inhabitant environment and chose to monitor the activities which are disjoint. In future we want to investigate active learning in a multi-inhabitant environment and clustering of similar activities in nature using overlapped clustering in agnostic settings. We have also discussed about the data representation technique for active learning and crowdsourcing as data representation is very important for making active learning applicable. In future we also want look into more transparent data representation technique for building an effective activity recognition framework.

REFERENCES

[1] T. Labeodan, W. Zeiler, G. Boxem, and Y. Zhao, "Occupancy measurement in commercial office buildings for demand-driven control applicationsa survey and detection system evaluation," *Energy and Buildings*, vol. 93, pp. 303 – 314, 2015.

[2] A. Corna and et al., "Occupancy detection via ibeacon on android devices for smart building management," in *DATE '15*, pp. 629–632.

[3] M. M. Baig and H. Gholamhosseini, "Smart health monitoring systems: an overview of design and modeling," *J Med Syst*, vol. 37, no. 2, p. 9898, 2013.

[4] C. Perera, C. Liu, and S. Jayawardena, "The emerging internet of things marketplace from an industrial perspective: A survey," *Emerging Topics in Computing, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[5] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Mach. Learn.*, vol. 15, no. 2, pp. 201–221, 1994.

[6] H.-T. Cheng, F.-T. Sun, M. L. Griss, P. Davis, J. Li, and D. You, "Nuactiv: recognizing unseen new activities using semantic attribute-based learning." in *MobiSys*, pp. 361–374.

[7] D. Tran and A. Sorokin, "Human activity recognition with metric learning," in *Computer Vision–ECCV 2008*. Springer, 2008, pp. 548–561.

[8] W. Kerr, A. Tran, and P. Cohen, "Activity recognition with finite state machines," in *IJCAI'11*, 2011, pp. 1348–1353.

[9] T. Diethe, N. Twomey, and P. Flach, "Bayesian active transfer learning in smart homes," in *Advances in Active Learning : Bridging Theory and Practice, ICML 2015*, 2015.

[10] R. Liu, T. Chen, and L. Huang, "Research on human activity recognition based on active learning," in *ICMLC*, vol. 1, 2010, pp. 285–290.

[11] M. Stikic, K. Van Laerhoven, and B. Schiele, "Exploring semi-supervised and active learning for activity recognition," in *ISWC*, 2008, pp. 81–88.

[12] Y. chen ho, C. hu lu, I. han chen, S.-S. Huang, C.-Y. Wang, and L. chen fu, "Active-learning assisted self-reconfigurable activity recognition in a dynamic environment," in *ICRA*, 2009, pp. 813–818.

[13] W. S. Lasecki, Y. C. Song, H. Kautz, and J. P. Bigham, "Real-time crowd labeling for deployable activity recognition," in *CSCW*, 2013, pp. 1203–1212.

[14] E. Hoque and J. Stankovic, "Aalo: Activity recognition in smart homes using active learning in the presence of overlapped activities," in *PervasiveHealth*, 2012, pp. 139–146.

[15] H. T. Nguyen and A. Smeulders, "Active learning using pre-clustering," in *In Proceedings of the 21st International Conference on Machine Learning.*, 2004, pp. 623–630.

[16] S. Dasgupta and D. Hsu, "Hierarchical sampling for active learning," in *ICML*, 2008, pp. 208–215.

[17] A. McCallum and K. Nigam, "Employing em and pool-based active learning for text classification," in *ICML*, 1998, pp. 350–358.

[18] P. Donmez and J. G. Carbonell, "Paired sampling in density-sensitive active learning." in *ISAIM*.

[19] B. Settles, "Active learning literature survey," Tech. Rep., 2010.

[20] S. Hanneke, "Theory of disagreement-based active learning," *Found. Trends Mach. Learn.*, vol. 7, no. 2-3, pp. 131–309, 2014.

[21] M. Aminian, "Active learning for reducing bias and variance of a classifier using jensen-shannon divergence," in *Machine Learning and Applications, 2005. Proceedings. Fourth International Conference on*, 2005.

[22] Z.-H. Zhou, Y.-Y. Sun, and Y.-F. Li, "Multi-instance learning by treating instances as non-i.i.d. samples," in *ICML*, 2009, pp. 1249–1256.

[23] S. Sun and D. R. Hardoon, "Active learning with extremely sparse labeled examples," *Neurocomput.*, vol. 73, no. 16-18, pp. 2980–2988, 2010.

[24] M. B. et al, "Agnostic active learning," *J. Comput. Syst. Sci.*, vol. 75, no. 1, pp. 78–89, 2009.

[25] A. Beygelzimer and et al, "Importance weighted active learning," in *ICML*, 2009, pp. 49–56.

[26] S. Javdani and et al, "Near optimal bayesian active learning for decision making," *CoRR*, 2014.

[27] C. Elkan, "Using the triangle inequality to accelerate k-means," 2003.

[28] S. C. H. Hoi, R. Jin, and M. R. Lyu, "Large-scale text categorization by batch mode active learning," in *WWW*, 2006, pp. 633–642.

[29] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *J. Mach. Learn. Res.*, vol. 2, pp. 45–66, 2002.

[30] R. Yan, J. Yang, and A. Hauptmann, "Automatically labeling video data using multi-class active learning," in *ICCV*, 2003.

[31] P. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, no. 1, pp. 53–65, 1987.

[32] "Umbc-umb partnership awards a catalyst for collaboration: Research at umbc news feb 17 2015," http://research.umbc.edu/umbc-research-news/?id=49901.

[33] "Cloudengines pogoplug," https://pogoplug.com/.

[34] L. Zhao, G. Sukthankar, and R. Sukthankar, "Robust active learning using crowdsourced annotations for activity recognition." in *Human Computation*. AAAI.

[35] W. S. Lasecki, Y. C. Song, H. Kautz, and J. P. Bigham, "Real-time crowd labeling for deployable activity recognition," in *Proceedings of the 2013 conference on Computer supported cooperative work*, 2013, pp. 1203–1212.

[36] S. Bagaveyev and D. J. Cook, "Designing and evaluating active learning methods for activity recognition," in *UbiComp*, 2014, pp. 469–478.