Reasoning and Learning in Real-World Systems for Long-Term Autonomy (LTA)
Papers from the AAAI 2018 Fall Symposium

SocialAnnotator: Annotator Selection Using Activity and Social Context

H M Sajjad Hossain and **Nirmalya Roy**
Departmenf of Information Systems
University of Maryland Baltimore County

Abstract

Precise and eloquent label information is fundamental for interpreting the underlying data distributions distinctively and training of supervised and semi-supervised learning models adequately. But obtaining large amount of labeled data demands substantial manual effort. This obligation can be mitigated by acquiring labels of most informative data instances using *Active Learning*. However labels received from humans are not always reliable and poses the risk of introducing noisy class labels which will degrade the efficacy of a model instead of its improvement. In this paper, we address the problem of annotating sensor data instances of various Activities of Daily Living (ADLs) in smart home context. We exploit the interactions between the users and annotators in terms of relationships spanning across *spatial* and *temporal* space which accounts for an activity as well. We propose a novel annotator selection model *SocialAnnotator* which exploits the interactions between the users and annotators and rank the annotators based on their level of *correspondence*. We also introduce a novel approach to measure this *correspondence* distance using the *spatial* and *temporal* information of interactions, type of the relationships and activities. We validate our proposed *SocialAnnotator* framework in smart environments achieving $\approx 84\%$ statistical confidence in data annotation.

Introduction

Acquiring labeled data instances is an important task for training supervised and semi supervised machine learning models. In most of the problem domains, both domain knowledge and label information for a learning algorithm are compiled by the human annotators. As a result human intervention is indispensable for collecting ground truths. Labeling large amount of data suggests engaging more domain experts or extending the time for the labeling process. Adapting either of these approaches is a daunting task as it is difficult to find abundant domain experts who can relentlessly provide labels. Consider building an Activities of Daily Living (ADLs) classifier using accelerometry data. If the sampling frequency is 30 Hz and we collect data from a single user for a single day, we end up with approximately 2.5 million data instances. Moreover, the reliability, availability of domain experts, and the incurring costs associated with data annotation process makes it a painstaking step while building a machine learning model. It is possible to reduce the complexity of data annotation by dissecting the problem domain and identifying the relevancy of data with appropriate activity. For example, in case of ADLs we can select a handful number of activities or emphasize more on a specific period of the day instead of considering all of the data. From a machine learning perspective, we can view this as to look for most important data instances which can have significant impact on our classifier. By utilizing Active Learning (Bodó, Minier, and Csató), we can select the most informative data instances and pose the label queries to the annotators. There are alternative methods to reduce annotation effort other than Active Learning like utilizing *Crowdsourcing* platforms (Love and Hirschheim) or training the learning model using unlabeled data (Fiorini, Cavallo, and et al. 2017) (Gjoreski and Roggen 2017). However these approaches can invoke negative impact, for example, annotators in Crowdsourcing platforms are mostly not domain experts and can introduce noisy labels in the model.

Activity recognition using wearable and ambient sensors in smart home domain is a well studied problem in literature (Guan and Plötz 2017)(Shoaib et al. 2015). Existing activity recognition methods endure limitation in terms of data scarcity and scalability. The sensors produce an immense volume of data due to high sampling frequency in order to capture fine-grained information without any loss. In order to collect ground truth information, existing works have relied on the video feed heavily where each video frame is mapped with the timestamp (Hossain, Khan, and Roy 2017a). In this paper, we propose an online annotator selection model while exploiting active learning in smart home activity recognition domain. Even though active learning can be effective in acquiring labels, its foundation is built on impractical assumptions - *an annotator who is always available to provide the correct labels to every queries without incurring any cost and the active learner can query as many instances as possible* (Donmez and Carbonell 2008). In practical, a single annotator may or may not respond to all of the queries. Therefore, exploiting multiple annotators seems more practical (Yang and Wooldridge 2015), nevertheless their expertise level may differ drastically. Moreover, the labels received from these imperfect annotators are not always reliable, so if we pose an important query to a wrong annotator all the efforts will be pointless. Thus based on the informativeness of the selected data instance, it is always desirable to pose the query to the right annotator.

We first attempt to model the human relationships and assemble a knowledge base to represent the level of influence to others according to social and physical context. We formulate a corresponding distance metric using this knowledge base which expresses the level of $correspondence$ between connected users given the current context. While calculating this distance, we also consider the distance between activity space for a certain user. Activity distance enables us to find out similar activities with respect to their spatial and temporal properties. For example, if a person eats and watches television in the living room most of the time, then *eating* and *watching television* has similar spatial property. In such cases if an annotator can label *eating* activity efficiently, then our assumption is that he can also effectively provide label for *watching television* activity. In this paper we design an annotator selection model *SocialAnnotator* based on Contextual Multi-Armed Bandit (CMAB) algorithm where the *context* resembles the features of the queried instance and action corresponds to the annotator selection. We consider the time, space, prior context history and the approximate label received from the learner as the *context* information for CMAB. SocialAnnotator works in a collaborative manner where the connected users collaborate and provide label for each other.

Related Work

Activity recognition has been one of the core research areas in ubiquitous computing field for many years (Lara and Labrador 2013) (He et al. 2008). This rapid surge and advancement in learning activity pattern have also assisted a plethora of application domains ranging from sports (Daiber and Kosmalla 2017) (Hossain, Khan, and Roy 2017b) to health analytics (Samarah et al. 2017). Activity recognition research have been addressed from two pespectives using *computer vision* (Li and Vasconcelos 2017) (Jalal et al. 2017) and *sensor modalities* (Hossain, Khan, and Roy 2017a) (Davila, Cretu, and Zaremba 2017). Various machine learning models including both shallow learning (Lee and Cho) (Wyatt, Philipose, and Choudhury 2005) and deep learning (Guan and Plötz 2017) (Bhattacharya and Lane 2016) algorithms have been exploited in existing activity recognition literature over the years. Activity recognition models exploiting supervised and semi-supervised learning algorithms have to heavily rely on the number of labeled data instances. Some literature have proposed models using unsupervised learning algorithms (Twomey et al. 2017) (Münzner et al. 2017) (Bouchard, Bouchard, and Bouzouane 2012) but if the distribution of the data is not clearly inherent, unsupervised algorithms fail to find the pattern in the data.

To address the problem of gathering ground truth information, active learning has been employed by few researchers. The authors of (Bagaveyev and Cook 2014) investigated several active learning approaches in smart home activity recognition context and evaluated with real world data set. Diethe et al. proposed a bayesian approach by utilizing active and transfer learning in (Diethe, Twomey, and Flach 2016). In (Liu, Chen, and Huang 2010) and (Stikic, Van Laerhoven, and Schiele 2008) the authors exploited uncertainty based active learning. (Ho et al. 2009) used an entropy based approach to measure the informativeness of data instances. In our previous work (Hossain, Khan, and Roy 2017a) we proposed a clustering based heuristic to find the most informative instances. Hasan et al. proposed a context aware model using active learning (Hasan and Roy-Chowdhury 2015). They utilized entropy and mutual information of the instances to filter out the most informative data instances. (Xu et al. 2014) applied active learning in a contextual multi-armed bandit setting to do the activity classification. However, while employing active learning it is not always guaranteed to receive correct and noise free labels (Donmez and Carbonell 2008). In this paper we take a radically different approach than the existing literature and focus on improving the impact of active learning by selecting proper annotator using social relationships.

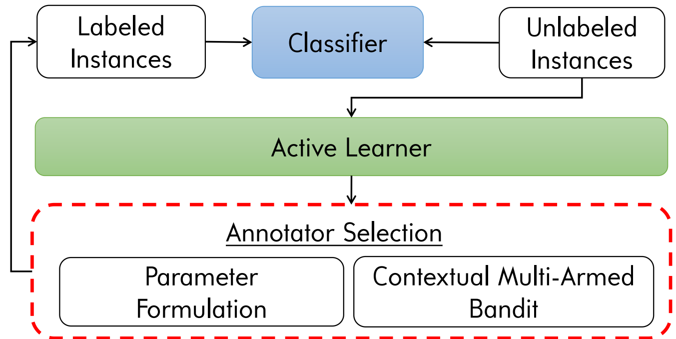


Figure 1: *A high level structure of the modules in SocialAnnotator framework.*

Overall Framework

SocialAnnotator framework is composed of three major components. We have an activity recognition classifier which is trained on labeled data instances from wearable devices that provide raw accelerometer data. After building a stable classifier, we start feeding unlabeled instances and predict the class label. We then start filtering uncertain data instances from the stream of unlabeled data instances in our *Active Learner* module. In our active learner module, we measure the entropy of the instances and select the instance with maximum entropy. We then send the selected instance to our *Annotator Selection* module. Note that in our daily life we interact with a number of people. The interaction can be physical or virtual through social network but every interaction is an opportunity to observe and share information. The key insight here is that we are connected and have more interactions with the people who we are related with us. These connected people might be direct witnesses of what we are doing in our day to day life. As a result these social relationships and correspondence lead us to have knowledge about the activity patterns of the people we are connected with.

We model the level of correspondence using a distance parameter. We calculate the *spatio-temporal* distance between two connected users using their probability distribution of location. The *spatio-temporal* distance lets us know about the intersection between their location distribution. We also incorporate a weight based on the strength of the relationship. The people with whom we interact more have higher potential to know about our daily routines. After formulating the distance parameter, we model a budget constrained context aware multi-armed bandit. The task of the bandit is to select annotator given the distance parameter and context. We design the bandit in such a way so that it does not act in a greedy way by introducing costs associated with each annotator and a budget constraint. We adapt a game theoretic approach where we have to ensure maximum gain and keep track of our budget as well. The costs associated with the annotators are not static, as the level of interaction evolves over time. For example, we have regular interactions with the people at our work place during the week days, but over the weekend we tend to mingle with close friends. Also a person can be connected through multiple relationships (close friend and colleague in a work place at the same time). Therefore, we consider the cumulative relationship weights of all the relations for quantifying the level of correspondence between two users. Figure 1 depicts a conceptual structure of our *SocialAnnotator* framework.

Distance Metrics

In this section we discuss the metrics which correlate our annotators with activities. We collect raw accelerometer and location data from the users and formulate the following distance metrics using the temporal and spatial information of these sensor modalities.

Spatio Temporal Distance

We calculate the spatio-temporal distance of the related users when an activity is performed. This distance metric implies if an user has any knowledge about the performed activity by another user he or she is related to. While computing this parameter, we also consider the neighboring locations of where the activity was performed. We calculate the likelihood of each related user j being in a location l_i given the current context x_t. We process this as a *categorical distribution*. Let us consider a set of activities W with whom the user u_i is connected. The location of each activity is an observation of our distribution, and the location set $L(W)$ is a sample of that distribution with cardinality m. Each location in $l_i \in L(P)$ has a prior probability. We denote the probabilities of locations as vector $p = (p_1, p_2, p_3 ... p_m)$. Let us consider q be the location probability distribution of an annotator a_i who is connected to user u_i through a relation. We then calculate the conditional distributions of p and q given context x_t and time t. Using these conditional distributions we calculate the distance between them using *Bhattacharyya distance* (Bhattacharyya). The distance between these two conditional distributions is defined as:

$$d_{st}(p(x,t), q(x,t)) = -ln(\mathbb{B}(p(x), q(x))) = \sum_{i=1}^{m} \sqrt{p_i(x,t) q_i(x,t)} \quad (1)$$

In Eqn 1, $\mathbb{B}$ is the *Bhattacharyya coefficient* which provides the measurement of overlap between the two probability distributions. This distance provides us information regarding the annotators who reside closer to the user. We calculate this spatio-temporal distance for all the connected users and take the annotators who were closest to the vicinity where the activity was performed. If no annotator was present in the vicinity where the user performed the activity, we assume that annotators dwelling in the neighboring locations may have knowledge about the activity label.

Activity-Activity Distance

We exploit the connectivity among activities to filter appropriate annotators. Our intuition is that if the properties of an activity W_i prevails in a similar spatial and temporal space to another activity W_j and an annotator a_k has efficiently provided reliable labels to activity W_j then a_k is a potential annotator who can provide label of activity W_i. To calculate this distance we consider three components of an activity pair - $correlation$, $spatial$ and $temporal$. Correlation calculates the co-occurrence frequency of the activity pair, the spatial and temporal component models the probability of an activity pertaining to the same location and time constraints. The distance is defined as:

$$d(w_i, w_j) = f(w_i, w_j) \mathcal{N}(||t_{a_i} - t_{a_j}||^2, \mu_t, \sigma_t) \mathcal{N}(||l_{a_i} - l_{a_j}||^2, \mu_s, \sigma_s) \quad (2)$$

In eqn 2, $f(w_i, w_j)$ denotes the co-occurrence frequency between a pair of activity, $l_{a_i}, l_{a_j}, t_{a_i}, t_{a_j}$ are the spatial and temporal parameters of the associated activities.

Relationship Weight

The strength of the social relationship can be integral in selecting annotator. There may not be any annotator who directly witnessed the user doing an activity. However, human being follows a cognitive routine most of the time and the persons mostly associated with his life are acquainted with the routine. For example, the family members living with the user are usually more familiar with his routine. Some annotators can also be remotely connected (e.g. updates on social network, talking over the phone or even playing online games together). So certain relationships provide more emphasis and demand more attention while choosing the annotator. For this reason, we try to provide weight to each connected user according to the relation. However this weight can not be static for all of the users as in real life not all relationships are same and they evolve over time. For example, consider the relationship with your office colleagues, initially they could be just colleagues but over time some might become your close friend. On the other hand one might be in touch with their parents on regular basis, but a different person might not. So for each person the weight of relationship is different. We use the relationship intensity strength proposed in (Srba and Bieliková 2010) to model our relationship weight. The interaction between two users (e.g. phone call, messaging, meeting etc.) or shared information (e.g. playing soccer together, common hobby) are designated as "rate factor". Depending on the social aspect these rate factors regulates the strength of a relationship. The partial relationship weight between user k and j for one factor is defined as:

$$Y_f(k, j) = \frac{\omega_{kj} \sum_{i=1}^{l} f_t}{1 + ln(1 + l_c)} \quad (3)$$

In eqn 3 ω_{kj} is the weight of the rate factor, l is the count of rate factors, l_c is the count of instances of the rate factor and f_t models the time influence. The final weight $Y(k,j)$ is measured by taking the arithmetic mean of the partial weights of all the rate factors.

Now that we have formulated all our distance metrics, we now define our final user to user distance metric. The activity-activity distance metric provides the distance between activities and finds the similarity among them. We maintain the count of such activities for which the annotators have performance score more than a pre-defined threshold δ. We utilize this count as an additional weight W_c for the annotators. The final distance is calculated using the following equation:

$$D(k,j) = Y(k,j)\, W_c\, d_{kj}\Big(p(x,t), q(x,t)\Big) \tag{4}$$

Methodology

In this section, we discuss active learning, the contextual multi armed bandit problem and the modeling of arms or actions of the bandit, the rewards and the context of our problem domain.

Active Learning

Active learning is fitting for problems pertaining to large amount of unlabeled data instances. In the context of activity recognition using wearable devices, we have to process overwhelming number of data instances which makes active learning befitting. We only label the data instances which provide highest gain which is reducing the generalized error of our classifier. In our proposed model we propose to use Active learning using pool-based sampling as we receive a stream of data in a very short period of time. We select a data instance from a pool of instances in a greedy way. Queries are typically conforming to the measure of uncertainty. Here our assumption is that the instances which are least certain are close to the decision boundary and labeling these instances will provide maximum gain. To measure the uncertainty we calculate the entropy of the provided instances and query the instance with maximum entropy. We calculate the maximum entropy and select an instance by following equation

$$\begin{aligned} x_H &= \underset{x}{\operatorname{argmax}}\, H_\theta(Y|x) \\ &= \underset{x}{\operatorname{argmax}} - \sum_y P_\theta(y|x) log\, P_{theta}(y|x) \end{aligned} \tag{5}$$

Contextual Multi-Armed Bandit

A contextual bandit problem is composed of N arms or actions. In our context an action refers to selecting an annotator. Our goal is to maximize this reward in each iteration. However, by selecting a sub set of the actions in a regular manner might always provide maximum reward. For example, a person's spouse or close friend has better idea about his daily activity routine than any one else. So by selecting the spouse or close friend in each round will maximize the reward outcome. If we consider the annotators as resources, prompting the same set of annotators will lead to resource exhaustion. In order to tackle this, we introduce a resource constraint or budget for each of the annotator. The annotators who ensures higher potential reward, incur higher cost. As a result given an overall budget our aim is to maximize the total reward while ensuring aggregated resource consumption remains bounded by a given budget. Let us denote the action set as $\mathcal{A} = \{a_1, \cdots, a_k\}$. We consider the cardinality of $\mathcal{A}$ to be finite as an user is connected to a finite number of people. A d-dimensional feature vector $x_t \in \mathcal{X}$ denotes the context information received at time t. At each time t, an agent or policy π decides to choose an action a_i based on the context x_t and receives reward r_i^t. The history of taken actions and received reward is denoted by $\mathcal{H}_{t-1} \rightarrow \{a_i(\tau), r_\tau, x_i(\tau)\}$ for $i = 1, \ldots, N$ and $\tau = 1, \ldots.., t-1$ where $a_i(\tau)$ denotes the chosen action which generated reward r_τ. The reward of an action is generated from an unknown distribution regulated by the given context. Let us consider the optimum action at t is a_i^* and its corresponding reward is $\tilde{r}_i^t$.

We want to select the action which results in reward close to the optimum one, so the aim is to maximize the reward in each step and minimize the difference between the overall optimum reward and the reward received. The difference between the optimal reward and the aggregated reward received is called *regret*. We provide a formal definition of *regret* as following

$$\mathbb{R} = \sum_{t=1}^{T} \mathcal{R}_t = \sum_{t=1}^{T} (\tilde{r}_i^t - r_i^t) \tag{6}$$

In this eqn, $\tilde{r}_i^t$ is the optimum reward at step t and r_i^t is the reward received. Let us define our reward function as $r_t = f(x_t, a_i(t))$, where $f(x_t, a_i(t))$ is the reward mapping function for arm $a_i(t)$. In order to maximize the reward function, the agent needs to learn the underlying function f which maps the context to action. In order to acquire knowledge about the latent function f, the agent has to explore other actions instead of choosing the optimum action which provides the best outcome. ϵ is our exploration parameter. The predictive distribution of our reward function depends on the current context and the history of actions taken. This is a normal distribution with mean μ_r and variance V which are defined as following

$$p_\theta(r_t | \mathcal{H}_{T-1}, x_t) = \mathcal{N}(\mu_r(t), V_t) \tag{7}$$

Each action $a_i(t)$ is also associated with a cost $c_{a_i}^t$. The cost associated with an annotator is variable in each round as the distance between users defined in eqn 4 varies over time. The costs are independently and identically drawn from an unknown continuous distribution with mean μ_c. We adhere to the same settings in (Xia, Qin, and et al.): (i) the rewards of an action are independent of its costs (ii) the rewards and costs of an arm are not influenced by other actions (iii) the rewards and costs of an action are independent and identically distributed at each iteration. Let us define a known parameter, *budget* B which designates the number of time the algorithm can invoke annotators. This budget constrain also helps us to supervise the stopping time $t_s(B)$ of our algorithm which is defined as following

$$\sum_{i=1}^{t_s(B)} c_{a_i}^i \leq B < \sum_{i=1}^{t_s(B)+1} c_{a_i}^i \tag{8}$$

Let us denote $\mathcal{R}^*$ as our optimum aggregated reward at stopping time $t_s(B)$. We calculate the expected regret, evaluated over the randomness of rewards and costs by modifying eqn 6.

$$\mathbb{R} = \mathcal{R}^* - E\Big[\sum_{t=1}^{t_s(B)} (r_{a_t}^t)\Big] \tag{9}$$

Actions The action space for an user is proportional to the number of connected annotators. An action corresponds to selecting an annotator from the correspondence vector M. Each element $m_{ij} \geq 0$ is congruent to how relevant the annotator is with respect to the user in terms of our distance metric and labeling accuracy. The expected reward to cost ratio of an annotator a_i is $\rho_{a_i} = \frac{\mu_r^{a_i}}{\mu_c^{a_i}}$. According to (Xia, Qin, and et al.), if both reward and cost distribution of an action is known, pulling the arm with maximum ρ can provide the expected reward as the optimal algorithm. When the distributions are unknown, we should select the annotator with the maximum ρ and also ensure exploration on the other rarely selected annotators.

Context A context vector x_t portrays the features and characteristics of each annotator. The features considered in a context vector are the timestamp t, location s, n performance metrics of the annotator with respect to each activity $c_1, \ldots, c_n$. We do not include the sensor data in the context vector.

Reward Our reward mapping function randomly generates reward according to the conditional probability measure defined in eqn 7. Initially the model is uncertain about the value θ. Our reward mapping function f is defined to measure the reduction in variance of our classification model between two iterations. For making things simple, our objective is to minimize the squared loss of the true label and the label received from an annotator. We define our expected error as following

$$\begin{aligned}\mathbb{E}\Big[(\hat{y} - y)^2 | x_t, y_l\Big] &= \mathbb{E}_{Y|x}\Big[(y - \mathbb{E}_{Y|x}[y|x, y_l])^2\Big] \\ &+ (E_L[\hat{y}] - \mathbb{E}_{Y|x}[y|x])^2 + E_L\Big[(\hat{y} - E_L[\hat{y}])^2\Big]\end{aligned} \tag{10}$$

In eqn 10 $\mathbb{E}_L[.]$ is the expectation over the labeled training set L, $\hat{y}$ is the label received from an annotator and y is the true label of the instance. $\mathbb{E}_{Y|x}\Big[(y - \mathbb{E}_{Y|x}[y|x])^2\Big]$ indicates noise or uncertainty of y given x. The second term represents bias which is the error due to the selected action. The third term represents the output variance of our model. Therefore minimizing the variance will ensure to minimize the generalization error of our model. So we try to reduce error by selecting annotators that establish highest variance reduction of our activity recognition model. For any action a_i, number of times it is invoked $n_{a_i,t}$, average cost $\bar{c}_{a_i,t}$ and average reward $\bar{r}_{a_i,t}$ and the exploration parameter is $\epsilon_{a_i,t} = \sqrt{\frac{2log(t-1)}{n_{i,t}}}$. We calculate index $D_{a_i,t}$ for each annotator:

$$J_{a_i,t} = \frac{\bar{r}_{a_i,t}}{\bar{c}_{a_i,t}} + \frac{\bar{r}_{\epsilon_i,t}}{\bar{c}_{a_i,t}} + \frac{\bar{r}_{\epsilon_i,t}}{\bar{c}_{a_i,t}} D(k,i) \tag{11}$$

In eqn 11, the average reward to cost ratio represents the exploitation. The first influences our algorithm to choose the arms with higher rewards. The exploration term $\frac{\bar{r}_{\epsilon_i,t}}{\bar{c}_{a_i,t}}$ favors the annotators who provide less reward and as a result invoked infrequently with lower costs. Exploring weaker annotators may be conducive as our budget is limited. The final term enforces joint exploitation and exploration. Our whole methodology is summarized in Algorithm 1.

Algorithm 1 *SocialAnnotator* Annotator Selection

Require: U, A pool of unlabeled instances $\{(x)^u\}_{u=1}^{U}$,
$A = \{a_1, a_2, \ldots a_k\}$, A list of connected annotators

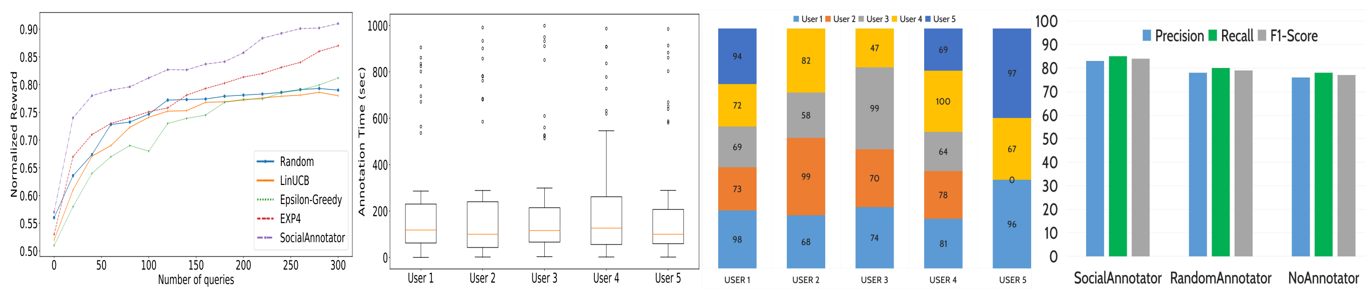
```
Output: Best annotator a_i.
Select instance x_t with maximum entropy
p ← location probability distribution of the user
Reward Index J ← {}
for annotator a_k ∈ A do
    q ← location probability distribution of annotator a_k
    Calculate spatio temporal distance d_s t(p(x_t, t), q(x_t, t))
    Distance d ← {}
    for each activity w_i do
        d(w_i, w_j) ← activity-activity distance
        d.insert(d(w_i, w_j))
    end for
    maximum activity-activity distance d_m ax ← max(d)
    Y_f(a_k, user) ← relationship weight
    D(a_k, user) ← annotator-user distance
    J_{a_k} ← reward index for annotator a_k
    J[k] = J_{a_k})
    Maximum reward J_max = max(J)
    i ← index of J_max
end for
return annotator a_i
```

Experimental Evaluation

In this section we evaluate *SocialAnnotator* using real data traces and compare the performance of our model using different bandit algorithm. We also evaluate our classifier based on annotator We provide a description of our setup and dataset and data collection process in the following:

Setup

We collected activity data using wearable devices from 5 users over the course of 16 days. We used android smart watch Moto360 to collect the accelerometer data. We also collected the location information of the users using GPS which we only used for ground truth. We developed smartphone apps for both ios and android platforms using which the users can add correspondence (friend, spouse, roommate etc.). Users were asked to log the interaction, location and activity data using this platform. Users were asked to log not only the in person interactions but also virtual or remote (messaging, talking over the phone, interaction through social network etc.) interactions as well. Logging too much



(a) ***(b)*** ***(c)*** ***(d)***

Figure 2: *(a) shows the distribution of normalized reward with respect to number of queries. (b) shows per-annotator labeling time distribution. (c) represents the stack plot of percentage of correctly labeled instances of all the connected users for each user. (d) Precision, recall and f1-score of our base classifier using different settings.*

interaction can become a burden, so only the number of interaction was required to log. We established ground truth using these log information. In our experiment we monitored 5 daily activities - {*eating, sleeping, phone calling, working, cooking*}.

The sensor data are directly uploaded to our lab server from the wearable device and we preprocess(feature extraction, filtering, noise reduction etc.) the data in the server. As previously stated in our pipeline in Figure 1, we train a supervised classifier first to recognize the performed activity. We have used a simple decision tree based classifier. Initially after training our model with labeled instances we achieved an average accuracy of 77%. Even if we have achieved low accuracy compared to the existing literature, we are only concerned about investigating how efficient labeling can help to improve the performance of activity recognition model. We have For our budgeted multi-armed bandit, the reward and cost of each annotator are sampled from a beta distribution. The parameters of the these distributions are sampled from [1,5]. The budget of our framework is chosen from the set {200, 300, 400, 500, 1000}. We compare our annotator selection model with different contextual multi armed bandit algorithm - LinUCB, ϵ-Greedy, EXP4 and Random sampling. In case of Random sampling, the annotator is chosen at random in each iteration. All the contextual bandit algorithms are executed up to 300 iterations per user in this experiment.

Table 1: *Statistics of SocialAnnotator compared to other multi-armed bandit algorithm*

	$\bar{r}$	$\bar{c}$	$\bar{r}/\bar{c}$	% opt
Random	0.763	0.793	0.962	1.67
LinUCB	0.758	0.814	0.932	0.8
ϵ-Greedy	0.796	0.886	0.8984	1.32
EXP4	0.864	0.810	1.067	61.17
SocialAnnotator	0.913	0.267	3.419	73.42

Bandit Performance

In Table 1, we list and compare average rewards($\bar{r}$), average costs($\bar{c}$), average reward to cost ratio ($\bar{r}/\bar{c}$) and the percentage of time optimal annotator gets selected (% opt) of different bandit algorithms. From the statistics we see that LinUCB and ϵ-Greedy perform worst with respect EXP4 and our model. Both these algorithms are not meant for problems with budget constaints and as a result they do not take budget into consideration. Our model can achieve higher reward at lower cost contrast to other bandits which verifies that we are choosing optimal annotator at each step. In figure 2a the trend of average reward obtained at each step is shown. It is evident that our proposed algorithm outperforms the other bandit algorithm settings. More context information like detailed interaction, fine grained location information etc. might further improve the model.

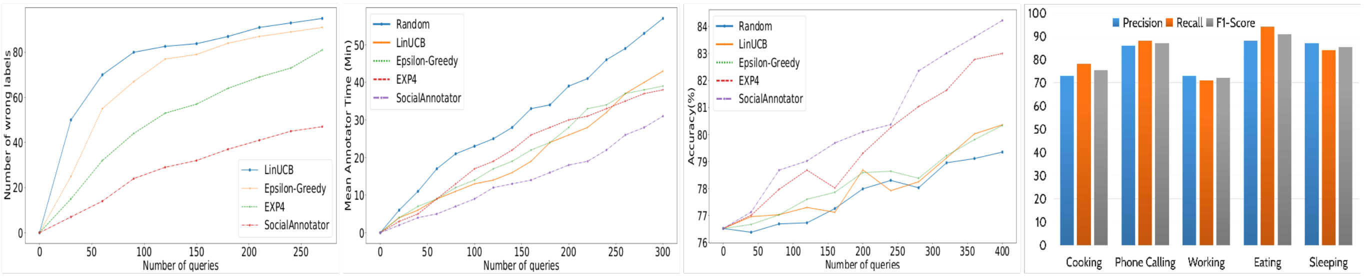
Annotator Selection

We monitor the performance of each annotator and maintain a score for each activity associated with the connected users. In Figure 2b we provide the annotation time distributions of each user using boxplot. A box depicts the majority of annotation times and the median time is marked with a solid line inside the box. It is noticeable from the figure that each user have different time distribution which means the efficiency, promptness and reliability of each user varies. We also deduce that the annotator might not provide the label at all. We show the percentage of correctly labeled instances by each user in Figure 2c. As *User 5* is only connected to *User 4* and *User 1* there are only three scores for him including the score of labeling his own activity. It is apparent that all the users are efficient in labeling their own activity.

Table 2: *Labeling result of each user*

	Correct Label	Wrong Label	No Label
User 1	324	49	27
User 2	306	68	26
User 3	285	83	32
User 4	310	73	17
User 5	345	37	18

We notice that *User 1* and *User 5* were able to label each others data quite precisely. We found that these two users were living in the same apartment and User 1 is *spouse* of User 5. Their quantity of interaction was also very high as apart from living together they were also talking with each other over the phone couple of times a day. We also notice from the figure that *User 2* and *User 4* were able to label the



(a) *(b)* *(c)* *(d)*

Figure 3: *(a) shows the number of wrong labels received in 300 iterations for different bandit algorithms. (b) per-annotator labeling time distribution (c) demonstrates the progression of accuracy after each iteration. (d) Precision, recall and f1-score of each activity.*

activity of each other with good accuracy (82% for *User 2* and 78% for *User 4*). After investigating into it, we observed that these two users were working together at the same place and had a lot of interactions. *User 3* and *User 4* also worked at the same place but they had very less interaction with each other which is reflected in their annotation efficiency. Receiving the label information as early as possible is aslo imperative. If we do not receive a label for the queried instance within a certain pre-defined threshold tme, we discard the annotator. As a result, if we pose the query to an user who may delay in providing the label or not provide the label at all, we not only spoil resources but also lose valuable information. In Figure 3b we show the mean annotation time needed using different bandit algorithm for varying number of queries. Our model exhibits lowest mean (30 mins) than all other approaches and getting the labels at the right time. As a result our model also ensures immediate result along with the conservation of information. To further validate our claim, we show the number of wrong labels received for different bandit algorithms in 300 iterations in Figure 3a. After 300 iterations, *SocialAnnotator* indicates lowest number of wrong labels, which proves that we are posing the queries to the right person at the right time. The cumulative labeling accuracy of each user is also described in Table 2.

Classifier Performance

We have achieved an overall accuracy of 77% for our base classifier. We trained our model with only 5% (1050) of the total labeled data instances. We apply active learning and incrementally query instances. We show the overall accuracy of our classifier after 400 iterations in Figure 2d. By employing *SocialAnnotator* we accomplish an average accuracy of $\approx$ 84% which is an improvement of 7% compared to our base classifier. *NoAnnotator* title demonstrates the results when we do not administer annotator selection. We notice that it only improve the accuracy by $\approx$ 1-2% even if we have applied active learning. So posing the query to the right person helps to improve the accuacy of our classifier. In Figure 3d we show the accuracy of individual activities after applying $SocialAnnotator$. We see that $Cooking$ and $Working$ show low accuracies with respect to other activities. After further investigation, we noticed that these two activities itself are very complex and experienced low accuracies in our base classifier as well (68% and 64%). However, *SocialAnnotator* actually increased the accuracy by $\approx$ 8-10%. Consequently $SocialAnnotator$ helps to improve the accuracies of complex activities which are hard to infer. Figure 3c shows the change of accuracy in 400 iterations. Our algorithm converges to optimum accuracy faster than other approaches.

Conclusion

In this paper we have proposed a novel annotator selection method $SocialAnnotator$, by exploiting social relationships among the users to improve the efficiency of active learning in activity recognition context. Our proposed model selects annotator based on the strength of the relationships and *spatio-temporal* distance metrics among the users. We also consider the similarities between the activities in our model to calculate the level of *correspondence* among the users. Prior works with active learning that propose to mitigate the labeling effort, have not considered the influence of annotators in their model. Our results show that, $SocialAnnotator$ can compliment active learning and establish reliabile, prompt and accurate label information. We have demonstrated that by using our methodology, we improved the accuracy of our base classifier by $\approx 7\%$. In our current approach while calculating the distance between two users, we only consider a very few interactions between them. In future we want to apprehend more interactions as well as more context information unobtrusively. We want to monitor the users phone usages and social network interactions without needing any feedback from them and add more sensor modalities like ambient infrastructure sensor to record the movements in detail. We also plan to do the regret anaylsis of our algorithm and derive the upperbound in future.

References

Bagaveyev, S., and Cook, D. J. 2014. Designing and evaluating active learning methods for activity recognition. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, UbiComp '14 Adjunct, 469–478.

Bhattacharya, S., and Lane, N. D. 2016. From smart to deep: Robust activity recognition on smartwatches using deep learning. In *IEEE International Conference on Pervasive Computing and Communication Workshops*.

Bhattacharyya, A. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society*.

Bodó, Z.; Minier, Z.; and Csató, L. Active learning with clustering. In *Active Learning and Experimental Design workshop, In conjunction with AISTATS 2010*.

Bouchard, K.; Bouchard, B.; and Bouzouane, A. 2012. Unsupervised discovery of spatial relationships between objects for activity recognition inside smart home. In *The 2012 ACM Conference on Ubiquitous Computing*, 655–656.

Daiber, F., and Kosmalla, F. 2017. Tutorial on wearable computing in sports. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*, 65:1–65:4.

Davila, J. C.; Cretu, A.; and Zaremba, M. B. 2017. Wearable sensor data classification for human activity recognition based on an iterative learning framework. *Sensors*.

Diethe, T.; Twomey, N.; and Flach, P. 2016. Active transfer learning for activity recognition. In *24th European Symposium on Artificial Neural Networks, ESANN 2016*.

Donmez, P., and Carbonell, J. G. 2008. Proactive learning: Cost-sensitive active learning with multiple imperfect oracles. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM.

Fiorini, L.; Cavallo, F.; and et al. 2017. Unsupervised machine learning for developing personalised behaviour models using activity data. *Sensors*.

Gjoreski, H., and Roggen, D. 2017. Unsupervised online activity discovery using temporal behaviour assumption. In *Proceedings of the 2017 ACM International Symposium on Wearable Computers, ISWC*.

Guan, Y., and Plötz, T. 2017. Ensembles of deep LSTM learners for activity recognition using wearables. *IMWUT*.

Hasan, M., and Roy-Chowdhury, A. K. 2015. Context aware active learning of activity recognition models. In *2015 IEEE International Conference on Computer Vision (ICCV)*.

He, Z.; Liu, Z.; Jin, L.; Zhen, L.-X.; and Huang, J.-C. 2008. Weightlessness feature x2014; a novel feature for single triaxial accelerometer based activity recognition. In *2008 19th International Conference on Pattern Recognition*.

Ho, Y.-c.; Lu, C.-h.; Chen, I.-h.; Huang, S.-s.; Wang, C.-y.; and Fu, L.-c. 2009. Active-learning assisted self-reconfigurable activity recognition in a dynamic environment. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1567–1572.

Hossain, H. M. S.; Khan, M. A. A. H.; and Roy, N. 2017a. Active learning enabled activity recognition. *Pervasive and Mobile Computing*.

Hossain, H. M. S.; Khan, M. A. A. H.; and Roy, N. 2017b. Soccermate: A personal soccer attribute profiler using wearables. In *IEEE International Conference on Pervasive Computing and Communications Workshops*.

Jalal, A.; Kim, Y.; Kim, Y.; Kamal, S.; and Kim, D. 2017. Robust human activity recognition from depth video using spatiotemporal multi-fused features. *Pattern Recognition*.

Lara, O. D., and Labrador, M. A. 2013. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys Tutorials*.

Lee, Y.-S., and Cho, S.-B. *Activity Recognition Using Hierarchical Hidden Markov Models on a Smartphone with 3D Accelerometer*.

Li, W., and Vasconcelos, N. 2017. Complex activity recognition via attribute dynamics. *International Journal of Computer Vision*.

Liu, R.; Chen, T.; and Huang, L. 2010. Research on human activity recognition based on active learning. In *2010 International Conference on Machine Learning and Cybernetics*, volume 1, 285–290.

Love, J., and Hirschheim, R. Crowdsourcing of information systems research. *EJIS*.

Münzner, S.; Schmidt, P.; Reiss, A.; Hanselmann, M.; Stiefelhagen, R.; and Dürichen, R. 2017. Cnn-based sensor fusion techniques for multimodal human activity recognition. In *Proceedings of the 2017 ACM International Symposium on Wearable Computers*, ISWC '17, 158–165.

Samarah, S.; al Zamil, M. G. H.; Al-Eroud, A. F.; Rawashdeh, M.; Alhamid, M. F.; and Alamri, A. 2017. An efficient activity recognition framework: Toward privacy-sensitive health data sensing. *IEEE Access* 5.

Shoaib, M.; Bosch, S.; Incel, Ö. D.; Scholten, H.; and Havinga, P. J. M. 2015. A survey of online activity recognition using mobile phones. *Sensors*.

Srba, I., and Bieliková, M. 2010. Tracing strength of relationships in social networks. In *Proceedings of the International Conference on Web Intelligence and International Conference on Intelligent Agent Technology*.

Stikic, M.; Van Laerhoven, K.; and Schiele, B. 2008. Exploring semi-supervised and active learning for activity recognition. In *Proceedings of the 2008 12th IEEE International Symposium on Wearable Computers*, ISWC '08, 81–88.

Twomey, N.; Diethe, T.; Craddock, I.; and et al. 2017. Unsupervised learning of sensor topologies for improving activity recognition in smart environments. *Neurocomputing*.

Wyatt, D.; Philipose, M.; and Choudhury, T. 2005. Unsupervised activity recognition using automatically mined common sense. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 1*, AAAI'05, 21–27.

Xia, Y.; Qin, T.; and et al. Budgeted multi-armed bandits with multiple plays. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*.

Xu, J.; Xu, J. Y.; Song, L.; Pottie, G. J.; and van der Schaar, M. 2014. Context-driven online learning for activity classification in wireless health. In *2014 IEEE Global Communications Conference*, 2423–2428.

Yang, Q., and Wooldridge, M., eds. 2015. *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI*. AAAI Press.