Discovery and Recognition of Unknown Activities

Juan Ye
University of St Andrews
St Andrews, KY16 9SX, UK
jy31@st-andrews.ac.uk

Lei Fang
University of St Andrews
St Andrews, KY16 9SX, UK
lf28@st-andrews.ac.uk

Simon Dobson
University of St Andrews
St Andrews, KY16 9SX, UK
simon.dobon@st-andrews.ac.uk

UbiComp/ISWC '16 Adjunct, September 12-16, 2016, Heidelberg, Germany
 ISBN 978-1-4503-4462-3/16/09...$15.00
DOI: http://dx.doi.org/10.1145/2968219.2968288

Abstract

Human activity recognition plays a significant role in enabling pervasive applications as it abstracts low-level noisy sensor data into high-level human activities, which applications can respond to. In this paper, we identify a new research question in activity recognition – discovering and learning unknown activities that have not been pre-defined or observed. As pervasive systems intend to be deployed in a real-world environment for a long period of time, it is infeasible, to expect that users will only perform a set of pre-defined activities. Users might perform the same activities in a different manner, or perform a new type of activity. Failing to detect or update the activity model to incorporate new patterns or activities will outdate the model and result in unsatisfactory service delivery. To address this question, we explore the solution space and propose an estimation-based approach to not only discover and learn new activities over time, but also benefit from no need to store any historic sensor data.

Author Keywords

Activity recognition; Online learning; Incremental learning; Hierarchical Clustering; Pervasive computing; Smart home

ACM Classification Keywords

H.1.2 [User/Machine Systems]: Human Factors

General Terms

Design, Algorithms, Experimentation, Measurement, and Performance

Introduction

Nowadays pervasive computing is less a vision but more a reality: we have witnessed many pervasive computing applications in our everyday life, such as health assessment (e.g., stress and depression detection [2], clinical assessment on cognitive and mobility scores [5]), activity-driven behaviour changing applications [4], home automation (e.g., automatic heating configurations [9]), and so on. Activity recognition lies at the heart of these pervasive computing applications [6], which is the ability to recognise and predict users' current and future activities from data collected from a wide range of sensors that are embedded in an environment such as RFID, infra-red positioning sensors, and that are worn on the users such as wearable sensors and mobile devices. Based on inferred user activities, applications are designed to deliver intended services in an automatic and unobtrusive manner.

Activity recognition has been a popular research topic in pervasive computing for the last decade, and most of the research focuses on how to accurately infer a set of *pre-defined activities of interest* using various machine learning and data mining techniques. There are two assumptions in a classic activity recognition process: *the set of activities* is fixed and *the user behaviour model* is fixed. However, neither of them holds when we deploy a system in an open and versatile environment, for a longer period of time (e.g., more than ten years), and target a wider range and a larger number of users. First of all, it is unrealistic to expect that users will only perform the pre-defined set of activities for a long time. When a user performs a new type of activity that has not been seen or learned during the training period, the pre-trained model might not be able to recognise it or misclassify it. Secondly, it is difficult, or sometimes infeasible to collect all possible user activities during the training period, as user behaviours tend to evolve with time. Both the above situations could result in the degradation of the performance of activity recognition techniques, and thus could lead to undesirable services to the users.

This paper aims to discover and recognise unknown activities that have not been pre-defined, initially programmed with the knowledge, or covered in a training data. This is a challenging research question. First of all, given that there might exist a wide range of patterns for a certain activity, it is difficult to distinguish a new type of activity or a new pattern of an existing activity. The requirements of learning unknown patterns is beyond the capabilities of most existing machine learning techniques. An incremental learning approach often starts with a very limited number of samples, and it often exhibits a learning curve. Also the inherently imperfection in sensor data adds extra complexity to the problem. Furthermore, a pervasive system requires to provide services in real time and often this system runs on resource-constrained devices (such as sensors or mobile devices). Therefore, it is impractical and unrealistic to store all the sensor data and analyse it in an offline batch mode, which cannot allow for quick response to real time sensor data. The goal of this paper is to propose an online incremental learning approach to address the above challenges.

We claim the following novelties and contributions. We explore the solution space of detecting unknown activities in an estimation-based technique. The proposed approach can accurately detect unknown activities and incrementally update the activity model to recognise new types of activities without the need of storing all the historic data. We have evaluated and compared the performance of the approach on a real-world publicly available dataset.

The rest of the paper is organised as follows. Section 2 introduces the relevant existing work, including general activity recognition, outlier detection, and user behaviour model evolution in other areas. Section 3 introduces the data processing on generating features from streaming sensor events. Section 4 presents the technique, whose performance will be evaluated and discussed in Section 5. In the end, we summarise the work and point to the future work in Section 6.

Related Work

In this paper, we propose to target discovering and recognising unknown activities, which has not been addressed in the community of pervasive computing, to the best of our knowledge. This work is relevant to, but not the same as, general activity recognition, outlier detection, and user profile evolution. In the following, we will introduce the literature in the above three areas and explain the difference and relevance.

Activity Recognition

Activity recognition based on wearable and environmental sensing technologies has been extensively researched in the last decades and a few recent surveys have broadly reviewed the existing techniques [1, 16, 3, 10]. In general, sensor-based activity recognition techniques can be grouped into knowledge- and data-driven approach, and the data-driven approach can be further classified into supervised and unsupervised learning techniques. A knowledge-driven technique leverage expert knowledge ranging from the early attempt on a small scale of common sense knowledge [11] to a more advanced and formal approach on a large scale of knowledge base such as ontologies [15] and WordNet [9, 13], and apply reasoning engines to infer activities from sensor data. A data-driven technique apply the off-the-shelf machine learning and data mining techniques to automatically establish the correlation between sensor data and activity labels. Hidden Markov Models and their derivatives are one of the most popular techniques [16, 17].

Learning unknown activity is relevant to unsupervised learning, but not the same. Unsupervised learning assumes a set of sensor data without the activity labels. A unsupervised learning technique will cluster the sensor data into different categories, and use expert knowledge to label the cluster with proper activities. Once the model is learnt, it is fixed. It does not necessarily deal with a sensor event that is not observed in the training data, nor incrementally update the model with new sensor observations.

Outlier Detection

Discovering unknown activity is an outlier detection problem; that is, detecting an observation that deviates from the models of the pre-known activities. Outlier detection is a long studied problem in data analysis, and has widely used in detecting abnormal sensor events in wireless sensor network [19]. The main types of outlier detection methods are model- and density- based. Model-based methods assume that the data follows a model that, excluding the specifics of its parameters, is known, so they focus on evaluating the model's parameters and singling out the outliers that obstruct their true value. In contrast, a density-based method does not assume the existence of a global model and uses the density of data near a data point.

Evolving User Profile Learning

Studying user behaviours is an interesting topic in computer security, web applications, and human computer interactions. The existing works are mainly about how to evolve from pre-existing profiles, not including how to detect a new profile. Iglesias et al. [8] proposes a technique to automatically creating and evolving user behaviour profiles. Here a user profile is the interactions with computer systems such as UNIX commands. The technique can discover a new profile and update an existing profile. For a given new sequence of commands, it will measure its potential to become a new profile. The potential represents the density of the data that surrounds a certain data sample, which is calculated as the accumulated distance between this sample to all the other samples in the profile. The authors have derive an incremental updating method to recursively calculate the potential without the need to store all the historic data.

Sensor Data Feature Extraction

In this section, we will introduce how we generate and represent features from streaming sensor events, which forms the basis of our proposed approaches. The features are general enough for any type of sensors, including numeric, binary, or featured sensor readings [16].

We segment sensor events into time slots of a fixed interval (say one minute). For each time slot, we extract features on sensor data and timestamps. A sensor feature vector is represented as $\boldsymbol{x} = [x_1, x_2, ..., x_S]$, where S is the number of sensors being installed, and $x_i (1 \leq i \leq S)$ is the normalised value on the ith sensor.

If a sensor is binary (e.g., RFID, switch sensor, or infra-red passive motion sensor), x_i is the frequency of this sensor being activated; that is, N_i/N, where N_i is the number of times the ith sensor being activated and N is the total number of sensor events reported in this time slot. If a sensor reports numeric values (e.g., a temperature or electricity consumption monitoring sensor), then we normalise the averaged values reported by this sensor in this time slot using the formula $(avg_i - min_i)/(max_i - min_i)$, where avg_i is the averaged values reported by the ith sensor in this time slot, and min_i and max_i are the minimum and maximum values that this sensor can produce, which can be obtained from the manufacture specification of the sensor or the historical records of the values reported by this sensor. If a sensor (e.g., accelerometers) has featured readings, its corresponding value x_i can be extended to a certain length (say three), rather than to be fixed as a single value.

Time is often a useful indicator of activities being undertaken [7]; for example, people have breakfast in the morning, and go to bed at night. There are several ways to extract the temporal features. All single dimensional linear mapping techniques have serious limitations. For example, the widely used 24-hour representation in-proportionally enlarge the difference between say 23:00 hour and midnight (00:00). Linear transformation like $\frac{|h-12|}{12}$ solves the above problem but also makes morning and afternoon hours indistinguishable, e.g. 8:00 and 15:00. To solve the above drawbacks, we project time into a two dimensional sphere, where distance between hours is represented by angle between the projected time vectors ; that is, the hour of the timestamp is represented as $(\cos\theta, \sin\theta)$, where $\theta = h \times (2\pi/24)$.

In the end, we combine the sensor and time features together in a D-dimensional feature vector, denoted as $\boldsymbol{x}$. As each dimension in $\boldsymbol{x}$ has independent semantics, we are more concerned with the angular difference at each dimension in $\boldsymbol{x}$, rather than as a whole. Therefore, we consider all the feature vectors as a directional vector, which leads us to choose the cosine distance to measure the similarity between feature vectors and the von Mises-Fisher model to estimate their distributions in the following section. As the vectors are directional, we normalise them to be a unit vector of norm one; that is, $|\boldsymbol{x}| = 1$.

Proposed Estimation-based Approach

We start with an intuitive way of constructing an activity profile – Single Centroid; that is, collecting all the feature vectors that map to one pre-known activity to form one single cluster and estimating its centroid vector. The assessment of whether a new vector matches to a profile depends on the closeness of this vector to the centroid. In the following, we will briefly introduce the four processing steps and focus on how we support incrementally updating.

Let $V = \{\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x}_n\}$ be a collection of feature vectors on a pre-known activity.

Creation Assume each activity can be characterised by a single centroid vector. Then the centroid vector defined as $\boldsymbol{c}$ (i.e., the mean direction [12]), together with the distance statistics such as the mean $\bar{d}$ and standard deviation σ are computed. The definition of the relevant statistics are listed below:

$$\boldsymbol{c} = \frac{\sum_{i=1}^{n} \boldsymbol{x}_i}{\| \sum_{i=1}^{n} \boldsymbol{x}_i \|}, \bar{d} = \frac{\sum_{i=1}^{n} d_i}{n}, \sigma^2 = \frac{\sum_{i=1}^{n} (d_i - \bar{d})^2}{n-1},$$

where d_i is a distance between a vector $\boldsymbol{x}_i$ and a centroid $\boldsymbol{c}$, and $\boldsymbol{c}$ is normalised to a unit length.

Assessment Given a new vector $\boldsymbol{x}$ that is formed from newly collected sensor events, we calculate its distance d to the centroid of a profile. If the distance is within p standard deviations to the mean distance, then we consider $\boldsymbol{x}$ matches the profile: $|d - \bar{d}| \leq p * \sigma$. If there exists more than one matched profile, we choose the profile with the minimum of the normalised distance differences; that is, $a \leftarrow \mathrm{argmin}_a \frac{|d - \bar{d}_a|}{\sigma_a}$. Then the activity a is an inferred activity label for this new vector $\boldsymbol{x}$.

Update The formulas presented in the creation step require all the collected feature vectors available to be calculated, which contradicts to our requirement on incremental online learning. In the following update step, we will introduce an online sequential way to compute them with constant space complexity and time complexity. The process works: re-calculate the centroid vector $\boldsymbol{c}$ based on the incremented data set $(V \cup \{\boldsymbol{x}_{n+1}\})$, and the distances of all the feature vectors to the new centroid vector c_{n+1}. The main challenge is the shift of the centroid vector whenever a new vector is introduced. To accommodate the requirement of not storing all the historic data, the following on-line updating procedures are derived. The algorithm works by keeping track of three on-line updating statistics from the data: n the total number of vectors, $\boldsymbol{\mu}_n = \sum_{i=1}^{n} \boldsymbol{x}_i$ and $\boldsymbol{M}_n = \sum_{i=1}^{n} \boldsymbol{x}_i \cdot \boldsymbol{x}_i^T$ a $D \times D$ matrix, where D is the dimension of $\boldsymbol{x}$.

When a new feature vector $\boldsymbol{x}$ arrives, the centroid, the mean and standard deviation distance can be updated as follows:

$$\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + \boldsymbol{x}, \quad \boldsymbol{M}_{n+1} = \boldsymbol{M}_n + \boldsymbol{x} \cdot \boldsymbol{x}'$$

$$\boldsymbol{c}_{n+1} = \frac{\boldsymbol{u}_{n+1}}{\| \boldsymbol{u}_{n+1} \|}, \quad \bar{d}_{n+1} = 1 - \frac{1}{n+1} \| \boldsymbol{u}_{n+1} \|$$

$$S_{n+1} = \boldsymbol{c}'_{n+1} \boldsymbol{M}_{n+1} \boldsymbol{c}_{n+1} - \frac{1}{n+1} \boldsymbol{u}'_{n+1} \cdot \boldsymbol{u}_{n+1}, \quad \sigma^2_{n+1} = \frac{S_{n+1}}{n}$$

The derivation of the above formula is listed in Appendix A.

Candidate A candidate pool consists of a set of candidate profiles, each of which accumulates vectors that do not match any pre-known profiles and are similar enough. As a candidate profile starts with a very limited number of samples (such as one), it cannot produce useful statistics such as the distance mean and standard deviation for assessing the similarity. To address this issue, we reuse the minimum of the mean and standard deviations of the distances learned on the pre-known profiles as a reference. The reason of choosing the minimum is that we want to make sure the vectors accumulated in one candidate are most likely to belong to one activity. Thus, later when we query the user about the activity label, we can just query once, rather than ask the user to confirm each vector in a candidate profile. The potential downside of this strict selection is that one unknown activity might have multiple candidate profiles; however, we consider this is reasonable as one activity can have different sensor patterns [18]. Once we have collected two vectors in a candidate profile, then we can produce the distance mean and standard deviation using their own vectors. To note that the vectors in a candidate profile are stored in memory, and they will be discarded when an activity profile is created from the candidate. We think this is less a problem as the number of vectors in a candidate profile is much less; e.g., 5 or 10, compared to the whole historic sensor observations. Also the number of the vectors in candidate can be decided flexible depending on the capacity of the hardware of hosting the reasoner or the application requirement. In our experiment, we set the threshold of a candidate size as 5; that is, when the number of vectors in a candidate profile is no less than 5, we extract all its vectors and update the existing activity profiles; that is, either create a new activity profile, or update an existing activity profile if the activity already exists.

Experiment and Evaluation

The main goals of the evaluation is to assess how good the proposed approach is at capturing unknown activities and learning and recognising the activity labels over time. In the following, we will introduce the evaluation methodology including the dataset and measurement, and present and discuss the evaluation results.

Experiment Methodology

We choose a real-world smart home dataset for evaluation, even though the technique can be applied to any other applications. The dataset is collected by the University of Amsterdam and on a single-resident house that is instrumented with wireless sensor network. The sensors are state-change binary sensors attached to household objects like cupboards and doors [14]. The recorded activities include making breakfast, cooking dinner, leaving the house, sleeping, and taking shower, etc.

There are two main goals of the evaluation: (G1) effectiveness of discovering unknown activities and (G2) effectiveness of learning and inferring unknown activity labels. G1 is measured in the accuracies of distinguishing unknown and known activities; that is, $A_d = \frac{N_{ua}+N_{ka}}{N}$, where N_{ua} and N_{ka} are the numbers of times that an unknown and known activity are correctly detected, and N is the total number of test instances. G2 is measured in the recognition accuracies; that is, $A_r = \frac{N_a}{N}$, where N_a is the number of times that an activity label is correctly inferred.

To evaluate the above two goals, we design two experiments: (E1) focuses on evaluating the known/unknown detection accuracy, in which we do not perform the online incremental learning strategy. The reason to include this experiment is to see whether the devised unknown activity detection algorithms presented in the **Assessment** steps are effective without the possible help/interference of model updating. That is, the performance is assessed statically on the test data which is not in the training data and will not be used to update the model neither.

The experiment (E2) on the other hand assesses the performance of online learning with respect to both known/unknown detection (G1) and activity recognition (G2); that is, how well the algorithm incrementally learns to discover and recognise unknown activities and how the update affects G1 and G2's performances. Note that in this experiment, as the model is updating, the evaluation of both G1 and G2 therefore is dynamic. For example, once a new activity profile is added in the **Candidate** step, we immediately mark the new activity as known. All the future vectors under consideration from that particular activity will be considered as known all together. However, it is rare that a newly added candidate set, usually consists of only 2-5 vectors, can fully represent the whole activity. Therefore, the detection rate for that activity will be greatly affected at least at the initial stage after the candidate set is included. Activity recognition evaluation will suffer the similar problem. We note this evaluation is harsh but more realistic in the real world setting. That is, one cannot expect the availability of all data let alone the future unseen ones. This assessment therefore shows how the proposed approaches handle the scarcity of information and learn new activities progressively.

The process of each experiment works as follows. We segment all the sensor events in the dataset into one-minute time slots and generate feature vectors for each slot. Given the number of pre-known activities N, we run I iterations (in our experiment, I = 100), in each of which we randomly generate a set of N known activities and shuffle the whole dataset. For the known activities, we split the corresponding data set into P% for training (i.e., create activity profiles) and (1-P)% for testing. The (1-P)% testing data will be combined with the unknown activity data to be fed to proposed algorithms to evaluate G1 and G2. The number of known activities N varies from 1 to $|A| - 1$, where $|A|$ is the total number of activities.

Results

Known/Unknown Detection

Figure 1 presents the results of the experiment E1; that is, the accuracies of correctly detecting known and unknown activities when there is no update performed. The 80% of the chosen known activities' data is used in the creation step to train the pre-known activity profiles. The results have shown that the proposed approach can fairly well separate new types of activities from pre-known activities.

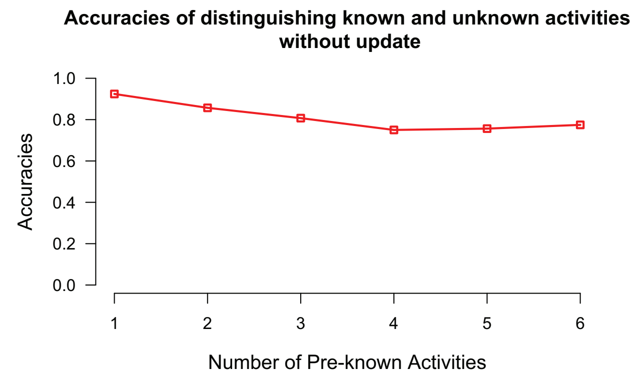

Figure 1: Accuracies of distinguishing known and unknown activities when there is no update performed

Unknown Activity Recognition

Figure 2 and 3 present the results of the experiment E2; that is, the accuracies of detecting unknown activities together with activities recognition when the activity model will be incrementally updated. In this two figures, 20% of the known data is used as training. As shown in Figure 2, the accuracies of distinguishing vectors from known and unknown activities are quite stable even when the pre-known knowledge is scarce. When the pre-known activity size is small, the detection problem is obviously harder: as most of the patterns need to be learnt on-line (which differs from the no-update setting and known/unknown are fixed). The slight upward trend in Figure 2 confirms this observation. On the other hand, the relative small difference between the six settings also demonstrates the updating procedures work well; in other words, the unknown activities are learnt as the learning process goes on.

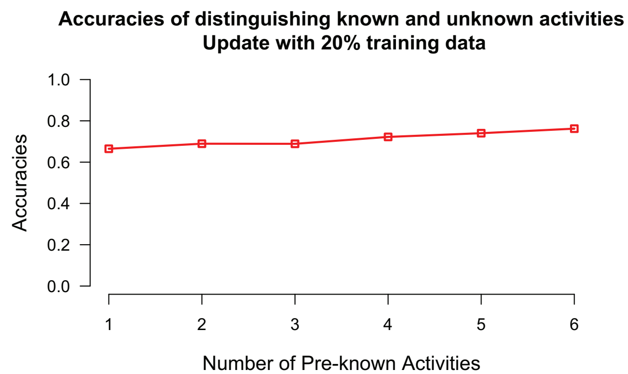

Figure 2: Accuracies of distinguishing known and unknown activities when an online updating is performed

Figure 3 presents the accuracies of recognising activity labels. The figure has a similar trend to Figure 2. Again the general upward trend shows the increasing complexity takes some effect; while the relative small difference across the settings also demonstrates the effectiveness of the devised on-line updating procedures.

Conclusion and Future Work

This paper addresses a new research question in activity recognition: discovering and recognising unknown human activities. This is a critical requirement for large-scale and long-term deployment of an activity-aware pervasive systems in real-world environments, where it is inappropriate to assume that users will only perform a pre-defined closed set of activities all the time or the patterns of users performing such activities will be fixed. Towards addressing

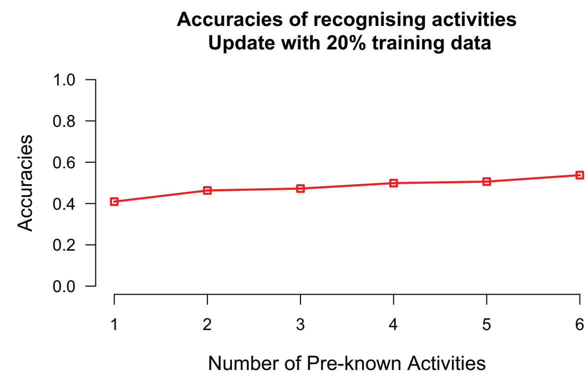

Figure 3: Accuracies of recognising activity labels when an online updating is performed

this question, this paper explores the solution space and proposes a basic approach – estimation-based single centroid approach. We novelly adapt the statistical technique to enable incremental updating without the need of storing any historic data, which is ideal for not only resource-constrained devices but also long-term running of the system. In the proof-of-concept evaluation, the approach works well in detecting new unknown activities, however, there is still a room to improve on learning the activity labels.

In terms of future work, we intend to look for more advanced statistical techniques to improve the recognition accuracies. In our current experiments, we have found some of the sensor features are correlated; for example, the kitchen sensors are usually activated at the same time or within short time intervals. Sensor readings are erroneous by their nature; capturing the possible correlations between sensor readings therefore can further improve confidence of the recognition. In terms of the evaluation, we plan to deploy both algorithms in real-world applications to assess the effectiveness of detecting and learning unknown activities further, and more importantly conduct the user studies to find the opportune moment to query the users for labelling the unknown activities.

Appendix A

The derivation of online updating on a singe centroid profile.

Proof. The update formula for $\boldsymbol{c}_{n+1}$ is trivially proved by the definition of $\boldsymbol{c}$. Note $\boldsymbol{c}_{n+1}$ is a unit vector. For simplicity, we omit the subscription $n+1$ from the parameters.

$$\begin{aligned}\bar{d} &\triangleq \frac{1}{n+1}\sum_{i=1}^{n+1} d_i = \frac{1}{n+1}\sum_{i=1}^{n+1}(1-\boldsymbol{c}'\cdot\boldsymbol{v}_i)\\ &= 1-\frac{1}{n+1}\boldsymbol{c}'\cdot\boldsymbol{u} = 1-\frac{1}{n+1}\parallel \boldsymbol{u}\parallel\end{aligned}$$

The last equality holds because $\boldsymbol{c}$ is a unit vector that parallels with $\boldsymbol{u}$. Therefore, $\boldsymbol{c}'\cdot\boldsymbol{u} = 1\cdot\parallel \boldsymbol{u}\parallel\cos 0 = \parallel \boldsymbol{u}\parallel$, which proves the equation for $\bar{d}$. Define the sum of squares SS as

$$\begin{aligned}\text{SS} &\triangleq \sum_{i=1}^{n+1}(d_i-\bar{d})^2 = \sum_{i=1}^{n+1}(1-d_i-(1-\bar{d}))^2\\ &= \sum_{i=1}^{n+1}(\boldsymbol{c}'\boldsymbol{v}_i)^2 - 2(1-\bar{d})\sum_{i=1}^{n+1}\boldsymbol{c}'\cdot\boldsymbol{v}_i + \sum_{i=1}^{n+1}(1-\bar{d})^2\\ &= \boldsymbol{c}'\boldsymbol{M}\boldsymbol{c} - 2\frac{1}{n+1}\boldsymbol{u}'\cdot\boldsymbol{u} + \frac{1}{n+1}\parallel \boldsymbol{u}\parallel^2\\ &= \boldsymbol{c}'\boldsymbol{M}\boldsymbol{c} - \frac{1}{n+1}\parallel \boldsymbol{u}\parallel^2 = S_{n+1}\end{aligned}$$

which completes the proof of the standard deviation. □

REFERENCES

1. J.K. Aggarwal and M.S. Ryoo. 2011. Human Activity Analysis: A Review. *ACM Comput. Surv.* 43, 3, Article 16 (April 2011), 43 pages. DOI: http://dx.doi.org/10.1145/1922649.1922653
2. Luca Canzian and Mirco Musolesi. 2015. Trajectories of Depression: Unobtrusive Monitoring of Depressive States by means of Smartphone Mobility Traces Analysis. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (ACM UbiComp'15)*. ACM.
3. Liming Chen, J. Hoey, C.D. Nugent, D.J. Cook, and Zhiwen Yu. 2012. Sensor-Based Activity Recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C* 42, 6 (2012), 790–808. DOI: http://dx.doi.org/10.1109/TSMCC.2012.2198883
4. Sunny Consolvo, James A. Landay, and David W. McDonald. 2009. Designing for Behavior Change in Everyday Life. *Computer* 42, 6 (June 2009), 86–89. DOI:http://dx.doi.org/10.1109/MC.2009.185
5. P. Dawadi, D. Cook, and M. Schmitter-Edgecombe. 2015a. Automated Cognitive Health Assessment from Smart Home-Based Behavior Data. *IEEE Journal of Biomedical and Health Informatics* PP, 99 (2015), 1–1. DOI:http://dx.doi.org/10.1109/JBHI.2015.2445754
6. Prafulla N. Dawadi, Diane J. Cook, and Maureen Schmitter-Edgecombe. 2015b. Modeling patterns of activities using activity curves. *Pervasive and Mobile Computing* (2015), –. DOI: http://dx.doi.org/10.1016/j.pmcj.2015.09.007

7. Sarah Gallacher, Eliza Papadopoulou, Nick K. Taylor, and M. Howard Williams. 2013. Learning User Preferences for Adaptive Pervasive Environments: An Incremental and Temporal Approach. *ACM Trans. Auton. Adapt. Syst.* 8, 1, Article 5 (April 2013), 26 pages. DOI: http://dx.doi.org/10.1145/2451248.2451253

8. Jose Antonio Iglesias, Plamen Angelov, Agapito Ledezma, and Araceli Sanchis. 2012. Creating Evolving User Behavior Profiles Automatically. *IEEE Transactions on Knowledge and Data Engineering* 24, 5 (2012), 854–867. DOI: http://dx.doi.org/10.1109/TKDE.2011.17

9. Wilhelm Kleiminger, Friedemann Mattern, and Silvia Santini. 2014. Predicting household occupancy for smart heating control: A comparative performance analysis of state-of-the-art approaches. *Energy and Buildings* 85 (2014), 493 – 505. DOI: http://dx.doi.org/10.1016/j.enbuild.2014.09.046

10. O. D. Lara and M. A. Labrador. 2013. A Survey on Human Activity Recognition using Wearable Sensors. *IEEE Communications Surveys Tutorials* 15, 3 (Third 2013), 1192–1209. DOI: http://dx.doi.org/10.1109/SURV.2012.110112.00192

11. Seng W Loke. 2004. Representing and reasoning with situations for context-aware pervasive computing: a logic programming perspective. *The Knowledge Engineering Review* 19, 03 (2004), 213–233.

12. Kanti V. Mardia and Peter E. Jupp. 2008. *Directional Statistics*. Wiley.

13. EmmanuelMunguia Tapia, Tanzeem Choudhury, and Matthai Philipose. 2006. Building Reliable Activity Models Using Hierarchical Shrinkage and Mined Ontology. In *Pervasive '06*. Springer Berlin Heidelberg, 17–32.

14. Tim van Kasteren, Athanasios Noulas, Gwenn Englebienne, and Ben Kröse. 2008. Accurate Activity Recognition in a Home Setting. In *UbiComp '08: Proceedings of the 10th International Conference on Ubiquitous Computing*. ACM, Seoul, Korea, 1–9.

15. Juan Ye, Stamatia Dasiopoulou, Graeme Stevenson, Georgios Meditskos, Efstratios Kontopoulos, Ioannis Kompatsiaris, and Simon Dobson. 2015. Semantic web technologies in pervasive computing: A survey and research roadmap. *Pervasive and Mobile Computing* 23 (2015), 1 – 25. DOI: http://dx.doi.org/10.1016/j.pmcj.2014.12.009

16. Juan Ye, Simon Dobson, and Susan McKeever. 2012. Situation Identification Techniques in Pervasive Computing: a review. *Pervasive and mobile computing* 8 (Feb. 2012), 36–66. Issue 1.

17. Juan Ye, Graeme Stevenson, and Simon Dobson. 2014a. KCAR: A knowledge-driven approach for concurrent activity recognition. *Pervasive and Mobile Computing* 0 (2014). DOI: http://dx.doi.org/10.1016/j.pmcj.2014.02.003

18. Juan Ye, Graeme Stevenson, and Simon Dobson. 2014b. USMART: An Unsupervised Semantic Mining Activity Recognition Technique. *ACM Trans. Interact. Intell. Syst.* 4, 4, Article 16 (Nov. 2014), 27 pages. DOI: http://dx.doi.org/10.1145/2662870

19. Y. Zhang, N. Meratnia, and P. Havinga. 2010. Outlier Detection Techniques for Wireless Sensor Networks: A Survey. *IEEE Communications Surveys Tutorials* 12, 2 (Second 2010), 159–170. DOI: http://dx.doi.org/10.1109/SURV.2010.021510.00088