Modeling and Reasoning of Contexts in Smart Spaces

Jae Woong Lee
Department of Computer Science
State University of New York
Oswego, New York
jaewoong.lee@oswego.edu

Sumi Helal
School of Computingg & Communications
Lancaster University
Lancaster, UK
s.helal@lancaster.ac.uk

***Abstract*— Context can be defined to be a meaningful and descriptive state of a physical space, in which relevant activities of the humans and movements of objects in the same space are consecutively performed. Context aware computing thus requires the instantiation and use of a suitable context model and algorithms capable of discerning a most relevant context at the present time. In our previous research, we developed a method based on stochastic analysis, for inferring contexts from sensor datasets collected from a given smart space. By utilizing three statistical techniques including conditional probability table (CPT), K-means clustering, and Principal Component Analysis (PCA), our inference method is able to predict and generate contexts which can potentially happen. Generated contexts are then used as references when the present context needs to be found. In this paper, we build on our prior work and propose a method that analyzes the current state space and determines the present context guided by the collection of generated potential contexts. We first reconsolidate our context models and context graph in which contexts are structured, and then introduce reasoning methods which apply Euclidean distance and Cosine Similarity. PCA is also used to statistically analyze the state space, which helps to achieve better performance. Using experimental evaluation, we validated the accuracy of our proposed approach in identifying the present context.**

Keywords—context awareness computing, smart spaces/cities, sensors, IoT, context graph, principal components analysis, context modeling and reasoning

I. Introduction

A smart space is defined as a space digitalized by sensors and actuators into an intelligent system which can provide convenient services to people located in the space. The services include automation of daily chores, accessibility services for people who need assistance and support in spaces like home or work place. Ideally, the smart space and its intelligent system should discern which state the residents are in and decide which services they are expecting or are needed. Since appropriate services can be provided only after reasoning and determining the current state, the initial step in the system is to accurately understand what context(s) the space and its users are in. Thus, it becomes ever more important to follow a context-aware design of the smart space intelligent system. In order to recognize contexts, contexts need to be first modeled and organized. Dey [1] defined context as “any information that can be used to characterize the situation of an entity; an entity is a person, place, or object that is considered relevant to the interaction between a user and applications themselves.” In the definition of Yau [2], it is “any detectable attribute of a device, its interaction with external devices, and/or its surrounding environment.” According to both definitions, context can be described by particular entities which are involved and their interactions. Because the entities enable to be aware of current contextual status, they focus more on how to capture a context. In our prior work, we defined a context to be an abstract state of a given space which can be described by the status of entities located in the space [3].

The status of entities in a smart space can be assumed to be easily obtainable through sensors attached to these entities. This is a reasonable assumption as many entities today follow the Internet-of-Things design principles in which an entity provides APIs to access it by other devices and to obtain status data. A context is recognized by analyzing collected status data, and then maps to operate actuators and/or smart devices, which provide necessary and/or convenient services. This smart space research is relevant and extendable to the Smart city which also hugely benefits from being aware of contexts for individual citizens, public services, vital infrastructure, natural resources, and much more.

In this paper we address modeling and reasoning of contexts. The recognition process demands accurate analysis of sensor data since the context is realized by the sensors attached to the entities in the space. It requires building well-structured profiles for contexts, which therefore can be easily managed and efficiently manipulated. We developed an approach to model contexts by utilizing stochastic analysis of sensor data [4]. The key idea comes from two observations: the first being sensors which are triggered in a same given context are mostly related. The second observation is that some sensors are particularly important and contributive to specific contexts. Hence, a whole dataset can be logically divided into groups each of which contains specific context-related sensor data. The performance of smart spaces and how intelligent they are and how context-rich they may be obviously depend on various factors including the sensors and IoT technologies used. This interplay is discussed in this paper.

The overall framework of our proposed approach consists of two phases: the first phase is to build context models by utilizing the methods in [4], and the second phase is to determine a present context which is the most representative of the current state space, and doing so by considering all contexts models. In the first phase, the context models are derived from a given sensor dataset which was generated under supervised learning. In this learning phase, three machine learning and statistical methods are utilized. First, two methods of Conditional Probability Table (CPT) and K-means clustering enables us to find k numbers of groups, each of which is declared as a context. Second, Principal Components Analysis (PCA) is used to find important and representative sensors per each context, which finally defines the complete context model. Once the context models are built, one context will be discovered by comparing the current state space and all the context models. At this reasoning step, two methods of calculating Euclidean distances and cosine similarity are utilized. This paper first introduces modeling of contexts from the collection of state spaces and then addresses reasoning methods.

On the other hands, during the process in modeling contexts, it is observed that particular pairs of contexts are related with causality or in timely order. For instance, a context "sleeping at night" is followed by another context "having breakfast." However, the opposite pair that is "having breakfast" followed by "sleeping at night" never happens. It infers that a collection of contexts can be formed and located by the relations. Structure of contexts reflected by the properties is called a context graph, which eventually improves reasoning methods. From the previous example, when it is needed to reason a context in a given context "having breakfast," a context "sleeping at night" is not going to be considered and skipped during reasoning contexts, since it should not occur next. Thus, it reduces the number of contexts to comparing in the reasoning phase and derives benefits in computational complexity.

This paper is organized as follows. In the section II, we describe existing work related to context awareness computing and reasoning methods. Section III introduces context model and context graph which are fundamental factors of our approach. The principles of the proposed approach are overviewed in the section IV, which is followed by experimental validation. We conclude the paper with a short discussion and future work plan.

II. RELATED WORK

There were many context models proposed in human computer interaction [5], context awareness computing [6], and activity recognition learning [7]. The proposals attempted to model contexts from the current state of the space which was observed through human senses or electronic sensors. In some research, the context models were defined in the form of rules. In Context-Aware Simulation System for smart home (CASS) [8], for instance, the system defined rules to describe certain conditions, detected the conflicts of rules, and provided the ability to control a character to move it. In Context-driven simulation approach [3], contexts were defined as abstract and representative state spaces by specifying related sensors and their status. Additionally, the context models defined the causality in between other contexts, which enabled to generate the entire daily living scenario.

The research in reasoning contexts commonly have oriented methods in comparing predefined contexts and the current situation. Ontology facilitated efficient modeling and reasoning for context [9], however it still needed human users' efforts in configuration which remains burdens and ambiguousness in defining contexts. In order to avoid the burdens and increase automaticity, research in activity recognition on deriving meaningful high-level information from low-level information could be used. The goal of the research is to cluster from collected sensor datasets which are triggered by activities. For instance, [10] proposed Cluster Based Classification for Activity Recognition Systems (CBARS), in which a supervised learning model was built first, and unsupervised learning for new data was applied for new activity recognition. The challenge was that it needed a supervised learning model. Activity recognition using Active Learning in the Overlapped activities (AALO) [11] addressed that challenge. It is an active recognition system that can accurately classify specified activities according to locations and times in which the activities are performed. Cluster-Based Classifier Ensemble (CBCE) [12] developed a method for combing multiple classifiers including Naïve Bayes (NB) models, hidden Markov models (HMMs), and conditional random fields (CRFs). This ensemble of classifiers can recognize activities in given sensor datasets, however, they do not provide a method to define abstract information of context to represent the other state spaces in a cluster.

As the context-awareness computing is getting integrated into end-user service applications, the more practical and applicable methods have been being researched. Mobile cloud services [13] and recommender systems [14][15] applied historic computational methods such as cosine similarity for recognizing contexts. They aim to find the present context and a service which is proper and suggested within the present context. The methods remain an issue of high computational complexity in finding similar contexts and services.

III. OVERVIEW OF CONTEXT MODELING

A context is sensed by changes of status of objects and/or residents located in a certain space. To introduce the mechanism of recognizing contexts, we necessarily define the terms – state space, context, and context graph and then explain the principles of the reasoning algorithm in the following section.

A. State Space and Context

In a smart space, the status of the space is recognized by sensors attached on objects and human residents and shortly named state space. A state space is formally defined by the collection of the sensors' status and denoted by $\boldsymbol{S}$. Thus, a state space with $\boldsymbol{\omega}$ sensors is expressed in $\boldsymbol{S} = \{\boldsymbol{s_1}, \boldsymbol{s_2}, \dots, \boldsymbol{s_i}, \dots, \boldsymbol{s_\omega}\}$ where $\boldsymbol{s_i}$ contains a status (usually value) of sensor $\boldsymbol{s_i}$. During the daily living scenario, the state space persistingly changes since activities of human residents or movements of objects affect the status of sensors. Fig. 1 shows how the state space $\boldsymbol{S}$ changes. The universal set of state spaces is a collection of the all $\boldsymbol{S}$ and noted by $\widehat{\boldsymbol{S}} = \{\boldsymbol{S}_1, \boldsymbol{S}_2, \dots, \boldsymbol{S}_c, \dots, \boldsymbol{S}_\zeta\}$ where $\boldsymbol{\zeta}$ means the number of state spaces obtained in a given daily living period and $\boldsymbol{S}_c$ stands for the current state space. Note that $\widehat{\boldsymbol{S}}$ is recursive. Fig. 1 illustrates changes of the $\boldsymbol{S}$ in a given time period starting $\boldsymbol{S}_1$. As time goes further, the $\boldsymbol{S}$ will change more, and may return back to $\boldsymbol{S}_1$ or any state space in $\widehat{\boldsymbol{S}}$.

Context is a unique state space that is meaningful and representative enough to describe what is currently happening in the space [3]. Consider that a kitchen in which various sensors are attached on microwave, windows, ceiling light, electric stove, knife, and refrigerator, and a context named "making breakfast" is recognized. Because microwave, electric stove, knife and refrigerator are tools used in making breakfast, state spaces triggered by sensors attached on these objects are big contributors for the context. However, state spaces involved by sensors on windows and ceiling light are less meaningful and thus ignored in reasoning the context. In short, the state spaces which mostly contribute in becoming contexts are selected as a context. For instance, Fig. 1 shows are only a few brown colored state spaces are declared as contexts. Once a context is chosen, others state spaces after it belong to the context, since the context lasts until a new context begins.

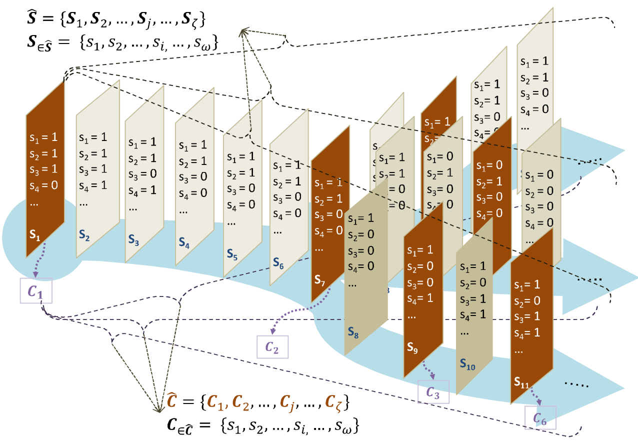

Fig. 1. A set of state spaces $\widehat{S}$ and contexts $\widehat{C}$ in a given daily living scenario.

Because a context is a selected state space, it basically has the identical structure with the collection of sensors as follows: $C = \{s_1, s_2, \ldots, s_i, \ldots, s_\omega\}$. As a multitude of contexts can be selected, the collection of the contexts is noted by $\widehat{C} = \{C_1, C_2, \ldots, C_\rho, \ldots, C_\eta\}$ where η means the number of contexts and C_ρ stands for the present context. Note that $\eta \leq \zeta$, since $\widehat{C}$ is a subset of $\widehat{S}$ $(\widehat{C} \subseteq \widehat{S})$. The status of these sensors in a given context is considered as condition for being the context, since it distinguishes itself from non-context state spaces or other contexts. In other words, when the sensors in a given state space should have the same status for the corresponding sensors in a target context, it is said that the state space represents a context.

B. *Context Graph*

Contexts in $\widehat{C}$ are driven and transitioned in a certain order. It implies causal relations between contexts: for instance, C_1 causes C_2, and C_2 causes C_3. Thus, $\widehat{C}$ can be considered as a collection of contexts with causality and is called Context Graph C_G as illustrated in Fig. 2. Some of the relations between contexts are determinative, for instance, the relation from C_1 to C_2 shows that C_1 is always followed by C_2. On the other hands, relations caused by C_2 are non-determinative, since there are multiple contexts following next, for instance, either C_3 or C_5 is a subsequence of C_2. Contexts following a given context C_j are defined as $C_N = \{C_1, C_2, \ldots, C_\tau, \ldots, C_\chi\}$ and $C_N \subseteq \widehat{C}$. Note that every C_j has its own C_N.

In the real daily living scenarios, there could be a sequence of contexts which is repeated as a daily routine. For example, "making breakfast" context occurs every morning, "sleeping" context should start every night for most people. Thus, contexts are cycled, and it is hard to declare an originally and naturally rooted context. Note that some contexts such as C_1 in Fig. 2, graphically look like a root, however it might be a next context of a certain context in the real scenarios.

IV. Principles of Context Reasoning

A. *Overall Approach*

Reasoning a context is the same as discovering a next context in a given context. Thus, the approach aims to find a next context with the given state space which is considered as the current state space S_c. Since S_c occurs in the present context C_ρ, it can become one of the next possible contexts C_N, which can be found from C_G. It implies that the reasoning a context is the same as finding a next context in a given context and transitioning into the found context.

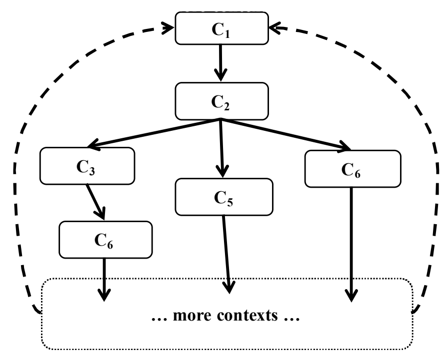

Fig. 2. Context Graph C_G is created from $\widehat{S}$. Note that the dashed edges are transactions which are not illustrated in Fig. 1, however they might occur as the state space changes. For simplicity, we assume that it returns to C_1.

Reasoning process is fundamentally event-driven, since state space is changed by various sensor events such as activities of resident or movement of objects in the space. If reasoning procedure qualifies a context as a next context C_v, transition into the next context is indeed performed as shown in Alg. I. On the other hands, it stays in the current context and waits for next changes of the state space if it fails to find C_v. Alg. I. runs in two steps: *initialization* and *reasoning*.

Alg. I. *Context Reasoning Main*

INPUT: a training set of state spaces $\widehat{S}$
OUTPUT: a context

1. Context Graph $C_G \leftarrow$ *Build a context graph* with $(\widehat{S})$
2. State Space $S_c \leftarrow$Initialize the current state space
3. Context $C_\rho \leftarrow$ *Initialize the present context* with (S_c, C_G)
4. Context $C_v \leftarrow$ **NULL** // selected next context
5. **while** S_c changes by any events
6. $\quad C_v \leftarrow$ *Reason context* with (S_c, C_ρ, C_G)
7. $\quad$**if** $C_v != $ **NULL**
8. $\quad\quad C_\rho \leftarrow C_v$ // transition to C_v
9. $\quad\quad$// output C_ρ as the new present context
10. $\quad$**end if**
11. **end while**

1) Initialization Steps (*line 1 to 4*): before reasoning contexts, it is needed to establish a context graph C_G with a given set of state spaces $\widehat{S}$ which is formed from training datasets. C_G is built by applying stochastic analysis method of sensor datasets [4]. The stochastic analysis method consists of three main phases. First, the dataset is analyzed by *conditional probabilistic table* (CPT) based on Bayesian networks and estimates the number of contexts, that is K. Then the dataset is clustered into K contexts by *K-means clustering*. Thirdly, conditions for becoming each context are discovered by utilizing *Principal Components Analysis* (PCA). Use of PCA enables to find the most important sensors for a given context and optimizes both the context modeling and recognizing. It will be explained in detail with the reasoning step.

Once $\boldsymbol{C_G}$ is established, current state space $\boldsymbol{S_c}$ and present context $\boldsymbol{C_\rho}$ are initialized. $\boldsymbol{S_c}$ is simply configured by the current status of sensors. Initialization of $\boldsymbol{C_\rho}$is performed by the Alg. II. The purpose of the algorithm is to find an initial present context $\boldsymbol{C_\upsilon}$ which has the highest similarity with $\boldsymbol{S_c}$.

Alg. II. *Initializing Context*

INPUT: current state space $\boldsymbol{S_c}$; context graph $\boldsymbol{C_G}$
OUTPUT: the highest similarity scored contexts $\boldsymbol{C_\upsilon}$

Next context $\boldsymbol{C_\upsilon} \leftarrow \mathbf{NULL}$
for each context $\boldsymbol{C_\tau}$ in $\boldsymbol{C_G}$
  if similarity $(\boldsymbol{S_c}, \boldsymbol{C_\tau}) >$ similarity $(\boldsymbol{S_c}, \boldsymbol{C_\upsilon})$
    then $\boldsymbol{C_\upsilon} \leftarrow \boldsymbol{C_\tau}$ // replace C_τ as a next context
  end if
end for

2) Reasoning Steps (*line 5 to 11*): when $\boldsymbol{S_c}$ changes due to sensor event, it is attempted to discover a context which is the most similar to $\boldsymbol{S_c}$ as shown in Alg III. If a proper context model is found, it becomes the new present context $\boldsymbol{C_\rho}$. Like in Alg. II, similarity between $\boldsymbol{S_c}$ and each target context $\boldsymbol{C_\tau}$ is calculated. The proposed algorithm can reduce the number of target contexts in calculating the score due to $\boldsymbol{C_G}$. By the definition of $\boldsymbol{C_G}$, since $\boldsymbol{C_\rho}$ is followed by $\boldsymbol{C_N}$ of $\boldsymbol{C_\rho}$, only contexts in $\boldsymbol{C_N}$ are considered for reasonging process. If reasoning process is unable to find any context model and therefore $\boldsymbol{C_\upsilon}$ is empty (line 7 in Alg. I.), it skips transitioning to next context and the current context $\boldsymbol{C_\rho}$ continues.

Alg. III. *Reasoning Context*

INPUT: current state space $\boldsymbol{S_c}$; present context $\boldsymbol{C_\rho}$; context graph $\boldsymbol{C_G}$
OUTPUT: the highest similarity scored contexts $\boldsymbol{C_\upsilon}$

Next Contexts $\boldsymbol{C_N} \leftarrow$ find all next contexts of $\boldsymbol{C_\rho}$ in $\boldsymbol{C_G}$
Set of Candidates Contexts $\widehat{\boldsymbol{C}}_{\mathbf{N}} \leftarrow \mathbf{NULL}$
for each context $\boldsymbol{C_\tau}$ in $\boldsymbol{C_N}$
  if similarity condition between $\boldsymbol{S_c}$ and $\boldsymbol{C_\tau}$ is matched
    then $\widehat{\boldsymbol{C}}_{\mathbf{N}} \leftarrow \widehat{\boldsymbol{C}}_{\mathbf{N}} \cup \boldsymbol{C_\tau}$ // add C_τ to $\hat{C}_{\mathrm{N}}$ as a candidate
  end if
end for
$\boldsymbol{C_\upsilon} \leftarrow$ the highest similar scored $\boldsymbol{C_\tau}$ in $\widehat{\boldsymbol{C}}_{\mathbf{N}}$.

Dissimilar to Alg. II which selects any context and declare it as $\boldsymbol{C_\rho}$, it needs more consideration to discover possible next contexts $\boldsymbol{C_N}$. If a context is too different from $\boldsymbol{S_c}$, it must not have a chance to become $\boldsymbol{C_\rho}$. In some cases, contexts in $\boldsymbol{C_N}$ have too low similarity and should not be any context. In order to address the cases, therefore a context requires to be filtered by a threshold $\boldsymbol{\theta}$ which stands for a particular similarity grade. We will introduce the reasoning methods and define $\boldsymbol{\theta}$ in the next section.

B. *Context Reasoning Methods*

$\boldsymbol{sim}(\boldsymbol{S_c}, \boldsymbol{C_\tau})$ measures the similarity between $\boldsymbol{S_c}$ and each possible next context $\boldsymbol{C_\tau}$ ($\in \boldsymbol{C_N}$). Since state space and context conditions are vectorized with status of sensors, two computational methods are utilized.

1) Euclidean distance: when the distance is within a fuzzy threshold $\boldsymbol{\theta_d^{C_\tau}}$ which specfies a range for the condition of $\boldsymbol{C_\tau}$, it signifies no difference between $\boldsymbol{S_c}$ and $\boldsymbol{C_\tau}$, hence $\boldsymbol{C_\tau}$ becomes the next context. Note that $\boldsymbol{\theta_d^{C_\tau}}$ is context-specific and thus each context has its own value. It is obtained by averaging distances from each state space to context during the initialization step [4]. Some conditions may be defined with a range of values, instead of one value. In this case, in order to apply Euclidean distance, the range is represented by a median of the range. The Euclidean distance is calculated by the following equation:

$$\boldsymbol{dist}(\boldsymbol{S_c}, \boldsymbol{C_\tau}) = \|\boldsymbol{S_c} - \boldsymbol{C_\tau}\| = \sqrt{\sum_{i=1}^{\omega} \left(s_i^{S_c} - \mathbf{s}_i^{C_\tau}\right)^2}$$

where $\boldsymbol{s_i^{S_c}}$ means sensor s_i in the current state space $\boldsymbol{S_c}$ and $\mathbf{s}_i^{C_\tau}$ means sensor $\boldsymbol{s_i}$ in the next possible context $\boldsymbol{C_\tau}$.

If the distance is below than $\boldsymbol{\theta_d^{C_\tau}}$, the $\boldsymbol{C_\tau}$ is considered to possibly be the next context. On the other hands, when it is greater than $\boldsymbol{\theta_d^{C_\tau}}$, there is no chance for $\boldsymbol{S_c}$ to become $\boldsymbol{C_\tau}$. The similarity is calculated and compared for all of $\boldsymbol{C_\tau}$ which satisfy the distance condition, that is $\boldsymbol{dist}(\boldsymbol{S_c}, \boldsymbol{C_\tau}) \leq \boldsymbol{\theta_d^{C_\tau}}$.

$$\boldsymbol{sim_{dist}}(\boldsymbol{S_c}, \boldsymbol{C_\tau}) = 1 - \frac{\boldsymbol{dist}(\boldsymbol{S_c}, \boldsymbol{C_\tau})}{\|\boldsymbol{S_c}\|} = 1 - \frac{\boldsymbol{dist}(\boldsymbol{S_c}, \boldsymbol{C_\tau})}{\sqrt{\sum_{i=1}^{\omega}\left(s_i^{S_c}\right)^2}}$$

If none of $\boldsymbol{C_\tau}$ does verify the condition, the current state space cannot not be any of context, therefore no change of context occurs.

2) Cosine similarity: dissimilar to the Euclidean distance which values the distance between the end points of two vectors, cosine similarity verifies appropriatesness of angle between two vectors. The $\boldsymbol{sim_{cos}}(\boldsymbol{S_c}, \boldsymbol{C_\tau})$ is calculated.

$$\boldsymbol{sim_{cos}}(\boldsymbol{S_c}, \boldsymbol{C_\tau}) = \frac{\boldsymbol{S_c} \cdot \boldsymbol{C_\tau}}{\|\boldsymbol{S_c}\| \cdot \|\boldsymbol{C_\tau}\|} = \frac{\sum_{i=1}^{\omega} \boldsymbol{s_i^{S_c}} \cdot \mathbf{s}_i^{C_\tau}}{\sqrt{\sum_{i=1}^{\omega}\left(s_i^{S_c}\right)^2} \cdot \sqrt{\sum_{i=1}^{\omega}\left(\mathbf{s}_i^{C_\tau}\right)^2}}$$

As the cosine of two vectors is increasing to 1, they are considered more similar. All of $\boldsymbol{C_\tau}$ those $\boldsymbol{sim_{cos}}(\boldsymbol{S_c}, \boldsymbol{C_\tau})$ is within threshold $\boldsymbol{\theta_a^{C_\tau}}$ for nudging the condition of $\boldsymbol{C_\tau}$, is considered as a possible next context. Failure to find such $\boldsymbol{C_\tau}$ delays to transit to any of next context.

$\boldsymbol{sim_{dist}}(\boldsymbol{S_c}, \boldsymbol{C_\tau})$ fits for finding similarity of state spaces with Boolean stated sensors such as sensors generating two status of 0/1 or On/Off. In other words, when the change of sensor status is monotonous and thus magnitude of vectors $\boldsymbol{S_c}$ and $\boldsymbol{C_\tau}$ is stable, the Euclidean distance can be applied. However, it is not applicable when the status of the sensors is non-Boolean and generates values within certain ranges, since the magnitude of vectors can significantly change. For the state spaces containing those sensors, $\boldsymbol{sim_{cos}(S_c, C_\tau)}$ is used.

3) Projection by PCA: in the reasoning methods above, each vectorized element is individually calculated. As the given space is extended and the state space is scaled with more sensors, the computaional complextiy extremely grows up and affects the performance in reasoning contexts. Since the number of sensors considerably matters, it is critial to reduce the size of the sensors. PCA is the way to find important components and reduce the dimention of a given dataset by projecting into lower dimention.

Principal components are sensors that show definite variance patterns that explicitly express the change of states. We want to know in which pattern the dataset is scattered. For this, a matrix of covariances $\boldsymbol{cov}$ is calculated first. In a $\boldsymbol{\omega}$-dimensional dataset, $\boldsymbol{cov}$ is calculated by:

$$\boldsymbol{cov}\ (\hat{s}^{i}, \hat{s}^{j}) = \frac{\sum_{k=1}^{\omega}\left(\hat{s}_k^i - \mu_i\right)\left(\hat{s}_k^j - \mu_j\right)}{(\boldsymbol{\omega} - 1)}$$

where $\hat{s}^i$ and $\hat{s}^j$ are the set of sensor status in dimensions i and $\boldsymbol{j}$, repsectively; $\boldsymbol{i}$ and $\boldsymbol{j}$ are the sensor values in each dimension. The total covariances establish a $\boldsymbol{\omega} \times \boldsymbol{\omega}$ covaraince matrix $\boldsymbol{R}$:

$$\boldsymbol{R}(\hat{S}) = \begin{bmatrix} \boldsymbol{cov}\ (\hat{s}^1, \hat{s}^1) & \dots & \boldsymbol{cov}\ (\hat{s}^1, \hat{s}^\xi) \\ \vdots & \ddots & \vdots \\ \boldsymbol{cov}\ (\hat{s}^\xi, \hat{s}^1) & \dots & \boldsymbol{cov}\ (\hat{s}^\xi, \hat{s}^\xi) \end{bmatrix}$$

From covariance matrix, we calculate the eigenvectors, each of which can conduct linear transformations of sensor data and characterize its variance; the eigenvalues then measure how well the sensor data is scattered. We choose the eigenvectors that show the most variant spread of data as principal components. If data is evenly scattered with an axis transformed by an eigenvector (i.e., the data pattern is reconized explicitly), it is an important eigenvector, which means it's the desired principal component and has a high eigenvalue. The challenge is in determining the threshold for which eigenvalues are high enough to be acceptable. We propose threshold $\boldsymbol{\theta_e}$ for total eigenvalues of selected eigenvectors. In our approach, first the eigenvectors are sorted by eigenvalues in descending order; then, eigenvectors with higher values are chosen until the sum of corresponding eigenvalues exceeds $\boldsymbol{\theta_e}$.

Eigenvectors satisfying the condition establish a feature matrix. The original high-dimensional dataset is transformed into a low-dimensional dataset through the feature matrix. Finally, a transformed context can be defined in $\boldsymbol{C^T} = \{\boldsymbol{s_1}, \boldsymbol{s_2}, \dots, \boldsymbol{s_\psi}\}$ where $\boldsymbol{\psi} \le \boldsymbol{\omega}$.

V. EXPERIMENAL VALIDATION

The experimental validation evaluates whether contexts are properly reasoned by the proposed approach without violation. Instead of reasoning a single context, we applied a day-length state spaces and performed reasoning of a set of contexts. Since the contexts are sequentially located within the set, they are considered as a contextual scenario or a contextual schedule notated in $\boldsymbol{C_S} (\subseteq \boldsymbol{C_G})$. Thus, the aim of the validation is to analyze the $\boldsymbol{C_S}$ and decide whether it is reasonable and feasible. In our validation process, *Fidelity* represents the degree of the reasonability and *Feasibility* shows whether $\boldsymbol{C_S}$ is feasible. The approach will be validated by showing how many $\boldsymbol{C_S}$ are feasible. For this goal, we conducted following steps:

1) Preparation: we inferred all possible contexts from a set of training sensor datasets obtained from the real scenarios and formed $\boldsymbol{C_G}$. We also collected other sensor datasets as test datasets, which would be used for testing. Note that both training sensor datasets and testing sensor datasets were formed as state spaces of $\hat{\boldsymbol{S}}$.

2) Test: we reasoned the test datasets and obtained $\boldsymbol{C_S}$, which would be used for measuing fidelity.

3) Evaluation: if the reasoning approach is flawless, $\boldsymbol{C_S}$ which is obtained from test is found along with $\boldsymbol{C_G}$. If it fails to find $\boldsymbol{C_S}$ satisfying particular conditions, it means that the scheduling contexts is not available and the reasoning process is stuck in the middle of $\boldsymbol{C_G}$. It describes a situation in which no other contexts are found by the reasoning approach. Conditions for evaluation algorithms will be explained next.

For the training datasets, we collected sensor datasets for 11 days in Gator Tech Smart House (GTSH) [17]. The datasets captured status of 19 sensors which were deployed on 18 objects while 8 activities were performed. Context models and $\boldsymbol{C_G}$ are created by inferring the datasets proposed by [4].

The testing datasets are synthesized by Persim 3D simulator [16]. It adapted a context-driven simulation approach and enabled to automatically synthesize different datasets. For the experiments, we collected 11 test datasets, which generated 11 $\boldsymbol{C_S}$ by using the reasoning approach. We evaluate the 11 $\boldsymbol{C_S}$ by proving that they are discovered in $\boldsymbol{C_S}$ and meet following conditions:

- *Condition 1*. $|\boldsymbol{C_S}| \ge 2$, where $|\boldsymbol{C_S}|$ stands for the number of elements (that is contexts) in the set $\boldsymbol{C_S}$.

It represents there is at least one context transition and the reasoning approach finds another context excluding the start context. In general, if a daily living scenario ends in the starting context without any transitions, it is considered that the reasoning approach fails. In our experiment, no test dataset passed the condition.

- *Condition* 2: $\boldsymbol{P(C_S)} \ge \boldsymbol{OCT}$, where $\boldsymbol{P(C_S)}$ stands for the probability for occurrence of $\boldsymbol{C_S}$ and $\boldsymbol{OCT}$ is threshold of the occurrence probability.

Note that $\boldsymbol{C_G}$ is a directed graph which defines various paths, which are considered as all contextual schedule $\boldsymbol{C_S}$. If $(\boldsymbol{C_S})$ is too low, it is considered that $\boldsymbol{C_S}$ rarely occurs, and thus it should be reasoned seldom. This condition is the primary evaluator which measures the fidelity and the feasibility. To achieve this purpose, we defined statistical models. First, we adapted Bayesian network, which enables to calculate join probability distribution on the entire dataset by using CPT. We formed the CPT of all contexts in $\boldsymbol{C_G}$ and calculated the probability of $\boldsymbol{C_S}$.

$$\text{BN Probability } (\boldsymbol{BNP}) = \prod_{C_i, C_j \in C_S} \boldsymbol{P(C_i, C_j)}$$

where $\boldsymbol{P(C_i, C_j)}$ stands for the conditional probability of $\boldsymbol{C_j}$ given $\boldsymbol{C_i}$. Probability for occurrence of $\boldsymbol{C_S}$ is computed by Odds and $\boldsymbol{BNP}$. $\boldsymbol{OCT}$ is an arithmetic average of $\boldsymbol{OCP}$ of all $\boldsymbol{C_S}$, and we obtained 0.301.

$$\text{BN Odds } (\boldsymbol{BNO}) = \frac{1-\boldsymbol{P}(\boldsymbol{C_S})}{\boldsymbol{P}(\boldsymbol{C_S})}$$

$$\text{Occurrence probability } (\boldsymbol{OCP}) = \boldsymbol{BNP} * \boldsymbol{BNO}$$

Next, we applied $\boldsymbol{C_S}$ which is obtained from the testing datasets from the equations and get $\boldsymbol{BNP}$, $\boldsymbol{BNO}$, and $\boldsymbol{OCP}$. For validation, we need to compute *Fidelity* and then decide *Feasibility*. *Fidelity* is measured by the rate of $\boldsymbol{OCP}$ of the test datasets over $\boldsymbol{OCT}$. When the $\boldsymbol{OCP}$ is above $\boldsymbol{OCT}$, it guarantees that the $\boldsymbol{C_S}$ is reasonable and exact rate is not meaningful. Thus, we roughly set *Fidelity* with maximum value which represents 100%, which means the rate is 1.

The *Fidelity* is categorized under three Feasibility states. The three categories are defined in TABLE I. The category *possible* presents a given $\boldsymbol{C_S}$ may be feasible and acceptable even though it is not fully guaranteed. Note that it is based on probabilistic analysis and the $\boldsymbol{OCT}$ and $\boldsymbol{OCP}$ is affected as it is trained and/or tested by more datasets. In our experiment, we observed the range for the *possible* category is 64%.

TABLE I. Fidelity-Feasibility Categorization

Category	Range	Perception about a given $\boldsymbol{C_S}$
feasible	100%	Feasible and solid
possible	64% ~ 99%	Probably feasible and acceptable
infeasible	~ 63%	Not feasible

TABLE II shows $\boldsymbol{OCP}$ of each training dataset. Out of the 11 datasets, 6 show 100% that stands for feasible schedules, 3 are in the possible range (64% ~ 99%) that is they might be reasonable. Thus, they are all considered as feasible cases. On the other hands, two datasets (of 4 and 9) shows below the range. The contextual schedules of the datasets are very rarely occurring and eventually have very low conditional probability. In summary, the proposed approach successes in reasoning contexts in overall 81% datasets.

TABLE II. Fidelity and feasible-decision table by occurrence probability. We declare *possible* feasibility between 64% to 100%.

Dataset No	BNP	OCP	Fidelity	Feasibility
1	0.048	0.214	71.11%	Possible
2	0.154	0.412	100%	Yes
3	0.206	0.360	100%	Yes
4	0.005	0.051	16.91%	No
5	0.048	0.214	71.11%	Possible
6	0.048	0.214	71.11%	Possible
7	0.154	0.412	100%	Yes
8	0.206	0.360	100%	Yes
9	0.015	0.153	50.72%	No
10	0.206	0.360	100%	Yes
11	0.206	0.360	100%	Yes

VI. CONCLUSION

The proposed approach conducts a reasoning process to determine the present context in a smart space. We employ mathematical and statistical analyses including Euclidean distance, Cosine Similarity and PCA to determine present context. PCA is found to reduce the computational complexity of our method by filtering out sensors which are not contributing or minimally contributing to the context. Our experimental validation shows positive performance result. In the future, we are planning to research and develop methods to optimally map the present context to most relevant services for the human residents in the space. We will also examine unsupervised learning methods, and assess their success in recognizing present contexts without training. This would be important for real-world applications where training may not be possible.

REFERENCES

[1] D. G. Abowd, A. K. Dey, J. P. Brown, N. Davies, M. Smith, and P. Steggles, “Towards a better understanding of context and context-awareness,” Intern. Symp. on Handheld and Ubiq. Comput. Germany, 1999, LNCS, Vol. 1707, pp.304-307.

[2] S. S. Yau, Y. Wang, and F. Karim, “Development of situation-aware application software for ubiquitous computing environments,” IEEE Intern. Comp. Softw. and Appl. Conf. Oxford, UK, 2002, IEEE Computer Society Washington, DC, pp.233-238.

[3] J. W. Lee, A. Helal, Y. Sung, K. Cho “A context-driven approach to scalable human activity simulation,” ACM SIGSIM Conf. on Principles of Advanced Discrete Simulation, New York, 2013, pp.373-378.

[4] J. W. Lee, S. Helal, “Context awareness computing in smart spaces using stochastic analysis of sensor data,” 2nd Intern. Conf. on Intel. Human Syst. Integration, California, 2019, pp.3-9.

[5] G. Fischer, “User modeling in hu0.412man-computer interaction,” J. of User Modeling and User-Adapted Interaction, 11(1-2), 2001, pp.65-86.

[6] D. Salber, A. Dey, G. Abowd, “The Context Toolkit: aiding the development of context-enabled applications,” Conf. on Human Factors in Computing Systems, ACM New York, 1999, pp.434-441.

[7] M. Hasan, A. Roy-Chowdhury, “Context aware active learning of activity recognition models,” IEEE Intern. Conf. on Computer Vision, IEEE Express, Washing DC., 2015, pp.4543—4551.

[8] J. Park, M. Moon, S. Hwang, K. Yeom, “CASS: A context-aware simulation system for smart home,” 5th ACIS Intern. Conf. on Software Engin. Resear., Management & Appl., IEEE Express, 2007, pp. 461-467.

[9] X. H. Wang, D. Q. Zhang, T. Gu, H. K. Pung, “Ontology based context modeling and reasoning using OWL,” IEEE Annual Conf. on Pervasive Comput. and Comm. Workshops, IEEE Washing DC., 2004, pp.18-22.

[10] Z. S., Abdallah, M. M. Gaber, B. Srinivasan, S. Krishnaswamy, “CBARS: Cluster based classification for activity recognition systems,” Intl. Conf. on Adv. Mach. Learni Tech. and Appl. Springer, Berlin, 2012, pp.82-91.

[11] E. Hoque, J. Stankovic, “AALO: Activity recognition in smart homes using active learning in the presence of overlapped activities,” Intl. Conf. on Perv. Comput. Tech. for Healthcare, IEEE Express, 2012, pp.139-146.

[12] A. Jurek, C. Nugent, Y. Bi, S. Wu, “Clustering-based ensemble learning for activity recognition in smart homes,” Sensors 14(7), 2014, pp.12285-12304.

[13] X. Wu, “Context-aware cloud service selection model for mobile cloud computing environments,” J. Wireless Comm. & Mobile Comput., 2018, Vol. 2018 (6). pp.1-14.

[14] D. Shin, J. Lee, J. Yeon, and S. Lee, “Context-aware recommendation by aggregating user context,” IEEE Conf. on Commerce and Enterprise Comput., Austria, 2009, pp.423-430.

[15] K. Madadipouya, S. Chelliah, “A literature review on recommender systems algorithms, techniques and evaluations,” Brain: broad research in Artificial Intelligence and Neuroscience, 2017, Vol 8(2), pp.109-124.

[16] J. W. Lee, S. Cho, S. Liu, K. Cho, S. Helal, “Persim 3D: Context-driven simulation and modeling of human activities in smart spaces.” IEEE T. on Automation Science and Engineering 12(4), 2015, pp.1243-1254.

[17] A. Helal, W. Mann, H. Elzabadani, J. King, Y. Kaddourah and E. Jansen, "Gator Tech Smart House: a programmable pervasive space", IEEE Comput. Mag., 2005, pp.64-74