Towards Summarized Representation of Time Series Data in Pervasive Computing Systems

Faraz Rasheed, Youngkoo Lee, and Sungyoung Lee

Computer Engineering Dept. Kyung Hee University
446-701 Suwon, Republic of Korea
faraz@oslab.khu.ac.kr, {yklee, sylee}@khu.ac.kr

Abstract. Ubiquitous computing systems are connected with a number of sensors and devices immersed in the environment, spread throughout providing proactive context aware services to users. These systems continuously receive tremendous amount of information about their environment, users and devices. Such a huge amount of information deserves special techniques for efficient modeling, storage and retrieval. In this paper we propose the modeling of context information as time series and applying the time series approximation techniques to reduce the storage space requirements and for faster query processing. We applied an algorithm based on non-linear interpolation to approximate such data and evaluated the approximation error, storage space requirements and query processing time.

1 Introduction

Ubiquitous Computing envisions [1] an environment where devices and sensors can interact seamlessly to provide proactive services to users. There are a number of projects currently being done in ubiquitous computing field [2][3][4]. Context awareness is an integral feature of ubiquitous computing systems. In order to provide proactive services to user, a system must be aware of the users environmental context and user preference. We take context as anything that helps in situational understanding.

Usually a ubiquitous system is connected with a number of sensors which sense the physical and computational environment, the users present in the environment and their activities. Hence various types of data are continuously injected to the ubiquitous computing system. The size of this data grows tremendously with time slowing down the query processing and other database operations. This huge amount of data requires careful management to address the issues in ubiquitous data management [5].

We proposed in our earlier papers the idea of knowledge aggregation which we call Context Summarization [6][7]. The idea is to represent the information in such a form that it takes comparatively less storage space and can still answer queries with appropriate precision. We have proposed several techniques that can be applied on different type of data.

Our summarization process is totally hidden from the user and she still queries the data as she is using the original data. Since the summarization process change the

J. Ma et al. (Eds.): UIC 2006, LNCS 4159, pp. 658 – 668, 2006.

representation of the data transparently, there is a need for query answering system which can translate the queries on original data so that they can be answered using the aggregate information store. In this paper we are going to describe this query answering component of our summarization system. We will explain how this query answering system can be applied to time series data which includes temperature values, humidity values, available network bandwidth at particular time, network bandwidth usage by particular user, light intensity, etc.

2 Context Summarization

Context Summarization is a process of approximation or the compact representation of data such that it consumes relatively less storage space and can still answer user queries with acceptable degree of confidence. Most of the data received by pervasive system is for the software system's (middleware and application) internal use so that they can process it and provide the proactive services to users promptly and efficiently. Thus we can change the internal representation of this data such that most of the queries can be answered using the summarized information.

Ubiquitous computing systems are equipped with a number of sensors, most of which continuously emit data at different constant rates. Such sensors include temperatures, humidity, light intensity, noise level, signal strength, available network bandwidth, network utilization, etc. This kind of data which is received continuously can be modeled as time-series. The approximate representation of time series can be achieved by selecting the representative points which provides the best approximation to achieve the desirable degree of confidence value. The confidence value is the amount of precision provided by a particular summary of the data. The precision can be modeled by the overall error or the amount with which the summary representation differs from the original time series. Different error measures can be used like Euclidean distance, root mean square error, averaged error. For our experiments, we have used the root mean square error as it is more widely used in approximation theory. The queries could be answered by interpolating through these points or by evaluating the fitted curve to a selected portion of the data.

Such a summarization not only reduces the storage space consumption but also improves the time taken to answer queries. As the ubiquitous systems are essentially distributed, such a summarized representation also improves the network usage for data migration among different nodes. It also improves the efficiency of knowledge reasoning, user preference and machine learning and data mining by minimizing the size of available data and thus the query processing time.

We are using our project called CAMUS (Context Aware Middleware for Ubiquitous Systems [4]) to apply the summarization techniques. Hence, all these techniques are implemented inside the middleware which also host the knowledge repository and provides query interface to user. In the next section we will explain the architecture of our system and in the subsequent section we will show how do we apply the summarization techniques and answer queries over numerical type context. The architecture of our summarization module has been discussed earlier in [7].

3 Applying Query Answering Interface (QAI)

Now we will describe the application of summarization and query answering using time series data (temperature values). A worth mentioned point is that the ubiquitous systems are very resource hungry as they are performing a lot of operations; receiving continuous stream of data from sensor, data filtering, modeling, storage, data processing, reasoning, situational understanding of current context, finding and providing the appropriate service and so on. Hence, the summarization module should be very smart and efficient. That is the reason why we have selected to apply only simple, optimized and less complex techniques to summarize the information and to calculate and return the query result.

We will present a summarization algorithm to approximate time series data based on non-linear Interpolation with equi-spaced points. For interpolation, we used cubic-spline and piece-wise hermite interpolation techniques which we found better in terms of query response time and error. In these techniques an important task is the selection of equi-spaced points which provides the best approximation of data. We have developed an algorithm to select such points. This smaller number of representative points is stored in database and used later to reply queries by interpolating through these points.

The algorithm attempts to achieve the approximation with a specific precision value (which is defined as root mean square error: rmse). But it does not guarantee that such an approximation would be achieved as it also considers the storage space requirement by approximation compared to the original dataset.

3.1 Time Series Numerical Valued Information – Non-linear Interpoaltion

Interpolation is a method of constructing new data points from a discrete set of known points. While linear interpolation connects the discrete points by straight line, non-linear interpolation techniques connects the discrete points using polynomials. We have utilized two of the most famous interpolation techniques namely spline and hermite interpolation. Let we have a temperature sensor reporting the temperature of location 'X' after every 5 minutes. Table 1 presents a sample data set

Table 1. Temperature values received after every 15 minute

Time	Temp.
2:05	35.5
2:10	34.5
2:15	38 °C
2:20	38.5
2:25	35.5
…	

Now let we summarize the information by selecting the points sampled at larger interval than 5 minutes (the original data arrival rate) as in figure 1.

SummaryMasterTable		
ID	**Interval_Start**	**RMSE**
23	2004-06-14 00:00	1.43
24	2004-06-14 03:00	1.42
25	2004-06-14 06:00	1.34
...	...	...

SummaryChildTable		
ID	**PointX**	**PointY**
23	1	35.5
23	9	39
23	17	35
23	25	36.5
23	33	37.5
23	36	34
24	1	34
...	...	...

Fig. 1. Sample Summary Tables (Non-linear interpolation)

Here the selected points for approximation are variable. Each record in summary table has the interval start time and the approximation error. The summary child table contains the selected points for each of these intervals.

The input to this algorithm is a set of 'n' data points and the maximum approximation error (max_error) that can be tolerated. The output is the set of 'k' points where k < n and the actual approximation error as root mean square error.

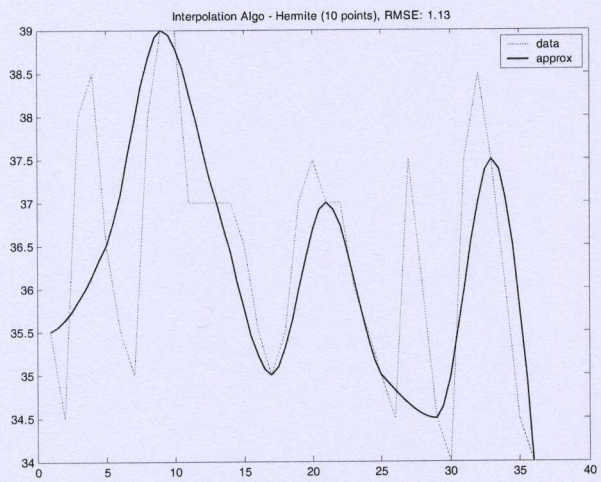

Fig. 2. Time series approximation using non-linear (Hermite) interpolation

The target is to find the set of 'k' data points representing the given-time series so that when interpolated through these points, the error value (rmse) is less than or equal to the max_error. The algorithm proceeds with 'c' which represents how many points should be approximated by each selected points. Practically, it is the gap

between selected points. Hence if c = 4, it would pick 1, 5, 9, 13, 17,... elements of time series. It then uses these points for interpolation and generating the approximate time series. As 'c' increases, the number of selected points 'k' decreases which results in the increase in 'rmse'. The algorithm attempts to find the optimal value of 'c' using which the desired precision can be achieved. The algorithm only attempts to find the c in a particular limit such that 'c1 <= c <= c2, where c2 is the minimum and c2 is the maximum value of 'c'. Above c2, the algorithm would select too many points which is against the spirit of summarization to save the storage space. Below c2, the algorithm would select too fewer points to deteriorate the rmse. Hence a careful selection of c, c1 and c2 is required. Figure 2 demonstrates the approximation done by non linear interpolation. The algorithm to find 'k' points is presented below:

```
1. start with c = v, mode = 'start' // v: arbitrary value
2. Find k-points with c
3. Find rmse with these k-points
4. Switch(mode)
5.a mode = start
      rmse > max_err
            c = c - 1
            set mode 'dec'
            goto 2
      rmse <= max_err
            c = c + 1
            set mode 'inc'
            goto 2
5.b mode = dec
      rmse <= max_err
            save points
            return
      rmse > max_err
            if c = c1          // minimum value of c
                  save points
                  return
            else
                  c = c - 1
                  goto 2
5.c mode = 'inc'
      rmse >= max_err
            c = c - 1
            set mode 'dec'
            goto 2
      rmse < max_err
            if c = c2          // maximum value of c
```

```
            save points
            return
      else
            c = c + 1
            goto 2
```

3.2 Query Answering and Discussion

Suppose we receive a query for attribute 'v' at timestamp 't' in data table 'dt'

SELECT 'v' FROM 'dt' WHERE timestamp = 't'

Query processing starts with searching the interval, in summary table, which contains the desired timestamp. Then it retrieves the points for interpolation from the child table. These points are then interpolated to find out the attribute value for 'v' at timestamp 't'.

The advantage of this technique is that it reduces the storage space significantly. The query processing is also simple and since the interpolation techniques generate a continuous curve, the summary representation can even answer for the timestamps for which there is no attribute value in the original data table. For example, if original data is received after every five minutes 2:00, 2:05, 2:10, ... and there is a query for timestamp 2:13, we can still interpolate and find out the approximate value. This could be important in certain scenario processing when the application wants some information at the time when there is interesting activity sensed by some other sources. Since query processing using summary tables involve two table lookups (compared to one table lookup in original data table), for the query to be processed faster there should be significant number of records in original table. In other words, the interpolation based summary would generally perform faster if the original data table contains very large amount of data compared to the summary tables.

4 Experiments and Evaluation

We have evaluated the summarization of time series data from three perspectives. First of all we have compared the storage space consumption by raw data and information summarized by different techniques. Secondly, we applied the similar queries on both raw data and summarized information and compared the time taken by both of these representations. Finally, we calculated the root mean square error incurred while answering the queries using various summarized representations.

4.1 System Specification

We performed all the experiments on Toshiba notebook using Pentium Mobile using Intel Centrino technology with 1500 MHz processor and 512 MB RAM. We are using Windows XP Professional OS. Our summarization module is implemented using Java programming language. We used mySQL v4.0 as DBMS (running on localhost) and also used MATLAB 6.5 for interpolation techniques. The system was not

considerably busy with other process while we performed the experiments and we did not use multiple threads for processing.

4.2 Time Series Numerical Valued Context Information (Temperature)

We used the data recorded by Cornell Lab of Ornithology for Bird population studies (USA). They installed a temperature logger in bird's nest and recorded the temperature after every 5 minutes. We used the data recorded in 88 days which amounted to around 22,000 records. We summarized this data over 2 hours interval using algorithm described in section 3.1. For query processing we interpolated through these points using cubic spline and piecewise hermite interpolation techniques. The effect on the storage space consumption by the summarization process is presented in the Table 2

Table 2. Storage space consumption by raw and summary data

	Raw data	**Summary data**
Rows	21,978	924
row size	48	97
table size	1035 KB	90 KB
Days	88	88
Interval	5 min.	120 min.

Obviously, the summary table takes less storage space than raw data. The row size in summary table is larger than that of raw data table as it contains more fields. The same information is presented in graphical form in Figure 3.

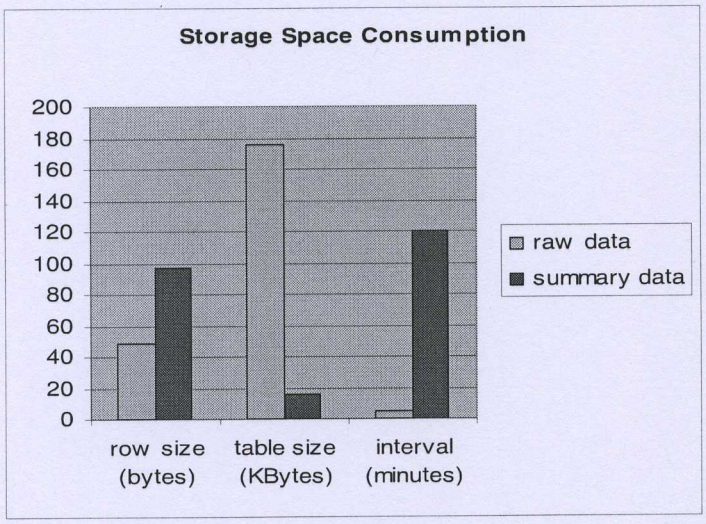

Fig. 3. Comparison of storage space consumption

Now for the time taken by query processing, we applied the simple query to find the temperature at a specific time over 22000 raw data records and 900 summary records. We calculated the time consumed to reply this query over raw data and the summary data calculated by 2 techniques described earlier. Chart in figure 4 compares the time taken to answer query by raw data table and tables using summarized representation.

The time taken to answer query using the interpolation (60ms and 46 ms) is less than that of raw data (140 ms). This is because the raw data is querying over much larger table (22000 records versus 900 records of summary table). The hermite interpolation is processing queries a little faster than cubic spline interpolation.

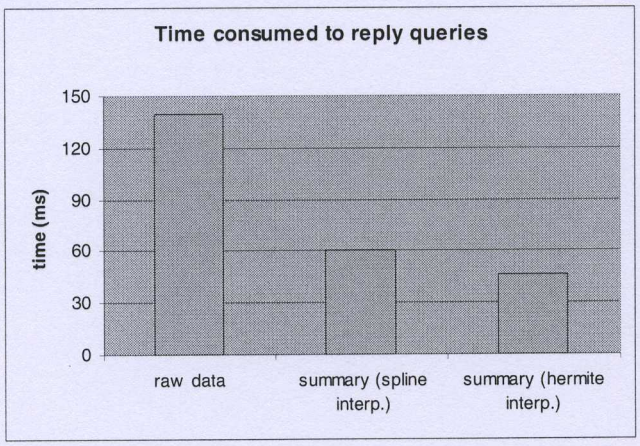

Fig. 4. Time taken to reply queries

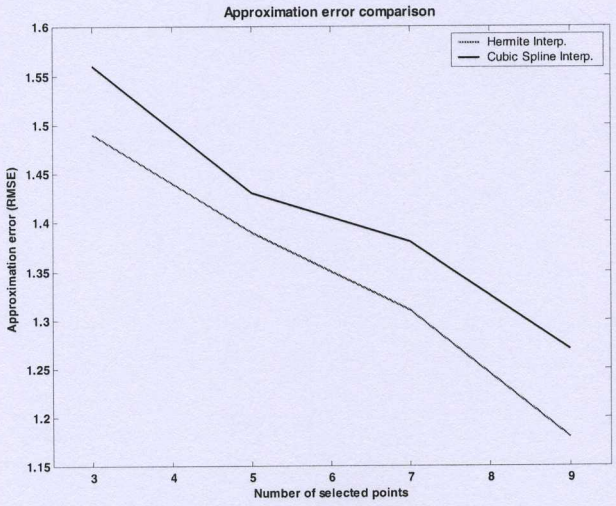

Fig. 5. Comparison of approximation error

Finally, to calculate the lost in precision caused by these queries, we used the root mean square error. The approximation error is calculated by keeping the storage space constant that is by taking the same number of points to interpolate using cubic spline and piecewise hermite interpolation. Figure 5 presents the comparison of approximation error:

It is obvious that as we increase the number of points, the approximation error is reduced. It is also important to note that the approximation error for hermite interpolation is less than the cubic spline interpolation for the same number of selected points.

5 Issues and Challenges

As we mentioned earlier that ubiquitous computing systems are resource hungry because of their distributed nature, a number of diverse sensors and devices they are to deal with and for the smart service selection and service delivery. Hence, performance is reasonably important issue for each of its components. That is the reason why we are keeping our summarization system as thin as possible which can work with very limited resources efficiently. There are quite a few mathematical and graph based techniques like curve fitting with limited points and interpolation to find the missing point in the graph but we are very selective because of the inevitable complexity. Precision and accuracy is another important issue. The tasks that are usually performed using context repository like preference and machine learning, logic reasoning, intention prediction and activity recognition; all of them require data with higher degree of precision. The more we summarize the information, the greater is the loss of precision. To reduce this precision loss we can use more comprehensive techniques but then there is a trade-off between precision and performance. But the important thing is that as the size of data reduces, all the ubiquitous system components that use the context repository tend to perform faster. Security is another significant issue. Since the summarization techniques change the representation of data, hence if the summarization is not done properly, the results produced would be misleading. Hence, summarization components are required to be designed and implemented with great care.

6 Related Work

Several existing systems support techniques like feature extraction and generalization [8][9][4] but we want to formally make summarization as part of the ubiquitous system's data management. Our idea is to generate summaries so that later we don't need the raw data any more and can reply to most of the queries with this summarized information with acceptable degree of confidence.

In DBMS, data mining and data ware housing [10] use the concept of histogram and multidimensional views of database and work on the aggregate, consolidated data instead of raw data to support the higher level decision making and to identify the hidden patterns in the data. The goal of data mining and OLAP is somewhat similar

but we want to transform the raw context to summarized form taking less storage space and provide improved and efficient reasoning and machine learning.

Researchers in DBMS have also analyzed the time series data streams for very large databases [11]. Here, they analyze the data coming in continuous streams with time. They have proposed solutions on how to manage, represent and store the time series data streams. This is also highly related to the context summarization.

7 Future Work and Conclusion

For the future work, we are currently trying to summarize the location map of wifi based location awareness system. This map contains the signal strength received at different locations by the device from access points in the surrounding. The size of this data is very huge; just for few floors the number of records grows in multiple of 10,000! We are also optimizing our Query Answering Interface (QAI) so that it can efficiently use multiple summary representations of different strengths to answer queries with different precision requirements accordingly. In the conclusion, we would say that data aggregation and context summarization is an interesting research area which needs to be further explored. We have presented in this paper the summarized representation of numerical valued time series information, also how and which kind of queries they can answer. The proper application of summarization process does conserve storage space and improves the query processing and data migration in distributed ubiquitous computing environment as supported by our experimental results.

References

1. M. Weiser, The computer for the 21st century. *ACM SIGMOBILE 1999 Review*
2. Chen Harry, Tim Finin, and Anupam Joshi: An Intelligent Broker for Context-Aware Systems. In: *Ubicomp 2003*, Seattle, Washington
3. Gaia: A Middleware Infrastructure to Enable Active Spaces. Manuel Román et al., In *IEEE Pervasive Computing*, Oct-Dec 2002
4. Hung Q. Ngo, Anjum Shehzad, Saad Liaquat, Maria Riaz, Sungyoung Lee: Developing Context-Aware Ubiquitous Computing Systems with a Unified Middleware Framework. *EUC 2004*: 672-681
5. Michael J. Franklin, Challenges in Ubiquitous Data Management. . Informatics: 10 Years Back, 10 Years Ahead, LNCS #2000, R. Wilhiem (ed)., Springer-Verlag 2001
6. Faraz Rasheed, Young-Koo Lee, Sungyoung Lee, Context Summarization & Garbage Collecting Context, UWSI 2005, In the proceedings of ICCSA 2005 page 1115-1124, Singapore, published by Springer Verlag
7. Faraz Rasheed, Young-Koo Lee, Sungyoung Lee, "Towards Using Data Aggregation Techniques in Ubiquitous Computing Environments," *percomw*, pp. 369-372, Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06), 2006
8. Mike Spreitzer, Marvin Theimer, Providing location information in a ubiquitous computing environment, *ACM SIGOPS Operating Systems Review , Proceedings of the fourteenth ACM symposium on Operating systems principles* Dec 1993, Volume 27 Issue 5

9. Jason I. Hong, James A. Landay, Support for location: An architecture for privacy-sensitive ubiquitous computing, *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, June 2004
10. Alex Berson , Stephen J. Smith, Data Warehousing, Data Mining, and OLAP, McGraw-Hill, Inc., New York, NY, 1997
11. Lin Qiao et al, Data streams and time-series: RHist: adaptive summarization over continuous data streams, *Proceedings of the eleventh international conference on Information and knowledge management*, Nov 2002