International Journal of Software Engineering and Its Applications
Vol. 9, No. 1 (2015), pp. 105-112
http://dx.doi.org/10.14257/ijseia.2015.9.1.09

Clustering for Context Inference in the Data Stream Mining

Shinsook Yoon and Chang-Keun Ryu[1]

Department of Information and Communication, Korea Nazarene University
[1]Department of Electronics, Namseoul University
{yss28@daum.net}, [1]{ckryu@nsu.ac.kr}

Abstract

In an environment in which several events are sensed in a complex manner and sequentially obtained, a clue can be obtained for inference of situations by classifying each event and analyzing the aspect of change of each event. The study proposes a method to efficiently decide the cluster centers in each subsequent time slot for efficient classification of events and inference of situations in a data stream environment. For the data stream under this condition, each time slot classified at a certain interval is set up, the events using clustering in each time slot are carried out, and to recognize how the aspect of change of each event sensed in a continuous time slot is carried out, the cluster centers are allowed to be rapidly captured.

Keywords: *Context inference, Data stream, Clustering, k-means, Time slot*

1. Introduction

In analyzing continuous variable data massively flowing in, data clustering helps classify and identify data in which various events. There is a method to analyze and classify characteristics of source signals in an environment in which high-capacity data sequentially flow in, but in a stream environment in which the data flow should be analyzed once on the memory, understanding and classifying characteristics of the data often have high efficiency. Clustering is a good tool to analyze high-capacity data, but in an environment in which new high-capacity data are sequentially flow in, it is a method needing reasonable adjustment as well.

In an environment in which several events are sensed in a complex manner and sequentially obtained, a clue can be obtained for inference of situations by classifying each event and analyzing the aspect of change of each event. To consistently monitor events identified and classified using clustering in the subsequent time slot at this time, *k*-means clustering with an advantage at the processing speed is preferred, and what is significant in *k*-means clustering is to rapidly understand the cluster centers. This refers to a method to rapidly decide the cluster centers when a cluster is classified in each continuous time slot [5, 8, 9].

This study proposes a method to efficiently decide the cluster centers in each subsequent time slot for efficient classification of events and inference of situations in a data stream environment. Data that flow in sequentially are classified at a certain time interval, and different individual events are mixed up in the sensor data obtained in each time slot. For the data stream under this condition, each time slot classified at a certain interval is set up, the events using clustering in each time slot are carried out, and to recognize how the aspect of change of each event sensed in a continuous time slot is carried out, the cluster centers are allowed to be rapidly captured.

This study is composed as follows: Chapter 2 summarizes related research, Chapter 3 proposes a method to quickly identify the cluster centers and efficiently analyze data

[1] Corresponding Author

ISSN: 1738-9984 IJSEIA

stream based on that. Chapter 4 makes experiment and evaluation of the proposed hypothesis and Chapter 5 draws conclusions.

2. Related Studies

Through a cluster analysis, characteristics of the whole data can be understood by checking the representative value of each cluster instead of checking millions of data directly.

Table 1. Clustering Type Compare

Clustering type	Advantage	Disadvantage
k-means	Fast O(*tkn*) *t*=semi-multiple, *k*=number of cluster, *n*=number of data	May be a local optimum Ambiguous to calculate the mean of categorical value K value should be set up in advance. Vulnerable for an outlier or noisy one Difficult for data other than circular ones
k-medoid	Compensate disadvantage of *k*-mean, which is vulnerable for an outlier. Use medoid instead of centroid	Slow O(*k* * pow((*n*-*k*), 2))
Clara	Sampled *k*-medoid Faster than *k*-medoid	Result value depends on the sample

1) Partitioning: refers to classifying data into section and calculate their center.

2) *k*-means: a method of dividing each section, finding centroid, dividing section again based on the centroid and finding a centroid again if there is any change[4].

Its advantage is fast, O(*tkn*): *t*=semi-multiple, *k*=number of the cluster, *n*=number of data. Generally, since *n* is very large, it gets similar to O(n). Its disadvantage is that it may be local optimum. At this time, it should be solved by changing the start point. It is ambiguous to calculate the mean of a categorical value. *k*-value should be set up in advance. It is vulnerable for an outlier or noisy, and it is difficult to cluster data other than circular ones. Differing the method of calculating the distance is a solution. It is calculated considering frequency information.

3) *k*-medoid: This compensates the disadvantage of *k*-mean vulnerable for an outlier, using a medoid instead of centroid.

Its disadvantage is slow processing speed, O(*k* * pow((*n*-*k*), 2)) To solve this, sampling is done[4].

4) Clara: This is sampled *k*-medoid. The advantage of this means is that it is faster than *k*-medoid, but it has a disadvantage that the result value depends on the sample [4, 6, 7].

5) Minimum Spanning Tree(MST)

This is a hierarchical clustering means, in which all of the first input data points form individual clusters. Clusters to which two data points with the shortest distance belong merge. Merging continues until there are k clusters with the same means. This is quite sensitive with an outliter and if the outliers are in a row, chaining effect occurs. In addition, since it uses all data information in the cluster as cluster merging information, execution speed is very slow.

6) Clustering Large Application based on RANdomized Search (CLARANS)

CLARANS algorithm is one combining PAM and CLARA, and CLARA uses fixed schedule sample only in all stages of search while CLARANS extracts a random sample from the whole data set. Its clustering process is thought to be search of a graph, the set of all *k*-medoids. If one medoid differs in a two *k*-medoid set, it is called a neighbor, and the number of neighbors randomly chosen is decided by variable max neighbors. If a better neighbor is found, CLARANS moves to the neighbor node to repeat the above process, and if not found, it chooses the clustering using the current *k*-medoid as a local optimal solution. If the local optimal solution is found, CLARANS chooses a random new node to search a new local optimal solution. The number of the local optimal solutions to search is decided by parameter NumLocal[2].

7) Clustering Using Representatives(CURE)

This is a hierarchical means to merge all data entered from an individual cluster till it becomes k clusters, but it does not calculate all points in the cluster, but uses the representative value of c clusters. These representative values are points calculated by choosing c points adequately distributed, which can express the shape of the clusters from each cluster and reducing up to α, using c representative values, it improved MST which is sensitive to an outlier and has a disadvantage of chaining effect. It improved the defect in which one cluster is separated into two or more, which is a disadvantage of centroid-based approach, caused by its use of one representative value [1].

8) Projected Clustering (PROCLUS)

Existing algorithms are those considering all dimensions of data. As mentioned in the clustering of higher-order data, as data dimensions increase, due to a problem of sparsity of the data, clustering considering all dimensions significantly decreases performance. PROCLUS finds clusters only on some dimensions not the entire dimensions. Projected cluster refers to a cluster existing only in some dimensions out of the entire dimensions D. PROCLUS is an algorithm, adding a process of finding related dimensions to CLARANS, and it chooses k medoid as in CLARANS algorithm, and then finds the dimensions with high relevance with each medoid for clustering. It repeats this process as many times as the time of iterations, and takes the clusters with a *k*-medoid with the lowest similarity in the result as a result [3].

3. A Novel Way of Data Stream Mining with Clustering

In the Internet of things environment and ubiquitous sensor network environment, a sensor senses the relevant object or peripheral environment status and reports change over a certain level to the host. The sensor is distributed or attached to the entire regions. However, sometimes, several sensors commonly recognize a single event and sometimes, they recognize several events. If various signals are obtained in measuring equipment or sensing device, in the signal processing domain, the signals can be classified and identified with technology like analysis of frequency. However, in an environment in which high-capacity variable data sequentially flow in, it is necessary to efficiently process the converted data using the assets in the arithmetic section of the computer. This is because in the Internet of things environment or Ubiquitous sensor network environment, all values converted to data are measured in the form of signals.

Analyzing continuous sensor data obtained variably and at high-capacity is the major interest of the recent research of data, and the method of quickly classifying various events included in the data stream sequentially flowing in is a significant issue. For this purpose, data clustering can be a useful method. This is the classification while prior knowledge about each class to classify in the high-capacity database has not been secured, and classification of data with affinity is data clustering. A cluster refers to a set of pattern

forming a mass gathering close to each other by a finite number of patterns given in the pattern space.

In the meantime, major data clustering research so far targeted high-capacity database while it seeks how to do that in a data stream environment is a research task of interest. What precedes in analyzing the data stream is cutting the data stream at a certain time interval. Next is analysis of data in the time section. It sets up a time interval with a reasonable means for a series of data stream and uses clustering in analyzing the data on each time section. The method most often used in clustering is *k*-means clustering. However, there is a problem to solve for using *k*-means clustering in a data stream analysis sequentially obtained. To evaluate the similarity of clusters and classify data, the central point in each subsequent time slot should be quickly decided. This study proposes a method of clustering data in each time slot and calculating the central point of the cluster efficiently in each time slot to classify this when complex events are implied in single sensor data for the data stream obtained variably along the passage of time.

This study proposes the procedures of finding the cluster centers in each time slot to classify events in an analysis of continuous data stream as follows:

Suppose that data streams $DS =< D_1, D_2, D_3, \cdots, D_t >$ enter respectively into time slots $TS =< T_1, T_2, T_3, \cdots, T_t >$.

Clusters $C_1, C_2, \cdots, C_k$ are found through *k*-means clustering by each time slot. The process is as follows:

1) *k* random samples initially obtained in the first time slot T_1 are chosen, and the initial cluster centers are set as $\mu_1, \mu_2, \cdots, \mu_k$. When the range of the chosen *k* value is [a,b], the value within this range is randomly selected to set as the cluster centers, and the center with the minimum error can be chosen.

Then, the cluster center obtained in the immediately prior time slot is taken to set as the initial center.

In other words,

When $\mu_1, \mu_2, \cdots, \mu_k \in T_{n-1}$,

In T_n, initial cluster centers are defined as follows

$(\mu_{n1}, \mu_{n2}, \cdots, \mu_{nk})^T = (\mu_{n-11}, \mu_{n-12}, \cdots, \mu_{n-1k})^T$.

2) For data $a \in A_i$ entered at T_i *(however,* $1 \leq i \leq t$ *)* , $min\|a - \mu\|^2$ is calculated, and then the smallest μ_j $(1 \leq j \leq k)$ is classified as the *j*th cluster.

3) Then, μ_i is renewed with the mean of the sample assigned to the *j*th cluster $1/n(C_j) \sum_{a \in C_j} a$.

4) Repeat till the mean $\mu_1, \mu_2, \cdots, \mu_k$ that minimize the variance of C_i, $1/n(C_i) \sum_{a \in C_i} \|\mu_i - a\|^2$ is found.

5) Repeat the above processes till it is the same as the μ_j value in the previous loop. Repeat the break 10 times max.

The above algorithm is expressed in a pseudo code as follows:

Table 1. Algorithm in a Pseudo Code

Input : $DS =< D_1, D_2, D_3, \cdots, D_t >$ $(D_t =< a_{t1}, a_{t2}, a_{t3}, \cdots, a_{ti} >)$
Output : $< C_1, C_2, C_3, \cdots, C_k >_t$
$< \mu_1, \mu_2, \mu_3, \cdots, \mu_k >_t$
$< \sigma_1, \sigma_2, \sigma_3, \cdots, \sigma_k >_t$
Initialize
1. t=1 //timeslot
2. i=0 //repeat number
3. k=3

4. l=*max(a)*, $a \in D_t$
5. while do
6. while do
7. i=i+1
8. if t=1
9. then random select $\mu_1, \mu_2, \mu_3, \cdots, \mu_k \in [0, l]$
10. else if t>1
11. then $< \mu_1, \mu_2, \mu_3, \cdots, \mu_k >_t = < \mu_1, \mu_2, \mu_3, \cdots, \mu_k >_{t-1}$
12. while (n≤k)
13. n=n+1
14. for m=1 to k
15. for j=1 to k
16. if min$\left\|a_{tn} - \mu_j\right\|^2 = \|a_{tn} - \mu_m\|^2$
17. Then $a_{tn} \in C_j$, and goto 8th line
18. end for
19. end for
20. end while
21. r=0
22. for n=1 to k
23. if μ_n =average(C_n),
24. then r=r+1 and $\mu_n = \mu_n$
25. else
26. then μ_n = average(C_n)
27. end for
28. if r=k break;
29. end while
30. t=t+1 //when we move to the another timeslot
31. print $< C_1, C_2, C_3, \cdots, C_k >_t \quad < \mu_1, \mu_2, \mu_3, \cdots, \mu_k >_t$
32. reset $< C_1, C_2, C_3, \cdots, C_k >_t \quad < \mu_1, \mu_2, \mu_3, \cdots, \mu_k >_t$
33. end while

The method proposed above is verified through an experiment and an evaluation of it is made in the next chapter.

4. An Experiment and Evaluation

Details of the experiment

1) A sound sensor was prepared to recognize the sounds generated by three types of sound sources.

2) Three kinds of event information were put in the sensor data that recognized the sounds 10 times per second and reported them to the host.

3) Time interval was set to 30 seconds for the sensor data obtained.

4) *k*-means clustering was made with the data collected within the time interval of 30 seconds.

5) According to the method made in the existing *k*-means clustering in the first time slot, the central point was calculated and found so that three event clusters would be classified.

6) From the second time slot and on, the cluster centers were found, according to the method proposed in this study in each time slot.

7) The number of times of finding cluster centers in each time slot was checked, and they were compared by each time slot.

Figure 1 below is a distribution chart of the entire data obtained by a sound sensor.

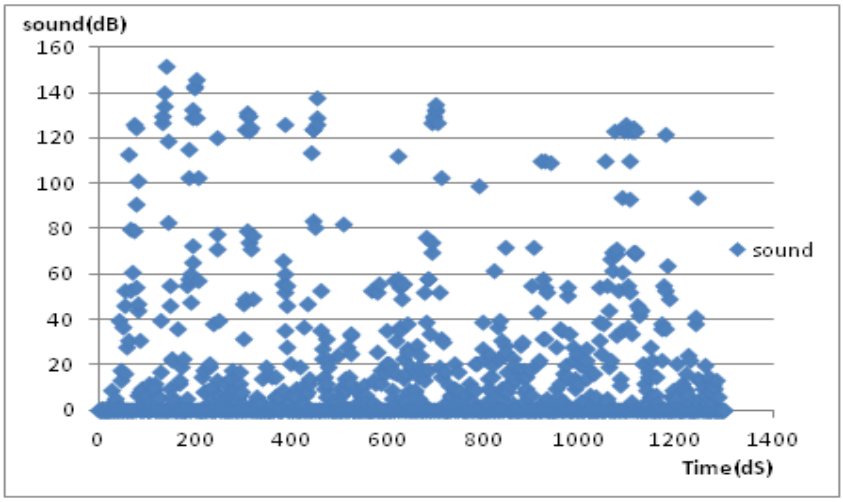

Figure 1. Distribution of the Data Sensed and Obtained by a Sound Sensor

In the above distribution chart, the horizontal axis indicates time (0.1 second), and the vertical axis, intensity of sound(dB).

As a result of an experiment, Figure 2 to Figure 5 below are results of clustering of the data obtained in each time slot.

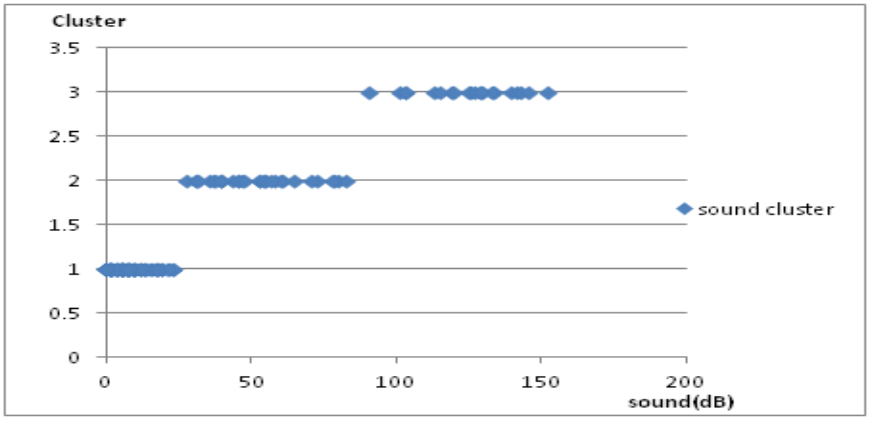

Figure 2. Results of Clustering of the Data

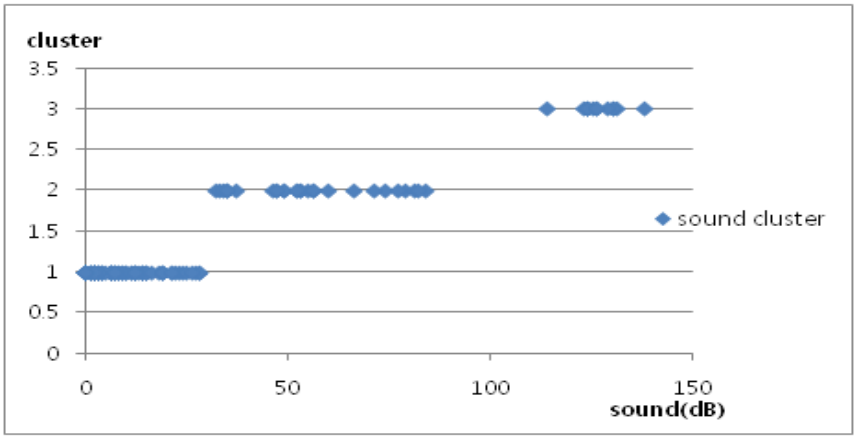

Figure 3. Results of Clustering of the Data

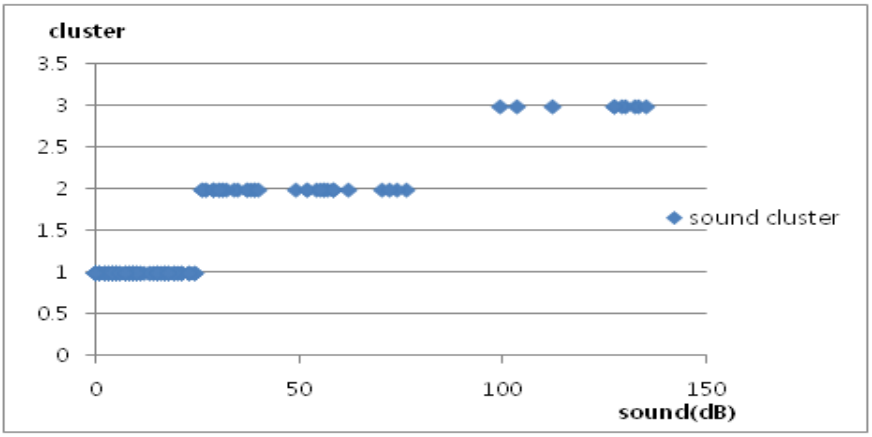

Figure 4. Results of Clustering of the Data

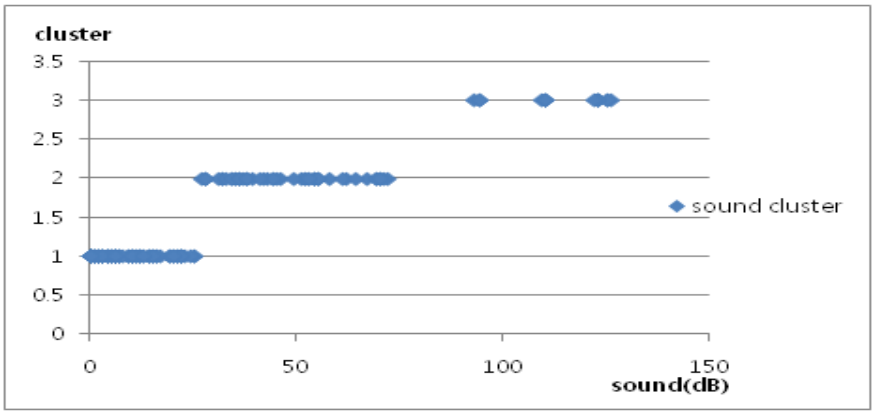

Figure 5. Results of Clustering of the Data

The vertical axis means classified cluster number (1,2,3) and the horizontal axis, intensity of sound (dB) value.

As a result of an experiment, in the section of 0 to 30 seconds, the calculation of the cluster centers could be completed at a loop of 9 times.

In the section of 30 to 60 seconds, the cluster centers could be calculated in 3 times, but since it was found at 6 times of calculation with the existing method, the new method shows a faster calculation. In the section of 60 to 90 seconds, the calculation of the center of each cluster was completed at 4 times, and it was found at 4 times as well in the existing method. It is found that there is no change in this section. In the section of 90 to 120 seconds, the newly proposed method could complete the calculation at 3 times, while the existing method could find the cluster centers at 5 times, so the new method could find the cluster centers in this section.

5. Conclusion

In the Internet of things or ubiquitous sensor network category, sensing activities of diverse sensors can be made. The details of sensing of a single event by various sensors can be reported, and a single sensor can sense and report several events as well. This study proposed a method of calculating the central point quickly in *k*-means clustering to classify various events included in data, when a single sensor senses and reports several events. Considering that it is fundamental in executing clustering and its priority task is to find the central point of the cluster, in finding the cluster centers more efficiently in a

subsequent time slot, follow-up classifying each cluster targeted and inferring situations considering the aspect of change of each event cluster will be helpful to secure efficiency.

Acknowledgements

Funding for this paper was provided by Namseoul University.

References

[1] S. Guha, R. Rastogi and K. Shim, "CURE: A Efficient Clustering Algorithm for Large Databases", Proceeding s of A CMSIGMOD, **(1998)**, pp. 73-84.
[2] R. Ng and J. Han, "Efficient and Effective Clustering Method for Spatial Data Mining", Proceeding s of the 20th VLDB Conference, **(1994)**, pp. 144- 155.
[3] C. C. Aggarwal, C. Procopiuc, J. L. Wolf, P. S. Yu and J. S. Park, "Fast Algorithms for Projected Clustering", Proceedings of the A CMSIGMOD International Conference on Management of Data, **(1999)** June, pp. 61-72.
[4] T. Kanungo, N. S. Netanyahu, C. D. Piatko, R. Silverman and A. Y. Wu, "An Efficient k-Means Clustering Algorithm", Analysis and Implementation, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, **(2002)** July.
[5] D. H. Suh and S. S. Yoon, "Weighting Method Based on Event Frequency for Multi-sensor Data Fusion in Wireless Sensor Network for the People with Disability", Journal of Assistive Technology, vol. 5, no. 1, **(2011)**, pp. 37-47.
[6] T. Kanungo, N. S. Netanyahu, C. D. Piatko, R. Silverman and A. Y. Wu, "An Efficient k-Means Clustering Algorithm, Analysis and Implementation", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, **(2002)** July.
[7] M. Dipti and T. Patel, "K-means based data stream clustering algorithm extended with no. of cluster estimation method", International Journal of Advance Engineering and Research Development (IJAERD), vol. 1, no. 6, **(2014)** June.
[8] M. Khalilian, N. Mustapha, N. Suliman and A. Mamat, "A Novel K-Means Based Clustering Algorithm for High Dimensional Data Sets", Proceedings of the International Multi-Conference of Engineers and Computer Scientists, vol. 1, **(2010)**.
[9] H. M. Koupaie, K. Lumpur, S. Ibrahim, K. Lumpur and J. Hosseinkhani, "Outlier Detection in Stream Data by Clustering Method", International Journal of Advanced Computer Science and Information Technology (IJACSIT), vol. 2, no. 3, **(2013)**, pp. 25-34.
[10] Y. Gu, L. Sun and J. Wang, "Comparative Analysis of Single and Mixed Spatial Interpolation Methods for Variability Prediction of Temperature Prediction", International Journal of Hybrid Information Technology, vol. 6, no. 1, **(2013)** January, pp. 67-76.
[11] S. Ji, L. Huang and J. Wang, "A Distributed and Energy-efficient Clustering Method for Hierarchical Wireless Sensor Networks", International Journal of Future Generation Communication and Networking, vol. 6, no. 2, **(2013)** April, pp. 83-92.
[12] M. Prabukumar and J. Cristopher Clement, "Compressed Domain Contrast and Brightness Improvement Algorithm for Colour Image through Contrast Measuring and Mapping of DWT Coefficients", International Journal of Multi-media and Ubiquitous Engineering, vol. 8, no. 1, **(2013)**, pp. 55-70.

Authors

Shinsook Yoon, she received the M.S. degrees in computer engineering from Hoseo University in 2008. Her research interests included in stream data processing and data fusion in wireless sensor network.

Chang-Keun Ryu, he is a professor at Namseoul University. He received the B.S. degree in electronic engineering from Dankook University in 1981, and the M.S. and Ph.D. in electronic engineering from Dankook University in 1993. His research interests are included in data fusion in wireless sensor network and system architecture.

