Distributed WSN Data Stream Mining based on Fuzzy Clustering

Hakilo Sabit, Adnan Al-Anbuky, Hamid Gholam-Hosseini
Sensor Network and Smart Environment Research Centre (SeNSe)
School of Engineering, AUT University
Auckland, New Zealand
{hsabit, adnan.anbuky, h.gholam}@aut.ac.nz

Abstract
This paper proposes a distributed wireless sensor network (WSN) data stream clustering algorithm to minimize sensor nodes energy consumption and consequently extend the network lifetime. The paper follows the strategy of trading-off communication for computation through distributed clustering and successive transmission of local clusters. We present an energy efficient algorithm we developed, Subtractive Fuzzy Cluster Means (SUBFCM), and analyze its energy efficiency as well as clustering performance in comparison with state-of-the-art standard data clustering algorithms such as Fuzzy C-means and K-means algorithms. Simulations show that SUBFCM can achieve WSN data stream clustering with significantly less energy than that required by Fuzzy C-means and K-means algorithms.

1. INTRODUCTION
The limitations of WSN nodes such as: memory, processing power, available energy, and wireless bandwidth have necessitated alternative computation and communication approaches than traditional methods for WSN to live up to their envisioned applications under such constraints. Traditionally, data from multiple sources (sensor nodes) are transmitted to a central location (sink) where efficient and complex analysis takes place. Despite its vast knowledge base and common applications, processing of data at a central location is inefficient and demanding for such energy and bandwidth constrained systems. In a densely deployed WSN, massive quantity of data is generated at very high frequency. Such that transmitting every bit can shortly exhaust the limited onboard energy and results in short network lifetime. Besides, such a fast data stream transmission over limited bandwidth results in frequent communication packet losses. Furthermore, centralized processing scheme is too slow for applications with low latency requirements and leaves most of the processing capability of sensor nodes unutilized. Therefore, an alternative data processing method is required to efficiently operate the network for extended period of time off the limited onboard energy storage as battery replacement is not an option for unattended operation requirement of WSN applications.

Transmission is known to be the largest consumer of energy in WSN [1]. For the energy cost of transmitting 1Kb over a distance of 100 meters, a general purpose processor with 100MIPS/W power could execute ≈ 3 million instructions [2]. Hence performing in-network distributed data processing closer to the sources is an energy efficient alternative, in other words trading-off communication for computation. Therefore, in this paper we present distributed clustering of WSN data streams using the SUBFCM algorithm to reduce the total data transmission required without significantly affecting vital information in the data stream. This algorithm can be applied to early forest fire detection and monitoring applications, where sensors calculate fire weather index (FWI) based on its multi-modal sensor nodes and transmit only the potential fire index values to the central station.

2. RELATED WORK
Data stream mining is the process of extracting knowledge structures from continuous data streams. A data stream is an ordered sequence of instances that in many applications of data stream mining can be read only a small number of times using limited computing and storage capabilities. Examples of data streams include among others; computer network traffic, phone conversations, ATM transactions, web searches, and sensor data. In data mining we are interested in techniques for finding and describing structural patterns in data as a tool for helping to explain that data and make predictions from it [3]. One of the popular data mining techniques in centralized environment is

\[ {' typal values valid for MICAz motes } \]
data clustering. The general goal of clustering technique is to decompose or partition data sets into groups such that both intra-group similarity and inter-group dissimilarity are maximized [4]. For WSN environment to achieve significant energy conservation, clustering has to be performed distributed within the network due to the constraints mentioned above. There are several recent research works on distributed clustering. Density based clustering algorithm for distributed environment in [5], however it has poor response to data sets with varying densities [6]. Resource aware online distributed clustering [7&8] categorizes incoming data into existing models, hence not suitable for simultaneous distributed within the network due to the constraints. The environment to achieve significant energy group dissimilarity are maximized [4]. For WSN group such that both intra-group similarity and inter-group dissimilarity are maximized [4]. For WSN deployments. In this work, ZigBee protocol is used to organize nodes in a self-organizing group generation structure as in Figure1. ZigBee uses the IEEE 802.15.4 Low rate WPAN standard layers to build a complete protocol stack [11]. The ZigBee protocol stack maintains the nodes topology and group organization. Hence, the focus of this research is on the data stream clustering problem rather than that of organizing sensor nodes in groups. Building on this underlying protocol stack we develop an energy-efficient distributed data stream clustering algorithm (SUBFCM) for WSNs.

3. WSN ARCHITECTURE

Multi-tiered networks are known to enhance network scalability and energy efficiency for large scale deployments. In this work, ZigBee protocol is used to organize nodes in a self-organizing group generation structure as in Figure1. ZigBee uses the IEEE 802.15.4 Low rate WPAN standard layers to build a complete protocol stack [11]. The ZigBee protocol stack maintains the nodes topology and group organization. Hence, the focus of this research is on the data stream clustering problem rather than that of organizing sensor nodes in groups. Building on this underlying protocol stack we develop an energy-efficient distributed data stream clustering algorithm (SUBFCM) for WSNs.

4. THE SUBFCM ALGORITHM

Fuzzy C-Means (FCM) algorithm is the most widely used clustering algorithm in the field of data mining. However, FCM requires prior information of how many clusters C to partition the data space into. Unfortunately the number of clusters C within the WSN datasets is not pre-known. Hence in this research, Subtractive clustering and FCM algorithms are blended to implement an algorithm that does not require prior information of number of clusters in the data space. The proposed algorithm is called Subtractive Fuzzy Cluster Means (SUBFCM). The SUBFCM algorithm uses a subtractive clustering approach to determine the number of cluster prototypes C and the prototype centres c. The algorithm then partitions the stream into C fuzzy clusters using the prototype centres from the above step as initial fuzzy cluster centres. Initially, the SUBFCM algorithm assumes each D-dimensional data point \( u_i \), \( i = 1, 2, 3, ..., m \) as a potential cluster centre with a measure of potential of data points in the stream as:

\[
\text{pot}_i = \sum_{j=1}^{m} e^{-\alpha \left| u_i - u_j \right|}
\]

Where \( \alpha = 4/r_a^2 \), and \( r_a \) is a positive constant defining cluster radius. The measure of potential for a given data point is a function of its distances to all other points. A data point with many neighbouring points will have a high potential value. After computing the potential for every point, the point \( u_i \) with the highest potential \( \text{pot}_i \) will be selected as the first cluster centre \( c_1 \). The potential for every other point is then updated by (2):

\[
\text{pot}_i = \text{pot}_i - \text{pot}_i e^{-\beta \left| u_i - u_j \right|}
\]

where \( \beta = 4/r_b^2 \), and \( r_b > r_a \) as a constant. Following the update process, the data point with the highest remaining potential is selected as the next cluster centre \( c_2 \), and the process repeats until a given threshold \( \varepsilon \) for the potential is reached and \( C \) such centres are computed. SUBFCM then uses clustering criterion of squared distance \( d_{ci}^2 \) between the \( i-th \) stream sample and the \( c-th \) prototype and define the objective function as:

\[
J_{\vartheta} = \sum_{c=1}^{C} \sum_{i=1}^{m} v_i \vartheta d_{ci}^2
\]
Where the squared distance function is given as:

$$d_{ci}^2 = \|u_i - c_i\|^2$$

(4)

where $v_{ci}$ represents the membership degree of the $i$-th stream sample to the $c$-th cluster.

Membership is determined under the conditions:

$$v_{ci} \in [0,1], \quad c = 1,2,3,\ldots,C, \quad i = 1,2,3,\ldots,m$$

(5)

$$\sum_{c=1}^{C} v_{ci} = 1, \quad i = 1,2,3,\ldots,m$$

(6)

where $\theta$ is a weighing exponent or fuzziness measure. If $\theta = 1$ the clustering model is reduced to the hard K-means model. The larger the $\theta$, the fuzzier the memberships. $\theta$ is usually set to 2 [12].

The stream partitioning takes place by optimizing the criterion function (3) through iteration updating the cluster prototype centres $c_j$ and the membership function $v_{ci}$ as (7) and (8) respectively:

$$c_j = \frac{\sum_{i=1}^{m} v_{ci}^{(l)} y_{i}}{\sum_{i=1}^{m} v_{ci}^{(l)}}$$

(7)

$$v_{ci} = \left(\sum_{c=1}^{C} \left[\frac{\|u_i - c_i\|^2}{\|u_i - c_j\|^2}\right]^{\frac{2}{m-1}}\right)^{-1}$$

(8)

The iteration should stop when:

$$\max = \left\{ v_{ci}^{(l+1)} - v_{ci}^{(l)} \right\} < \xi$$

(9)

where $\xi$ is the termination criterion. $0 < \xi < 1$ and $l$ is the iteration step.

SUBFCM takes the fuzzy radius $r_a$ and fuzziness measure $\theta$ as inputs and reveals the structures in the data stream space. The parameter $r_a$ determines the granularity of the structures. The smaller it is the higher the resolution of the structures and the more computation overhead. The SUBFCM Algorithm steps are shown in Table 1 below.

### 5.1 Local sensor nodes’ task

Local sensor nodes capture the environmental observations at a specified sampling rate for each onboard sensor. The observations are averaged over a specific number of readings suitable for the sampling rate chosen. Finally each sensor node transmits the data objects to its group head periodically.

Let a sensor node $s_{nl}$ be assigned a task of capturing $n$ readings of each onboard sensor $s_1$ and $s_2$, average each reading and transmit its data object each period $t$.

So, after each period $t$, each local sensor node $s_{nl}$ forms a data object packet by combining the average readings and transmit to its group head.

### Table 1: The SUBFCM algorithm

**Subtractive fuzzy cluster means algorithm (SUBFCM)**

**Step0:** Specify fuzzy radius $r_a$, & fuzziness measure $\theta$.

**Step1:** For each data object $y_i$:

1.1 Calculate potential measure -eq (1).

1.2 Choose max $(pot_j)$ as first cluster Centre $c_j$.

1.3 Revise the potential measures –eq (2).

1.4 If $max pot_i > \epsilon$, $j=j+1$, go to 1.2

Else end; Set $C = j$;

**Step2:** Calculate $d_{ci}$ -eq(4) and Initialize

$v_{ci} = d_{ci}$; Set iteration no. to $l$.

**Step3:** Increment $l (l = l + 1)$;

3.1 Calculate Cluster centre $c_j$ -eq (7).

Set centres to $c_{(l+1)} = c_j$.

3.2 Calculate membership $v_{ci}$ -eq (8).

Set membership to $v_{ci(l+1)} = v_{ci}$.

**Step4:** If $v_{ci(l+i)} - v_{ci(l)} > \xi$

go to step3

Else end; Output $[c_{c1}], [c_{c2}]$;

### 5.2 Group heads’ task

Group heads receive data objects from each member at the specified period and save the data objects into their buffer for clustering using SUBFCM algorithm. Upon executing the SUBFCM algorithm the group heads obtain the local structures within their regions. Successively each period, they have to transmit only the structures found at their local regions. Figure 2 shows the local clusters extracted from 200 data objects by the group head nodes.
5.3 Sink’s task

The sink computes global structures based on the local structures received from the group heads and triggers any defined alarms if alarm conditions are discovered. The global structures extracted at the sink can also be used to model the observed sets of phenomena. Figure 3 shows the global clusters extracted at the sink using the local clusters of Figure 2.

6. RESULTS AND DISCUSSION

6.1 Energy Consumption Analysis

For the purpose of energy consumption analysis we utilized the radio power model of [13] which is valid for MICAz motes. A simplified Microcontroller Unit (MCU) model of [14] is also utilized to compute the clustering algorithms’ power consumption. For FCM clustering algorithm case the sensor nodes are placed at a distance \( d \) from the sink which is also the distance Group Heads (GH) are placed during distributed algorithm. For the sake of simplicity, the FCM algorithm’s power consumption is assumed proportional to the number of iterations needed for the algorithm to converge. The subtractive clustering algorithm’s power consumption is assumed proportional to the number of cluster centres found. Fuzzy logic toolbox of MATLAB 7b was used for generating the results.

6.1.1 FCM and K-means algorithms energy Consumption

The total energy requirement of FCM and K-means algorithms to discover \( n \) structures from one scan of \( m \) sensor nodes is the sum of transmitting energy \((T_x = E_c * b + A_e * b + d^2)\), receiving energy \((R_x = E_c * b)\), and algorithm execution energy \((P_{al} = (h + g) C_{avg} V^2)\). Hence Total energy consumption \( E_{fcm} \) is:

\[
E_{fcm} = b((E_c + A_e d^2) + E_c) + (h + g) C_{avg} V^2 \quad (10)
\]

where \( T_x/R_x \) is transmitting/receiving energy, \( E_c \) is transceiver circuitry energy (50pj/bit), \( A_e \) is the transmitter amplifier energy (100pj/bit/m²), \( p_c \) is FCM/ K-means processing energy, \( d \) is sensors to sink distance, \( b \) is the data bit length (1 byte long for generality), \( h \) is number of cluster centres (FCM/K-means), \( g \) is the number of iteration FCM/K-means, \( C_{avg} \) is average capacitance per cycle (MCU datasheet), and \( V \) is the power supply voltage.

6.1.2 Distributed SUBFCM algorithm energy Consumption

The total energy requirement of SUBFCM algorithm \( E_{dist} \) to discover \( n \) structures from a single scan of \( m \) sensor nodes is the sum of energies required for local transmission \((T_{xlocal} = E_c * b + A_e * b + d^2)\), group head (GH) reception \((R_{xgh} = E_c * b)\), GH processing \((P_{gh} = (h1 + g1) C_{avg} V^2)\), GH transmission \((T_{xgh} = E_c * b + A_e * b + d^2)\), sink reception \((R_{xsink} = E_c * b)\) and sink processing \((P_{sink} = (h2 + g2) C_{avg} V^2)\).

Therefore total energy consumption of SUBFCM is:

\[
E_{dist} = b[E_c(A(d_1^2 + qd_2^2) + q) + (h + g) C_{avg} V^2 \quad (11)
\]

where \( d_1 \) is the intra-cluster distance, \( d_2 \) is group head to sink distance, \( h_1 \) no. of cluster centres at group heads, \( h_2 \) is no. of cluster centres at sink, \( g_1 \) is no. of iterations at group head (SUBFCM), \( g_2 \) is no. of iterations at sink (SUBFCM) is the ratio of no. of data bits after clustering to that of before clustering (compression ratio), and \( E_c, A, C_{avg} \) and \( V \) are as defined in section 6.1.1.
6.2 Experimental Evaluation

We evaluate the performance of distributed SUBFCM algorithm in comparison with FCM and K-means algorithms. We used real Temperature and Relative Humidity data for the purpose of this evaluation. To evaluate the energy efficiency of our distributed clustering algorithm against the FCM algorithm, the same sets of data are used in each case. In the case of FCM and K-means, the sensor readings (temperature & Relative Humidity) were directly transmitted to the sink. The energy consumption results of FCM algorithm and K-means algorithm match exactly and their plots overlap. Therefore in all the experimental results only the FCM energy consumption plots are shown to represent both.

For the distributed SUBFCM algorithm case, the 200 sensor nodes are organized in 10 groups of 20 sensors each. Each group falls within the influence of a single group head which collects the reading of all its members, process and send to the sink. Each group head in turn falls under the influence of a single sink. Hence, the distances from the sensors to their group head (intra-group distance) and the distances from the group heads to the sink are limited by the radio range of used platform, which is 120m for Chipcon’s CC2431 radios used here.

The intra-group distances are fixed at 30m for all simulations after several experiments reveal that decreasing below this value doesn’t have any significant effect for the radio model used. Longer intra-group distances also prove un-optimal.

Energy consumption simulations reveal that the consumption of FCM and K-means algorithms increases as the transmission distances increases. As shown in Figure 4, for multi-hop case of FCM and K-means algorithms, the energy consumption follows similar pattern as single hop except for sharp increases at each hop due to the extra receiving energy involved. In Figure 4, first hop transmits 50m and the second hop transmits to the sink. For SUBFCM, the energy consumption also increases continuously but smoother with transmission distance. Figure 4 shows that SUBFCM consumed 33.88% of the energy consumed by FCM and K-means to transmit structures in a 400byte dataset the same distance of 120m.

Figure 5 describes the energy consumption of single hop FCM and K-means algorithms in comparison to the SUBFCM algorithm for various compression ratios. It is clear that SUBFCM’s energy consumption decreases as the compression ratio increases. For compression ratio less than 0.2, SUBFCM consumes slightly more energy due to the computation over head. It is clear that the clusters accuracy decreases as the compression ratio increases. Therefore, the compression ratio should be set according to the accuracy needed.

We also evaluate the clustering accuracy of SUBFCM algorithm with respect to FCM and K-means algorithms. The cluster centres calculated by our WSN using distributed SUBFCM algorithm is slightly displaced compared to the cluster centres calculated by standard FCM and K-means algorithms for the same sets of data. The SUBFCM errors are within 0.1⁰C to 1.5⁰C with an average error of 0.51⁰C and 0% to 7% with average error of 2.21% for temperature and Relative humidity clusters respectively compared to FCM shown in Figure 6 for compression ratio of 0.7. Compared to K-means the errors are within 0⁰C to 2.6⁰C with average 0.78⁰C and 0% to 22% with average 6.36% for temperature and relative humidity clusters respectively as shown in Figure 7 for compression ratio of 0.7. The higher errors compared to the K-means can be explained by the dependency of the K-means algorithm on initial random centres.
7. CONCLUSION

In WSNs, it is energy efficient to process the acquired data within the network and transmit the results rather than transmitting raw data. Distributed SUBFCM algorithm clusters the acquired data streams within the network and transmits the local clusters to the sink. SUBFCM doesn’t require the number of centres to be pre-specified. Given the required accuracy and fuzziness measure it discovers multidimensional clusters in an energy efficient way. Simulations reveal that SUBFCM algorithm can partition multidimensional data sets in to clusters at less than 40% of the energy cost of Fuzzy C-means and K-means algorithms. For a specified compression ratio of about 0.7 it produces clusters with bounded average errors.

8. REFERENCES


