Energy-Efficient Routing Scheme for Distributed Regression in Wireless Sensor Network

Chaojun Hou and Guoli Wang∗
School of Information Science & Technology
Sun Yat-Sen University
Guangzhou 510275, China PR
∗isswgl@mail.sysu.edu.cn

Abstract: This paper presents an energy-efficient routing scheme for the in-network implementation of data regression modelling in wireless sensor networks (WSNs). With the kernel based distributed representation, a clustering based routing architecture is proposed to implement the distributed Gaussian elimination for solving the regression algebraic equation. In particular, the skeleton routing tree is built to coordinate the local computations associated with the clusters. The rest ramose nodes within each cluster are limited to contribute the corresponding local computations. Compared with the junction tree formation over all nodes, the notion of skeleton tree (ST) proposed here suppresses the redundant message passing among the ramose nodes with each cluster without deteriorating data modelling performance. Experimental results are reported to demonstrate the effectiveness of the proposed approach.

Key Words: Wireless Sensor Networks, Routing, Distributed Computation, Data Acquisition, Regression

1 INTRODUCTION

Data acquisition is one of the fundamental ingredients in WSNs, which plays a prominent role in many practical applications, such as environmental monitoring, military installations and scientific research[1,2]. Due to the limitations on node’s energy and bandwidth, it is expensive and inefficient to gather the entire correlated raw data from the sensors through the network and analyze offline. Thus online in-network data acquisition arouses much research interest on leveraging spatial-temporal correlations in measured data. Among the existing studies, data modelling is of special interest in acquiring much more complete information from sensor data[10], especially, when compared with simple data aggregation schemes[3, 4, 5]. Along this line, the huge quantities of observed data can be compactly represented in lower dimensional spaces without losing the shape and structure of the data. That is, data modelling can decrease the communication requirements significantly while maintaining the sensor field structure well.

Distributed regression, as a nonparametric model, offers a more general framework of data modelling in WSNs[8], thus is more flexible than parametric models such as the mixture of Gaussians built by the distributed EM algorithm[6], as well as the distributed hidden Markov graphical model by the embedded polygons iterative algorithm[7]. Except the cases of small homogeneous networks, for example, considered in [9], the key to the in-network implementation of data regression is partitioning a network into numerous small regions for distributed processing, so its associated local regression model can be properly built when the interactions with its neighbor regions are decoupled by the intersection message passing[10]. To this end, the spanning tree, a commonly used routing scheme, need to be reorganized so that the running intersection property is satisfied. The junction tree developed for solving distributed inference problems is an elegant formation to enforce the running intersection property. An asynchronous message passing protocol with the junction tree has been developed, in which all nodes assemble themselves into network junction tree. This routing architecture inevitably results in redundant communication requirements for those nodes with the same associated clique and set of factors. In particular, it is unnecessary for all nodes to solve the regression problem, which motivates this study to set up the hierarchical routing structure so as to eliminate the redundant message passings.

This study proposes a clustering based routing architecture to implement the distributed Gaussian elimination for solving the regression algebraic equation. In particular, the skeleton routing tree is built to enforce the running intersection property, on which the intersection message passings can be properly carried for decoupling the interactions among the neighbor regions. The rest ramose nodes are limited to contribute the corresponding local computations with one-way message passings. Compared with the junction tree formation over all nodes, the notion of skeleton tree proposed here suppresses the redundant message passing among the ramose nodes with each cluster without deteriorating data modelling performance. Experimental results are reported to demonstrate the effectiveness of the proposed approach.

This work is supported by National Nature Science Foundation under Grant 60473120 and Guangdong Nature Science Foundation under Grant 06023190. The second author acknowledges the support of the SRF for ROCS, SEM of China.
2 DISTRIBUTED REGRESSION IN WSN

Assume that we have N sensor nodes; each node \( N_i \) located at \( x_i = (x_i, y_i) \) can measure their physical environment, such as temperature, humidity or light. There are significant amount of redundancy in measurement readings from a sensor over time, and between different adjacent nodes[10].

We view the \( f(x, t) \) as an unknown true sensor field function defined in temporal and spatial domain. We intend to find a good fitting function \( \hat{f}(x, t) \) to extract the characteristic of the \( f(x, t) \) from the measurement data set \( S = \{ (x_i, t_k) \}_{i=1}^{N} \) in the time window \( T \). As the time window \( T \) is sliding, the \( \hat{f}(x, t) \) can update along with the approach of the new sample data.

2.1 Distributed Regression Model

In the centralized mode, the fitting function \( \hat{f} \) can be found in the least square sense by a given set of basis functions \( H = \{ h_k \}_{k=1}^{m} \). We would like to find the weight coefficients \( w = \{ w_k \}_{k=1}^{m} \) such that \( \hat{f}(x, t) = \sum_{k=1}^{m} w_k h_k(x, t) \), and pick the weight vector \( w^* \) as

\[
w^* = \arg \min_{w} \frac{1}{s} \sum_{i=1}^{s} \sum_{t_j \in T} \left( \hat{f}(x_i, t_j) - \hat{f}(x_i, t_j) \right)^2,
\]

where \( s = |S| \), the size of whole sample set \( S \).

In what follows, we assume that the giant network is decomposed into a set of overlapped regions, according to the principle that nodes in the same region are highly correlated. Denote the network cover \( U = \bigcup_{j=1}^{L} R_j \), where \( R_j \) is corresponding to region \( j \) and \( L \) is the number of regions. Region \( R_j \) can be defined by a support of kernel function \( \kappa_j(x) \), i.e., when \( x \in R_j \) iff \( \kappa_j(x) > 0 \). Further, let \( \sum_{j=1}^{L} \kappa_j(x) = 1, \forall x \in U \).

So the function \( \hat{f}(x, t) \) in the form of

\[
\hat{f}(x, t) = \sum_{j=1}^{L} \kappa_j(x) \sum_{h_l \in H_j} u_l h_l(x, t),
\]

where \( R_j \) is associated with a set of basis function set \( H_j \) with size \( d_j = |H_j| \). The estimation of \( f \) at \( x \) and \( t \) is in the form of the weighted linear combination of the spatial-temporal kernels and the basis functions \( \kappa_j(·) h_l(·, ·) \).

Denote \( M \) as the total number of all kernels. Define the data dot-product by

\[
(g, h) = \sum_{i,j} g(x_i, t_j) h(x_i, t_j)
\]

where summarize all \( (x_i, t_j) \) in \( S \). Every node \( N_i \) define the native index set \( L_i = \{ l_j | x_i \in \text{supp}(\kappa_j), j = 1, \ldots, L \} \). Abbreviate \( \text{supp}(i, j) = \text{supp}(\kappa_i(·)) \cap \text{supp}(\kappa_j(·)) \) as the support of the intersection between kernel functions \( \kappa_i \) and \( \kappa_j \).

The solution to (2) is in the least square form

\[
w^* = \arg \min_{w} \| Hw - f \|^2 = (H^T H)^{-1} H^T f,
\]

where \( f \) is a \( s \times 1 \) vector by arranging all sample data from \( S \) into a column, and \( H \) is a \( s \times M \) matrix. Each row vector in \( H \) is associated with one pair \( (x_i, t_k) \) from \( S \), which are arranged in \( (\kappa_i(x_k) h_j^l(x_k, t_k))_{i=1,L, j=1,d_j} \). We denote

\[
A = H^T H = (A_{i,j})_{i,j=1,L},
\]

\[
b = H^T f = (b_i)_{i=1,L},
\]

where \( A \) is a \( M \times M \) symmetry data dot-product matrix, and can be viewed as composed of \( L \times L \) block matrices and each block matrix, e.g., \( A_{i,j} \), is \( d_i \times d_j \), whose entry \( (A_{i,j})_{u,v} = \langle \kappa_i h_u^i, \kappa_j h_v^j \rangle \), where \( u = 1, \ldots, d_i \), and \( v = 1, \ldots, d_j \). Clearly \( A_{i,j} = A_{j,i}^T \).

2.2 In-network Message Passings

In [10, 11], Junction Tree (JT) structure is employed to solve the sparse linear system \( \text{Aw} = \text{b} \) in a distributed fashion, where messages asynchronously exchange between neighboring nodes to perform the Gaussian elimination. To do so, it necessarily augments the notion of \( L_i \) to \( C_i \), denoted as the label set, to satisfy the running intersection property in routing tree.

Now, suppose that \( N_i \) and \( N_j \) are the immediate neighbors of each other in routing tree, with the ordered label sets \( C_i \) and \( C_j \) respectively. Accordingly, we can collect the relevant terms to form the local distributed representations of \( A \) and \( b \): \( (A^{(i)}, b^{(i)}) \) and \( (A^{(j)}, b^{(j)}) \). Assume \( U, V \) is an ordered index subset of \( \{1, 2, \ldots, L\} \). Let \( A_{U,V} \) denote a partial block matrix of \( A \), whose entries are corresponding to the rows and columns in \( A \) indexed by \( U \) and \( V \) respectively. In the same way, we can define the partial vector \( b_U \).

Let \( S = C_i \cap C_j \), \( V = C_i \setminus S \), and \( N(i) \) is the set of \( N_i \)’s all neighbors. The message \( M_{i,j} \) sent to \( N_j \) from \( N_i \) is given by

\[
A^{(i,j)} = A_{S,S} - A_{S,V} (A_{V,V})^{-1} A_{V,S},
\]

\[
b^{(i,j)} = b_S - A_{S,V} (A_{V,V})^{-1} b_V,
\]

with \( A^{(i)} = A^{(i)} + \sum_{k \in N(i) \setminus \{ i \}} A^{(k,i)} \).

\[
b^{(i)} = b^{(i)} + \sum_{k \in N(i) \setminus \{ i \}} b^{(k,i)}.
\]

Clearly, \( M_{i,j} \) can be determined from both \( N_i \)’s local representation and all inward message received from \( N_j \’s \) neighbors except \( N_j \). The communication message between \( N_i \) and \( N_j \) is represented by \( C_i \cap C_j \). Indeed, a perfect Gaussian elimination is implemented for the sparse matrix \( A \).

We can see that the correlations among neighboring regions will be decoupled by the intersection messages. So far, the entire optimal solution \( w \) to (3) is locally solved by all node, and can be constituted from some nodes’ \( w_{C_j} \).
3 CLUSTERING BASED ROUTING SCHEME

Our message passing schedule is developed based on skeleton tree to structure the intersection message routing, which facilitates one-way inter-cluster communication feasible thus reduce the redundant communication requirements.

3.1 Skeleton Tree Structure Via Clustering

As mentioned before, since the nodes with the same label set will solve the same local coefficient set, it is unnecessary for each node to propagate the full bi-directional messages from or to its neighbors. For example, in Fig.1, the left most three nodes in region I perform the redundant message passing in bi-directional way since the anyone of them can solve $w_{ij}$. In addition, it is more robust yet easy to maintain local routing tree topology rather than the entire network. This is the motivation to introduce the skeleton tree structure.

More generally, the entire network is partitioned into a set of smaller non-overlapped clusters, denoted by $E_k$. That is,

$$U = \bigcup_{j=1}^{L} R_j = \bigcup_{k=1}^{K} E_k,$$

where $K$ is the total number of the clusters. Here, the cluster $E_k$ is defined by a set of connected nodes with the same native set $L_i$. Note that the clusters are mutually exclusive. In the other words, $E_k \cap E_j = \emptyset$ if $k \neq j$. Thus $N_{b_i}$ only belongs to one cluster.

It is followed from the notion of clustering mentioned above that the nodes in the same cluster have the same $L_i$ with the same partial set of the weight coefficients. It can be seen that it is easy to maintain each local routing tree due to a smaller size. Moreover, it weakens the effect of the neighboring clusters’ inner failures. In short, it is robust and scalable to manage the entire network topology through maintaining numerous self-organized clusters, where each cluster just need to known the boundary nodes of other neighboring clusters. In contrast, the JT structure is costly to maintain the whole routing tree within the entire network.

In practice, there is no need to run actual algorithm to build up the clusters, every node just requires to know which neighbors are within the same cluster and which are not, which can be easily compared its own native set $L$ with its neighbors’.

In practice, we can view the skeleton tree consisting of ST nodes as the shortest path among the clusters. With this in mind, the nodes within each cluster can be categorized into two groups: one is ST node, the other is ramose node.

Suppose that a spanning tree is available such that adjacent nodes have high quality communication links. In addition, for each nodes, the native set $L_i$ of its neighbors is known. At this stage, the skeleton tree can be built by the following steps:

1) Obtaining the boundary nodes:

The node, who finds its neighbor in another cluster, is denoted as the boundary node in its cluster. Notice that the boundary node is also a ST node.

2) Constructing the local skeleton tree:

The local skeleton tree can be constructed through finding the shortest routing path among the boundary nodes within the same cluster. The local skeleton tree discovering phase is launched by the boundary node, shown in Fig.2. Each node $N_i$ maintains a simple route table with the entry pair $(src, from)$, where $N_{src}$ is associated with one boundary node and $N_{from}$ is the outward node that routes to $N_{src}$.

Firstly, all of the boundary nodes, $N_{b_i}$, broadcasts message $(ID_{b_i}, ID_{b_j})$. After receiving the message $(src, j)$ from neighbor $N_j$, $N_i$ adds the entry $(src, j)$ to its $List_i$ and re-broadcasts the new message $(src, i)$ to its neighbors if all of the following conditions are met: (a) $N_j$ is one of the routing tree neighbor of $N_i$; (b) $L_i$ is the same as $L_j$; (c) $N_i$ is not a boundary node; (d) there is not a entry $(src, *)$ or $(*, j)$ in $List_i$. Note that the node, satisfying conditions above, must be an inter-node in cluster, and the message from $N_j$ is not duplicated in the entry pair of either srcID or fromID, so the node needs to re-broadcast the new message $(src, i)$.

Furthermore, the inner-node $N_i$ is set as a ST node if $|List_i| > 1$, for $N_i$ has at least two distinct entries in $List_i$, e.g. $(b_1, k)$ and $(b_2, l)$, this implies that $N_i$ is on the shortest path in the routing tree between the boundary node $N_{b_1}$ and $N_{b_2}$ from $N_{b_1}$ and $N_{b_2}$ respectively.

3) Constructing the whole Skeleton Tree:
SkeletonTreeSetup (i)
On event: Initialize( )
1: \( L_i \leftarrow \text{The native index set of node } N_i \).
2: \( N(i) \leftarrow \text{Neighbors of } N_i \) in routing tree.
3: \( NN(i) \leftarrow \emptyset \), \( \text{List}_i \leftarrow \emptyset \).
4: \( \text{nodetype}_i \leftarrow 0 \), \( \text{border}_i \leftarrow 0 \).
5: for each \( j \in N(i) \) do
6: \( L_j \leftarrow \text{Label set of node } N_j \).
7: if \( L_j \neq L_i \) then
8: \( \text{border}_i \leftarrow 1 \). // indicate to boundary node
9: \( \text{nodetype}_i \leftarrow 1 \). // indicate to ST node
10: end if
11: end for
12: if \( \text{border}_i = 1 \) then
13: Broadcast message(\((i, i)\)).
14: else
15: Wait for message received.
16: end if

On event: ReceiveMessage(\((src, j)\)):
17: if \( j \notin N(i) \) or \( C_j \neq C_i \) or \( \text{border}_i = 1 \) then
18: Exit and discard the message.
19: end if
20: \( \exists (\text{srcID, fromID}) \in \text{List}_i \) such that
21: \( \text{srcID} = src \) or \( \text{fromID} = j \) then
22: Exit and discard the message.
23: \( \text{List}_i \leftarrow \text{List}_i \cup (src, j) \).
24: Broadcast message(\((src, j)\)).

  // set node type and STC neighbors.
25: \( NN(i) \leftarrow NN(i) \cup j \).
26: if |\( \text{List}_i \)| > 1 and \( \text{border}_i = 0 \) then
27: \( \text{nodetype}_i \leftarrow 1 \). // indicate to a ST node
28: end if
29: end if

Figure 3: Local skeleton tree build-up algorithm.

The entire skeleton tree is made up by connecting all local skeleton trees through the adjacent boundary nodes in different clusters in the network. The other nodes, which is not the ST nodes, are denoted as ramose nodes. The ramose node \( N_k \) definitively has \((src, k)\) in \( \text{List}_v \), so that the \( N_k \) is its parent node and each ST node can be regarded as a root at a subtree of the ramose nodes.

The STC algorithm is summarized in Fig.3. It should be noted that when the algorithm completed, for ST nodes, \( NN(i) \) is the set of its ST neighbors, while for ramose node, the only one node in \( NN(i) \) is its parent node. Thus the \( NN(i) \) is used as the neighbors for handling with message passing.

4 EXPERIMENTAL RESULTS

Simulation study is presented here to illustrate the validity of the STC based distributed regression algorithm. In particular, we focus on the communication-efficiency as well as the convergence speed with respect to the JT structure in pure asynchronous mode. Here we use the raw sensed data set from the Berkeley lab deployment provided by Dr. Guestrin[10].

![Figure 4: A Map of WSN deployment: the placement of 46 sensors(dark circles). The rectangles indicate the support regions of the kernels used in the simulation.](image)

We use the one-day’s measurement data of 46 nodes out of the 54 nodes for simplicity. Firstly, we implement our algorithm on the data set of 30-second samples of light, temperature, and humidity. Because of network collision or other reasons, there are so many measurements missed irregularly, but the readings is still enough for our simulation study. In particular, the total readings among 46 nodes in the sliding window with 200min is more than 10,000, on average, more than the readings within a minute per node.

We have 5 kernels associated with the five overlapped regions in Fig.4. In this setting, every sensor at least locates in one region, while some sensors lies in two or three regions. As done in[10], we have four choices of basis function sets per kernel:

case 1 quadratic-space, quadratic-time, e.g.
\[
\hat{f}(x, y, t) = w_1 x + w_2 y + w_3 xy + w_4 + w_5 t + w_6 t^2
\]

(7)
case 2 linear-space, quadratic-time with five functions
case 3 linear-space, linear-time with four;
case 4 linear space, constant-time with three.

We set the weight kernel function
\[
\kappa_j (x) = \frac{K_j (x)}{\sum_{v=1}^{L} K_v (x)}
\]
where \( K_j (x) \) can be simply defined as \( 1/M(x) \), the number of regions which \( x \) belongs to. Alternatively, \( K_j (x) \) can be defined as
\[
K_j (x) = \min (|x_j - x_1|, |x_j - x_2|) + \min (|y_j - y_1|, |y_j - y_2|)
\]
where the \( j \)th region is a rectangle demarcated by the left bottom point \((x_{j,1}, y_{j,1})\) and the right top point \((x_{j,2}, y_{j,2})\). We prefer to use the later form of \( K_j (x) \) to \( \kappa_j (x) \) in our simulation.

Suppose that all of the nodes triggers the event of message passing at the same time span of interval \( T_C \), but in different starting time. Either in JT or STC scheme, each node only sends one message to its neighbors at every interval \( T_C \). The size of the message, e.g. \( M_{i \rightarrow j} \) from \( N_i \) to \( N_j \), is \( n(n+1)/2 + n \), which is associated with the \( n \times n \) symmetric matrix \( A^{i,j} \) and \( n \times 1 \) vector \( b^{i,j} \), where \( n \) denotes the
number of the basis functions in $C_i \cap C_j$. Fig. 3 illustrates
the communication overhead of the total message passing at
one time interval from all nodes. It can be seen that the
nodes operate in STC scheme can save about 33% com-
mutation cost in JT scheme. The saving percentage of
the communication cost is dependent on the practical rout-
ing tree topology, but is insensitive to the number of basis
functions. In particular, using the basis function set of (7),
the total communication cost in STC scheme at
$T_{C}$ is 8568 bytes, 2142 multiplying 4 bytes, while 12924 bytes with
3231 multiplying 4 bytes in JT scheme.

5 Conclusion And Future Work

This paper presents the notion of the skeleton tree, a new
paradigm for combining the clustering based routing ar-
chitecture with the message passing protocol based on the
junction tree, in the context of data regression modeling in
WSN. Compared with the junction tree formation over all
nodes, the skeleton tree based routing scheme suppresses
the redundant message passing among the ramose nodes
with each cluster without deteriorating data modeling per-
formance. Experimental results are reported to demon-
strate the effectiveness of the proposed approach. In par-
ticular, the routing scheme proposed here can save about
33% communication cost.

The temperature in a day often varies at different trend dur-
ing each time span. Thus it is interesting to find an adaptive
mechanism for adjusting the system parameters, such as the
length of $T_C$, the scope of region, etc., to compromise the
performance required and the communication cost. For ex-
ample, as the temperature varies slowly, we can enlarge the
length of $T_C$ to guarantee the performance requirements
while maintaining the communication cost at an acceptable
level.

Acknowledgment

The authors would like to sincerely thank Dr. Guestrin sup-
port the environmental sensory data for our simulation.

REFERENCES

[1] A. Mainwaring et al, D. Culler, J. Polastre, R. Szewczyk, and
J. Anderson. Wireless sensor networks for habitat monitor-
ing, in Proceedings of the 1st ACM international workshop
on Wireless sensor networks and applications Atlanta, Geor-

B. Stephen, D. Todd, B. Phil, G. David, and H. Wei, A macro-
scope in the redwoods, in Proceedings of the 3rd interna-
tional conference on Embedded networked sensor systems
San Diego, California, USA: ACM, 2005.

[3] S. Madden, R. Szewczyk, M. Franklin, and D. Culler, Sup-
porting aggregate queries over ad-hoc wireless sensor net-
works, In Workshop on Mobile Computing and Systems Ap-
plications, 2002.

[4] M. A. Sharaf, J. Beaver, A. Labrinidis, and P. K. Chrysanthis,
Balancing energy efficiency and quality of aggregate data in
2004.

average: Towards sophisticated sensing with queries, In In-
f ormation Processing in Sensor Networks, Berkeley, April
2004.