Retrieving the Relative Kernel Dataset from Big Sensory Data for Continuous Query

Tongxin Zhu[1], Jinbao Wang[2], Siyao Cheng[1], Yingshu Li[3], and Jianzhong Li[1]

[1] School of Computer Science and Tech, Harbin Institute of Technology, Harbin, China
{zhutongxinhit}@126.com, {csy,lijzh}@hit.edu.cn
[2] The Academy of Fundamental and Interdisciplinary Sciences, Harbin Institute of Technology, Harbin, China
{wangjinbao}@hit.edu.cn
[3] Department of Computer Science, Georgia State University, Atlanta, GA, USA
{yili@gsu.edu}

Abstract. With the rapid development of Wireless Sensor Networks (WSNs), the amount of sensory data manifests an explosive growth. Currently, the sensory data generated by some WSNs is more than terabytes or petabytes, which has already exceeded the computation and transmission abilities of a WSN. Fortunately, the volume of valuable data for a given query is usually small. For a given query Q, the dataset which is highly related to it is called the relative kernel dataset $\mathcal{K}^Q$ of Q. In this paper, we study the problem of retrieving relative kernel dataset from big sensory data for continuous queries. The theoretical analysis and simulation results show that our proposed algorithms have high performance in term of accuracy and resource consumption.

Keywords: Wireless sensor networks · Big sensory data · Relative kernel dataset.

1 Introduction

The Wireless Sensor Networks (WSNs) provide an efficient way to observe the complicated physical world. Benefiting from the wireless telecommunications, embedded systems and sensing techniques, the WSNs have rapidly developed and are widely used. According to the annual report of Gartner, 20.8 billion connected things will be in use worldwide in 2020 [1]. Besides, it is estimated that more than 250 things will connect each second by 2020, and more than 50 billion things will be connected to the Internet by 2020 [2]. Such large scale of WSNs causes the explosive growth of sensory data.

As an important instance of Big Data, Big Sensory Data also has four V's characters. (1) **Volume**. The amount of sensory data in some WSNs has already exceeded terabyte or petabyte [1, 2], so that the volume of Big Sensory Data is extremely huge. (2) **Variety**. As the applications of WSNs are becoming more and more complex, different types of sensors are applied in one WSN, so that

the big sensory dataset contains a huge variety of sensory data. (3) **Velocity**. In consequence of the high sampling frequency of the large amount of sensors in a WSN, the generating velocity of big sensory data is quite fast. (4) **Value**. The existence of noises and redundancies in big sensory data diminishes its quality. Although the total value brought by the big sensory data is high, the value-scale ratio is quite low.

The above properties bring many challenges for data processing in WSNs, and make the existing sensory data acquisition, routing [3–6], data collection [7–9] and data computation [10, 11] techniques no longer applicable. Taking the *Volume* property as an example, the current amount of data has already exceeded the transmission and computation ability of WSNs. Therefore, a series of new in-network processing algorithms with much lighter transmission and computation overloads should be considered.

Fortunately, the volume of valuable data for a given query Q is usually small although the volume of all sensory data in the WSN is quite huge. As the function of a WSN becomes complex, a WSN containing with n different types of sensors can support a variety of queries. Therefore, a given query Q is usually highly related to the sensory data generated by only k types of sensors, where $k < n$. The sensory data generated by that k types of sensors is denoted as the relative kernel dataset $\mathcal{K}^Q$ of Q.

Most WSNs support continuous queries since the function of the WSNs is to monitor the physical world in real time. Therefore, continuous queries are frequent in a WSN. The energy consumption of the WSNs can be reduced a lot if the energy consumption for processing the continuous queries is reduced. Processing continuous queries with the relative kernel datasets are energy efficient in both communication and computation in consequence of their quite smaller amount of sensory data compared with the raw big sensory data.

Although there exists some data reduction algorithms for reducing the amount of sensory data, they are not suitable for processing a given continuous query energy efficiently. Firstly, the simplest data reduction methods are based on sampling [12, 13]. However, the sampling methods are only suitable for simple statistic queries. For other queries, the valuable data of the given query may not be sampled due to the limitation of sampling frequency. On the other hand, the sampled data may not be necessary for the given query. Secondly, the compressed sensing technique is another classical method for in-network data reduction [14, 15]. However, the compressed sensing technique only considers the correlations between sensory data while ignoring the correlation between the sensory data and the given query. Then, the data reduction for a given query is not enough. Thirdly, to the best of our knowledge, the work proposed by Cheng et. al. is the only one that discussed the big sensory data processing problem up to now [16, 17]. However, the dominant dataset defined in [16] is completely irrelevant to the given query. That is, all queries in the WSN share one dominant dataset, which is not energy efficient enough for a given query.

Due to the above reasons, we investigate the relative kernel dataset retrieving problem for continuous queries in this paper. Two algorithms, the Relative

Kernel Dataset Retrieving (RKDR) Algorithm and the Piecewise Linear Fitting Based Relative Kernel Dataset Retrieving (PLF-RKDR) Algorithm, are proposed to solve this problem. The RKDR algorithm retrieves the relative kernel dataset for a given query by the linear correlation analysis. Then a method for answering the query by the relative kernel dataset is given. The PLF-RKDR algorithm improves the accuracy of the retrieved relative kernel dataset for a given query by applying the piecewise linear fitting to retrieve the relative kernel dataset for a given query. Besides, we also provide a method to answer the query approximately by the retrieved relative kernel dataset. The major contributions of this paper are as follows.

1. The formal definition of the relative kernel dataset for a given query is firstly proposed, considering the redundancies between different types of sensory data and the correlations between sensory data and the given query.

2. Two approximate algorithms, the RKDR algorithm and the PLF-RKDR algorithm, are proposed to retrieve the relative kernel dataset for a given query.

3. Extensive simulations on both simulation datasets and real datasets are carried out to verify the accuracies of the algorithms.

The rest of the paper is organized as follows. Section 2 provides the problem definition. Section 3 proposes the relative kernel dataset retrieving algorithm for a given query Q and provides a method to estimate the result of Q by the sensory data in that relative kernel dataset. Section 4 analyzes the performance of the proposed algorithms. Section 5 shows the simulation results. Section 6 discusses the related work. Finally, section 7 concludes the whole paper.

2 Problem Definition

2.1 The Wireless Sensor Networks Model

A wireless sensor work has n categories of sensors, which generates n types of sensory data from the monitored environment. These n types of sensory data are described as n attributes of the monitored environment, denoted as $x_1, x_2, \cdots, x_n$. The attribute set is denoted as $A = \{x_1, x_2, \cdots, x_n\}$. For each given query Q from users, the WSN returns a query result y_Q according to the sensory data. y_Q is described as the target value of query Q. Apparently, the target value y_Q is correlated with partial or all n attributes. Taking the atmospheric environmental monitoring WSN as an example, there are a variety of sensors collecting data of sulfur dioxide concentration, nitrogen dioxide concentration, *etc*, which are attributes of this WSN. When the user's query is the air quality, the WSN returns an air pollution index. In this application, the query Q is the air quality and the target value y_Q is the returned air pollution index.

2.2 The Correlation Model

A training set is applied to explore the correlations between n attributes and the given query Q. A training set contains m training examples $\{t_1, t_2, \cdots, t_m\}$.

A training example $t_j (1 \leq j \leq m)$ is presented by the values of n attributes and the corresponding target value of query Q, i.e. $\{x_{1j}, \cdots, x_{nj}, y_{Qj}\}$.

Linear correlation coefficient is a common metric for the correlation analysis, especially in WSNs. In most applications of WSNs, the sensory data can reflect the statement of the monitored physical world intuitively. That is, the simple linear correlation is usually adequate to reflect the relationship between the sensory data and the query of a WSN. Therefore, we apply linear correlation as the correlation metric in this paper. For each attribute $x_i (1 \leq i \leq n)$, the linear correlation R_i^Q between x_i and y_Q is calculated by the following formula,

$$R_i^Q = \frac{\sum_{j=1}^{m}(x_{ij} - \overline{x_i})(y_{Qj} - \overline{y_Q})}{\sqrt{\sum_{j=1}^{m}(x_{ij} - \overline{x_i})^2}\sqrt{\sum_{j=1}^{m}(y_{Qj} - \overline{y_Q})^2}} \tag{1}$$

where m is size of the training set, and x_{ij} denotes the the value of attribute x_i in the jth training example t_j, y_j denotes the target value in t_j. Besides, $\overline{x_i}$ is the average value of attribute x_i in the training set, and $\overline{y}$ is the average value of target values in the training set. R_i^Q has a value between 1 and -1, where 1 is total positive linear correlation, 0 is no linear correlation, and -1 is total negative linear correlation. That is, the greater absolute value of R_i^Q presents the stronger linear correlation between attribute x_i and query Q.

2.3 Problem Statement

In this paper, we aimed at retrieving the relative kernel dataset $\mathcal{K}^Q$ for a given query Q of a WSN. The relative kernel dataset $\mathcal{K}^Q$ is a subset of attribute set A. The definition of the relative kernel dataset $\mathcal{K}^Q$ is described as follows.

Definition 1. *(β-compatible) For any $0 < \beta < 1$, two attributes x_i and x_j is β-compatible if $|r_{ij}| \leq \beta$, where r_{ij} is the linear correlation coefficient between x_i and x_j defined as the following equation.*

$$r_{ij} = \frac{\sum_{k=1}^{m}(x_{ik} - \overline{x_i})(x_{jk} - \overline{x_j})}{\sqrt{\sum_{k=1}^{m}(x_{ik} - \overline{x_i})^2}\sqrt{\sum_{k=1}^{m}(x_{jk} - \overline{x_j})^2}} \tag{2}$$

Definition 2. *(β-compatible set) For any $0 < \beta < 1$, attribute set $S \subseteq A$ is a β-compatible set if any two attributes in S are β-compatible with each other.*

Definition 3. *(weight of set) For any attribute set $S \subseteq A$, the weight of S, denoted as $w(S)$, is defined as $w(S) = \sum_{x_i \in S} |R_i^Q|$, where R_i^Q is the correlation coefficient between attribute x_i and query Q.*

Definition 4. *((k, β)-Relative Kernel Dataset $\mathcal{K}^Q$) Given the attribute set A, query Q, the compatible parameter β, and the size of the relative kernel dataset k. The (k, β)-Relative Kernel Dataset of query Q is a subset of A, denoted as $\mathcal{K}^Q$, satisfying the following three conditions.*

(1) $\mathcal{K}^Q$ is a β-compatible subset of attribute set A,

(2) the size of $\mathcal{K}^Q$ is k, and

(3) $w(\mathcal{K}^Q) > w(S)$ for any set attribute set S satisfying the conditions (1)(2).

The sensory data in (k, β)-Relative Kernel Dataset $\mathcal{K}^Q$ is transmitted and computed each time query Q is issued by users. The smaller k indicates that less sensory data is transmitted and computed in a WSN, which saves more energy. However, the approximate result of query Q is estimated by the sensory data in $\mathcal{K}^Q$. Therefore, the smaller k will make the approximate answer less accurate.

Problem Statement. We formulate the **R**elative **K**ernel **D**ataset **R**etrieving (RKDR) problem in a WSN, as follows.

Input:

1. A query Q;
2. A training set with m training examples;
3. The required size of relative kernel dataset k;
4. The compatible parameter β.

Output:

1. The (k, β)-Relative Kernel Dataset $\mathcal{K}^Q$ of query Q;

3 Algorithms For the RKDR Problem

In this section, two algorithms, the Relative Kernel Dataset Retrieving (RKDR) Algorithm and the Piecewise Linear Fitting Based Relative Kernel Dataset Retrieving (PLF-RKDR) Algorithm, are proposed to retrieve the relative kernel dataset $\mathcal{K}^Q$ for a given query Q. We also provide two approximate methods to estimate the result of query Q by the sensory data in $\mathcal{K}^Q$.

Each attribute x_i of a WSN can be regarded as a vertex i with weight R_i^Q in a weighted graph $G(V, E)$. If attributes x_i and x_j are not β-compatible, there is an edge $(i, j) \in E$. Then, the RKDR problem can be reduced to the Weighted Maximum Independent Set (WMIS) problem. However, the WMIS problem is NP-hard. Therefore, we design heuristic algorithms for the RKDR problem.

3.1 Relative Kernel Dataset Retrieving Algorithm

The RKDR Algorithm retrieves (k, β)-Relative Kernel Dataset for a given query Q based on the linear correlation analysis. It contains the following two steps.

Step 1. Calculate the candidate relative kernel dataset $\mathcal{X}$.

At first, the candidate relative kernel dataset $\mathcal{X}$ includes all n attributes.

Step 1.1. Calculate the linear correlation coefficient R_i^Q between each attribute x_i in $\mathcal{X}$ and the target value y_Q of query Q by Equation(1).

Step 1.2. Calculate the linear correlation coefficient r_{ij} between any two attributes x_i and x_j in $\mathcal{X}$ by Equation(2). If x_i and x_j is not β-compatible (i.e. $|r_{ij}| > \beta$), indicating that attributes x_i and x_j are redundant in $\mathcal{X}$. If $|R_i^Q| < |R_j^Q|$, remove x_i from $\mathcal{X}$. Otherwise, remove x_j from $\mathcal{X}$. This step reduces the redundancy of the relative kernel dataset.

Step 2. Retrieve (k, β)-Relative Kernel Dataset $\mathcal{K}^Q$ from $\mathcal{X}$.

Sort the absolute values of the linear correlation coefficients of all attributes in $\mathcal{X}$, i.e. $\{|R_i^Q||x_i \in \mathcal{X}\}$, in descending order. The top-k coefficients are $\{|R_{a_1}^Q|, \cdots, |R_{a_k}^Q|\}$ and the corresponding k attributes are $\{x_{a_1}, \cdots, x_{a_k}\}$. Therefore, $\mathcal{K}^Q = \{x_{a_1}, \cdots, x_{a_k}\}$.

After the above two steps, the (k, β)-Relative Kernel Dataset for query Q is obtained. The detailed algorithm is shown in Algorithm 1.

Then once the query Q is issued by users again, we provide an approximate method to estimate the result of Q by the sensory data in $\mathcal{K}^Q$. For each attribute x_{a_i} in $\mathcal{K}^Q$, the linear function f_i between x_{a_i} and y_Q can be calculated by least-squares method. Then, each linear function f_i is assigned a weight according to $R^Q_{a_i}$. Therefore, $\mathcal{F}$ is an approximate function based on each linear function f_i and its weight to approximate the target value y_Q of query Q, shown as follows.

$$\mathcal{F} = \sum_{i=1}^{k} \left(\frac{|R^Q_{a_i}|}{\sum_{j=1}^{k} |R^Q_{a_j}|}\right) f_i \tag{3}$$

Algorithm 1: Relative Kernel Dataset Retrieving Algorithm

Input: query Q, a training set $\{t_1, \cdots, t_m\}$; the compatible parameter β; the required size k
Output: (k, β)-Relative Kernel Dataset $\mathcal{K}^Q$

$\mathcal{X} = \{x_1, x_2, \cdots, x_n\}$;
for *each arttribute* x_i *in* $\mathcal{X}$ **do**
  $R^Q_i = \frac{\sum_{j=1}^{m}(x_{ij}-\overline{x_i})(y_j-\overline{y})}{\sqrt{\sum_{j=1}^{m}(x_{ij}-\overline{x_i})^2}\sqrt{\sum_{j=1}^{m}(y_j-\overline{y})^2}}$;
for *each pair of attributes* x_i *and* x_j *in* $\mathcal{X}$ **do**
  if $|r_{ij}| > \beta$ **then**
    If $|R^Q_i| < |R^Q_j|$, remove x_i from $\mathcal{X}$; Otherwise, remove x_j from $\mathcal{X}$;
Sort $\{|R^Q_i| | x_i \in \mathcal{X}\}$ in descending order, and the top-k of them are $\{|R^Q_{a_1}|, \cdots, |R^Q_{a_k}|\}$;
$\mathcal{K}^Q = \{x_{a_1}, \cdots, x_{a_k}\}$;
Return $\mathcal{K}^Q$.

3.2 Piecewise Linear Fitting Based Relative Kernel Dataset Retrieving Algorithm

In the RKDR algorithm, only linear correlation is considered between each attribute and the target value. When the correlations between x_i and y_Q are exponential, quadric, logarithmic, or *etc*, the obtained relative kernel dataset $\mathcal{K}^Q$ may incur non-negligible error for estimating the result of query Q by $\mathcal{K}^Q$. Therefore, we improve the RKDR algorithm in this section. Instead of linear correlation coefficient, the PLF-RKDR algorithm applies the piecewise linear fitting method to estimate the correlation coefficients between attributes and the target value of Q. The PLF-RKDR algorithm has the following two steps.

Step 1. Calculate the candidate relative kernel dataset $\mathcal{X}$.

At first, the candidate relative kernel dataset $\mathcal{X}$ includes all n attributes.

Step 1.1. Calculate the correlation coefficient between each attribute $x_i \in \mathcal{X}$ and the target value y_Q by the piecewise linear fitting.

We applied the method in [18] to recursively retrieve the optimal segment points in piecewise linear fitting. For attribute x_i, the training examples are sorted by the increasing order of x_i, denoted as $(x_{i1}, y_{Q1}), \cdots, (x_{im}, y_{Qm})$. $F_i^{(s,e)}(x)$

is the linear fitting function of $(x_{is}, y_{Qs}), (x_{i(s+1)}, y_{Q(s+1)}), \cdots, (x_{ie}, y_{Qe})$ obtained by the least-squares method. The error sum of squares of linear fitting function $F_i^{(s,e)}(x)$ is $E(s,e)$. The optimal segment point (x_{ij}, y_{Qj}) satisfies that $j = \arg\min_{s<k<e}(E(s,k) + E(k,e))$. This process is conducted recursively until $E(s,e) - (E(s,j) + E(j,e)) \leq \gamma$. The detailed algorithm is shown in Algorithm 3, which returns the set of segment points SP_i. The size of SP_i is denoted as p_i.

According to the segment points in SP_i, training examples are divided into $p_i - 1$ subsets. Each subset is denoted as $S_j = \{(x_{ij^s}, y_{Qj^s}), \cdots, (x_{ij^e}, y_{Qj^e})\}$, where $1 \leq j < p_i$. The linear correlation coefficient R_{ij} for S_j and the linear function f_{ij} for S_j is calculated by the least-squares method. Therefore, the correlation coefficient R_i^Q of x_i is presented as the weighted average correlation coefficient of all $R_{ij}(1 \leq j < p_i)$. The piecewise linear function f_i is as follows.

$$f_i = \begin{cases} f_{i1}, \ x_{i1^s} < x_i < x_{i1^e} \\ \cdots, \ \cdots \\ f_{ip_i}, \ x_{ip_i{}^s} < x_i < x_{ip_i{}^e} \end{cases} \tag{4}$$

Step 1.2. This step is the same as the Step 1.2 of the RKDR algorithm.

Step 2. Retrieve the relative kernel dataset $\mathcal{K}^Q$ from $\mathcal{X}$.

This step is the same as the Step 2 of the RKDR algorithm in Section 3.1.

Therefore, (k, β)-Relative Kernel Dataset $\mathcal{K}^Q$ for query Q is obtained. The detailed PLF-KDDR Algorithm is shown in Algorithm 2.

Then once the query Q is issued by users again, we provide an approximate method to estimate the result of Q by the sensory data in $\mathcal{K}^Q$. For each attribute x_{a_i} in $\mathcal{K}^Q$, the corresponding piecewise linear function f_i is assigned a weight according to $R_{a_i}^Q$. $\mathcal{F}$ is a new piecewise linear function combining all piecewise linear function $f_i(1 \leq i \leq k)$ with weight $R_{a_i}^Q$. Therefore, the approximate result of query Q is obtained by put sensory data in $\mathcal{K}^Q$ into the function $\mathcal{F}$.

Algorithm 2: PLF-Relative Kernel Dataset Retrieving Algorithm

Input: query Q, training set $\{t_1, \cdots, t_m\}$; compatible parameter β; required size k; threshold γ

Output: (k, β)-Relative Kernel Dataset $\mathcal{K}^Q$

$\mathcal{X} = \{x_1, x_2, \cdots, x_n\}$;
for *each arttribute* x_i *in* $\mathcal{X}$ **do**
  Sort training examples by increasing order of x_i, i.e. $(x_{i1}, y_{Q1}), \cdots, (x_{im}, y_{Qm})$;
  $SP_i = \{1, m\} \cup$ **LS-PLF**$(1, m, \infty, \gamma)$; $p_i = |SP_i|$;
  $(x_{i1}, y_{Q1}), \cdots, (x_{im}, y_{Qm})$ are divided into $p_i - 1$ subsets $S_1, \cdots, S_{p_i - 1}$;
  for *each subset* $S_j(1 \leq j < p_i)$ **do**
    $R_{ij} = \frac{\sum_{k=j^s}^{j^e}(x_{ik} - \overline{x_{ij}})(y_{Qk} - \overline{y_{ij}})}{\sqrt{\sum_{k=j^s}^{j^e}(x_{ik} - \overline{x_{ij}})^2}\sqrt{\sum_{k=j^s}^{j^e}(y_{Qk} - \overline{y_{ij}})^2}}$;
    Calculate piecewise linear function f_{ij} by the the least-squares method;
  $R_i^Q = \sum_{j=1}^{p_i - 1} R_{ij} \times \frac{j^e - j^s}{x_{im} - x_{i1}}$;
for *each pair of attributes* x_i *and* x_j *in* $\mathcal{X}$ **do**
  if $|r_{ij}| > \beta$ **then**
    If $|R_i^Q| < |R_j^Q|$, remove x_i from $\mathcal{X}$; Otherwise, remove x_j from $\mathcal{X}$;
Sort $\{|R_i^Q||x_i \in \mathcal{X}\}$ in descending order, and the top-k of them are $\{|R_{a_1}^Q|, \cdots, |R_{a_k}^Q|\}$;
Return $\mathcal{K}^Q = \{x_{a_1}, \cdots, x_{a_k}\}$.

Algorithm 3: Least-Squares Based Piecewise Linear Fitting (LS-PLF)

Input: Two endpoints s and e, $E(s,e)$, the threshold γ
Output: The set of segment points SP

$E_{min} = \infty$, $J_{min} = 0$;
if $e - s > 1$ **then**
  $j = \arg\min_{s<k<e}(E(s,k) + E(k,e))$;
  if $E(s,e) - (E(s,j) + E(j,e)) > \gamma$ **then**
    $SP = SP \cup \{j\}$;
    LS-PLF$(s, j, E(s,j), \gamma)$;
    LS-PLF$(j, e, E(j,e), \gamma)$;
Return SP.

4 The Performance Analysis

For RKDR Algorithm, the computation complexity of computing linear correlation coefficients is $O(mn)$ for n attributes and m training examples. Besides, the computation complexity of calculating candidate relative kernel dataset in Step 1 is $O(mn^2)$, since there are $O(n^2)$ pairs of attributes. In Step 2, the computation complexity of sorting correlation coefficients is $O(n \log n)$. In conclusion, the computation complexity for RKDR algorithm is $O(mn^2)$.

For PLF-RKDR Algorithm, the worst computation complexity of calculating optimal segment points for each attribute is $O(m^3)$. Then the computation complexity of Step 1.1 is $O(nm^3)$. For Step 1.2, the computation complexity of calculating candidate relative kernel dataset is $O(mn^2)$. In Step 2, the computation complexity of sorting correlation coefficients is $O(n \log n)$. Therefore, the computation complexity for PLF-RKDR Algorithm is $O(nm^3)$.

5 Simulation Results

This section evaluates the performance of our proposed RKDR algorithm and PLF-RKDR algorithm by extensive simulations. Simulations on both simulation dataset and real dataset are carried out.

For simulation dataset, we generate two functions among $n = 15$ attributes and the target value, denoted as $y_Q = f_1(x_1, \cdots, x_{15})$ and $y_Q = f_2(x_1, \cdots, x_{15})$. Each training example contains fifteen randomly generated values of attributes $x_1, \cdots, x_{15}$ and a target value generated by function f_1 or f_2.

The real dataset is collected from a mobile device when a person held it making 7 types of motions, which are numbered as motion 1 to 7. For motion $i (1 \leq i \leq 7)$, y_Q is set as 1 when it happens, otherwise, it is set as 0. The attributes are the X-, Y- and Z-axis of a three-axis accelerometer and a three-axis gyroscope collected by the mobile device. Each training example contains 6 values of attributes and a target value indicating whether motion i happens.

In the following simulations, the relative error er is applied to evaluate the performance of our algorithms. In fact, $er_1 = |\frac{y'_Q - y_Q}{y_Q}|$ is the absolute error,

where y'_Q is the target value estimated by sensory data in our (k, β)-Relative Kernel Dataset $\mathcal{K}^Q$ and y_Q is the true target value of query Q. However, y_Q is unknown or inaccessible in real physical world. Therefore, y_Q is estimated by $\widehat{y_Q}$, which is the target value estimated by sensory data of all n available attributes. That is, the relative error er is defined as $er = |\frac{y'_Q - \widehat{y_Q}}{\widehat{y_Q}}|$.

Firstly, comparison experiments are carried out to compare the performances of our RKDR Algorithm and PLF-RKDR Algorithm. Secondly, only PLF-RKDR algorithm is evaluated in both simulation dataset and real dataset.

5.1 The Comparison Experiments of RKDR Algorithm and PLF-RKDR Algorithm

A group of comparison experiments are carried out to compare RKDR Algorithm and PLF-RKDR Algorithm on simulation dataset. The simulation dataset is generated by two functions f_1 and f_2, each with $m = 3000$ training examples. Each training example contains $n = 15$ attributes and a target value of the given query. We compare the relative error er of our RKDR Algorithm and PLF-RKDR Algorithm with k increases, where k is the size of the (k, β)-Relative Kernel Dataset. The simulation results are presented in Fig.1. Each data point presented in Fig.1 is the average of simulation results on 500 times of query.

Fig.1 shows that the relative error of PLF-RKDR Algorithm is much smaller than that of RKDR Algorithm no matter how much k is. Particularly, the relative error of RKDR Algorithm is almost twice as much as that of PLF-RKDR Algorithm when $k > 5$. Therefore, the performance of PLF-RKDR Algorithm is better than RKDR Algorithm. Besides, the relative error of both algorithms on simulation dataset of f_1 is much less than that of f_2, indicating that linear correlation coefficient may not be suitable for function f_2.

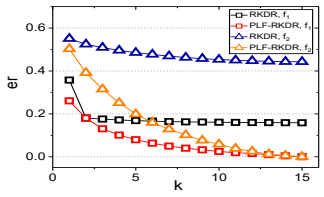


Fig. 1: RKDR vs PLF-RKDR

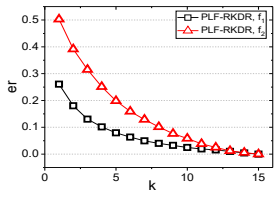


Fig. 2: The impact of k

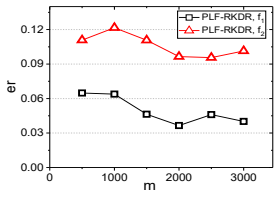


Fig. 3: The impact of m

5.2 The Performance of PLF-RKDR Algorithm on Simulation Dataset

The first group of simulations investigate the impact of k, the size of t(k, β)-Relative Kernel Dataset, on the performance of PLF-RKDR Algorithm on simulation dataset. We evaluate the relative error er with the increase of k. The

simulation results are presented in Fig.2. Each data point presented in Fig.2 is the average of simulation results on 500 times of query.

Fig.2 presents that the relative error er of PLF-RKDR Algorithm on both simulation datasets of function f_1 and f_2 decreases with k increases. Besides, the performances on simulation dataset with function f_1 and function f_2 are different with the same k since different query has different Relative Kernel Dataset. That explains why we retrieve relative kernel dataset for a given continuous query. Furthermore, Fig.2 also presents that, the relative error reduces to 0.05 when k is only half of n, this error is pretty small in practice. That is, the network can save a half of energy when sacrifices only a few accuracy.

The second group of simulations evaluate the impact of m, the size of training examples, on the performance of PLF-RKDR Algorithm on simulation dataset. Different scales of training set is applied, i.e. m is set as 500, 1000, 1500, 2000, 2500 and 3000. The size of (k, β)-Relative Kernel Dataset is set to $k = 8$.

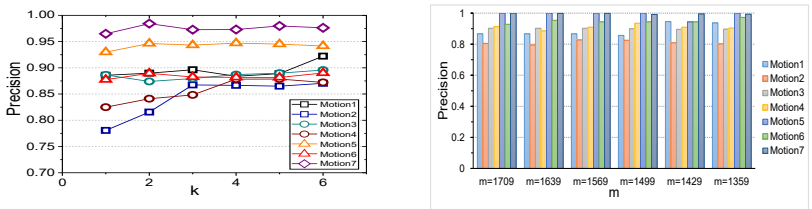
Fig.3 presents the performance of our PLF-RKDR algorithm on the impact of m. Each data point presented in Fig.3 is the average of simulation results on 500 times of query. Fig.3 presents that the relative error er decreases slowly with m increases. It is worth noting that even m is relatively small, the relative error is still under 0.15 when k is no less than a half of n.

5.3 The Performance of PLF-RKDR Algorithm on Real Dataset

As motioned above, the real dataset contains 6 attributes of X-, Y- and Z-axis of a three axis accelerometer and a three axis gyroscope. The target value of motion $i (1 \leq i \leq 7)$ is set as 1 when motion i happens, otherwise, it is set as 0. If the estimated target value is closer to 1, we judge that the motion happens.

The first group of simulations show the motion judgement precisions for each motion affected by k. The simulation results of $m = 4796$ training examples are presented in Fig.4(a). Each data point presented in Fig.4(a) is the average of the simulation results on 1511 times of query. It shows that the precision for each motion increases with the increase of k. As the figure shows, 80% of the motions are correctly judged by our PLF-RKDR Algorithm.

The second group of simulations study the motion judgement precisions under different sizes of training examples, which are shown in Fig.4(b). The size of (k, β)-Relative Rernel Dataset is set as $k = 4$. Each data point presented in Fig.4(b) is the average of simulation results on 585 times of query. Fig.4(b) presents that more than 80% of the motions are correctly judged. It also shows that there was no significant relationship between the precision and m.



(a) The precision affected by k (b) The precision affected by m

Fig. 4: The Performance of PLF-RKDR Algorithm in Real Dataset

6 Related Works

Most of the related works are about principal components analysis in WSN, which reduces data transmission, and then reduces the energy consumption. The work in [19] combines the compressive sensing and principal component analysis to efficiently recover the whole dataset through a small subset of data. Another work in [20] designs two distributed consensus-based algorithms to process principal component analysis in WSN. The authors in [21] propose algorithm to aggregate data through principal component analysis, which is a powerful technique for dimensionality reduction. However, these works only focus on retrieving principal components for the network and the given query has not been taken into account yet. Therefore, these methods are not suitable to retrieve relative kernel dataset from big sensory data for a given query.

The authors provide the centralized and distributed algorithms to retrieve dominant dataset from big sensory data under the condition that the information loss rate required by users is guaranteed in [16, 17]. However, the dominant dataset is a general one for all queries instead of a specific one for a given query. This work cannot retrieve relative kernel dataset for a given query.

7 Conclusion

This paper studies retrieving relative kernel dataset for continuous queries from big sensory data in WSNs. The RKDR Algorithm and PLF-RKDR Algorithm are proposed to retrieve (k, β)-Relative Kernel Dataset $\mathcal{K}^Q$ for a given query Q. Then we provide methods to estimate the approximate result of query Q by sensory data in $\mathcal{K}^Q$. Extensive simulations are conducted, which demonstrate that (k, β)-Relative Kernel Dataset $\mathcal{K}^Q$ retrieved by our algorithm can estimate the result of query Q with high accuracy.

Acknowledgment

This work is partly supported by the National Natural Science Foundation of China under Grant NO. 61632010, 61502116, U1509216, 61370217, the National Science Foundation (NSF) under grant NO.1741277.

References

1. G. Says, "6.4 billion connected things will be in use in 2016, up 30 percent from 2015," *Gartner, Inc*, 2015.
2. K. Tillman, "How many internet connections are in the world? right. now," *CISCO. URL: h ttp://blogs. cisco. com/news/cisco-connections-counter*, 2013.
3. J. Yu, Y. Qi, G. Wang, and X. Gu, "A cluster-based routing protocol for wireless sensor networks with nonuniform node distribution," *AEUE - International Journal of Electronics and Communications*, vol. 66, no. 1, pp. 54–61, 2012.

4. J. Yu, N. Wang, and G. Wang, *Constructing minimum extended weakly-connected dominating sets for clustering in ad hoc networks*. Academic Press, Inc., 2012.
5. J. Yu, X. Ning, Y. Sun, S. Wang, and Y. Wang, "Constructing a self-stabilizing cds with bounded diameter in wireless networks under sinr," in *INFOCOM, 2017 Proceedings IEEE*, 2017, pp. 1–9.
6. T. Shi, S. Cheng, Z. Cai, and J. Li, "Adaptive connected dominating set discovering algorithm in energy-harvest sensor networks," in *INFOCOM, 2016 Proceedings IEEE*, 2016, pp. 1–9.
7. J. Li, S. Cheng, Z. Cai, J. Yu, C. Wang, and Y. Li, "Approximate holistic aggregation in wireless sensor networks," *TOSN*, vol. 13, no. 2, pp. 11:1–11:24, 2017.
8. Z. He, Z. Cai, S. Cheng, and X. Wang, "Approximate aggregation for tracking quantiles and range countings in wireless sensor networks," *Theor. Comput. Sci.*, vol. 607, pp. 381–390, 2015.
9. Z. C. L. C. Quan Chen, Hong Gao and J. Li, "Energy-collision aware data aggregation scheduling for energy harvesting sensor networks," in *INFOCOM, 2018 Proceedings IEEE*.
10. S. Cheng, Z. Cai, and J. Li, "Curve query processing in wireless sensor networks," *IEEE Trans. Vehicular Technology*, vol. 64, no. 11, pp. 5198–5209, 2015.
11. X. Zheng, Z. Cai, J. Li, and H. Gao, "A study on application-aware scheduling in wireless networks," *IEEE Trans. Mob. Comput.*, vol. 16, no. 7, pp. 1787–1801, 2017.
12. S. Cheng, J. Li, Q. Ren, and L. Yu, "Bernoulli sampling based (ε, δ)-approximate aggregation in large-scale sensor networks," in *Proceedings of the 29th conference on Information communications*. IEEE Press, 2010, pp. 1181–1189.
13. Z. Huang, L. Wang, K. Yi, and Y. Liu, "Sampling based algorithms for quantile computation in sensor networks," in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*. ACM, 2011, pp. 745–756.
14. H. Zheng, S. Xiao, X. Wang, X. Tian, and M. Guizani, "Capacity and delay analysis for data gathering with compressive sensing in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 2, pp. 917–927, 2013.
15. H. Wang, Y. Zhu, and Q. Zhang, "Compressive sensing based monitoring with vehicular networks," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 2823–2831.
16. S. Cheng, Z. Cai, J. Li, and X. Fang, "Drawing dominant dataset from big sensory data in wireless sensor networks," in *INFOCOM, 2015 Proceedings IEEE*. IEEE, 2015, pp. 531–539.
17. S. Cheng, Z. Cai, J. Li, and H. Gao, "Extracting kernel dataset from big sensory data in wireless sensor networks," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 4, pp. 813–827, 2017.
18. T. L. L. Zong-tian, "Least-squares method piecewise linear fitting," *Computer Science*, p. S1, 2012.
19. R. Masiero, G. Quer, D. Munaretto, M. Rossi, J. Widmer, and M. Zorzi, "Data acquisition through joint compressive sensing and principal component analysis," in *GLOBECOM 2009. IEEE*. IEEE, 2009, pp. 1–6.
20. S. V. Macua, P. Belanovic, and S. Zazo, "Consensus-based distributed principal component analysis in wireless sensor networks," in *SPAWC, 2010 IEEE Eleventh International Workshop on*. IEEE, 2010, pp. 1–5.
21. A. Rooshenas, H. R. Rabiee, A. Movaghar, and M. Y. Naderi, "Reducing the data transmission in wireless sensor networks using the principal component analysis," in *ISSNIP, 2010 Sixth International Conference on*. IEEE, 2010, pp. 133–138.