Improved Data analysis in Wireless Sensor Networks using K-Nearest Neighbor Algorithm

Shivvasangari Subramani
[shivva1@umbc.edu]
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County, USA

Keywords: Wireless Sensor Networks, K-Nearest Neighbor Algorithm, Cross Validation

1 Introduction

Wireless Sensor Network has millions of nodes, deployed over the wide region for sensing various parameters like temperature, presence of an animal, earthquake etc. on for sensing various parameters like temperature, presence of an animal, earthquake etc. As these nodes are deployed in area where human access is difficult, they rely on battery power for functioning.

This study involves the generation of the data for these failed nodes for any event for which it was previously functioning. K-nearest Neighbor algorithm is used for the generation of data. The results obtained with the training data have approximately 92% accuracy.

2 Problem Definition

The main purpose of sensor networks is to monitor, combine, analyze and respond to the data collected by hundreds (thousands) sensors distributed in the physical world in a timely manner. The level of activity of node, i.e. transmission reception and processing of data, affects the energy available for the node. If the level of activity is high then battery drains over quickly and that node dies. Hence the network can’t receive any information from that node which is a crucial one in determining the parameter.

Also, base stations are highly prone to receive corrupted data from sensors due to noisy wireless environment. Hence the data received from the sensor nodes needs to be validated.

3 Proposed Method

As many nodes deployed in a region, it is possible to determine the approximate values for these missing data using machine learning concept of deciding the behavior of a node either from its past behavior and its data correlation with neighboring nodes for a given condition or by the influence of the neighbors of the node at the given time. This concept has been used as the base of the implemented algorithm.

3.1 Related Work

There has been some research already in this field, where k-nearest neighbor with mutual information was used for simultaneous classification and missing data imputation. The proposed algorithm was a novel KNN imputation procedure using a feature-weighted distance metric based on mutual information (MI). This method provided a missing data estimation which aimed at solving the classification task, i.e., it provided an imputed dataset which could enhance the classification performance. This distance metric which was MI based was used to implement an effective KNN classifier. To give the proof for the efficiency and robustness of the algorithm that was proposed results obtained from both artificial and classification data were used.

Publisher: Elsevier Science Publishers B. V. Amsterdam, The Netherlands, The Netherlands

4 Intuition

For finding the values of failed sensor nodes various machine learning algorithms can be used as follows:

1. Linear Regression
2. 1-Nearest Neighbor Algorithm
3. K-Nearest Neighbor Algorithm
The linear regression algorithm will collect data values from all sensor nodes. This is not desirable as in sensor networks the data of the nodes at a greater distance have no or little correlation with that of the failed node. For example suppose at a particular time instant we are interested in obtaining values of parameters like temperature, voltage, humidity etc from a sensor node ID 19, the values sensed and returned by node 19 are not have any correlation with the values of the far away nodes like node ID 41. Hence Linear regression is not a good choice.

1-Nearest Neighbor algorithm will also not work because the values obtained from the current sensor node will be influenced by only one node in the vicinity that is that node which has the least distance from the current node. The distance will be the Euclidean distance. Wireless Sensor Networks has a noisy environment. If this particular neighbor’s data is highly corrupted by noise then the result we will get for the failed node will obviously be wrong. So the results returned should not be the influence of one node.

It turns out that the k-nearest neighbor algorithm is the best option because the parameters obtained for the failed node will have the influence of nearest k-neighbors, which will minimize the noise. Again the distance to determine the nearest-k neighbors is the Euclidean distance. Advantages of this algorithm are that training is very fast, learn complex target functions, and don’t lose information.

5 Experiments

The ultimate goal of the experiments is to predict the data of the susceptible node at time t. The node for which data should be generated is termed as the query node. Two methods are used to find out the data of query node:

1. K-NN based method: Given the data of the neighbors at time t, predict the data of query node.
2. Past correlation method: Given the correlation of query node with neighbors before time t, and data of the neighbors at time t, predict the data of query node.

5.1 Datasets used

Temperature, humidity, and light data, along with the voltage level of the batteries at each node collected using the 54-node sensor network deployment. The data was collected every 30 seconds, starting around 1am on February 28th 2004.

Figure 1 shows the testbed used for the experiments. The numbered black dots are the sensors deployed in the various buildings of a campus.

URL for the dataset is: http://www-2.cs.cmu.edu/~guestrin/Research/Data/

5.2 K-NN Based Method

The algorithm implemented for K-NN method is as follows:

- Obtain the co-ordinates of all the sensor nodes physically distributed in the network.
- Select a node and the time (test instance in ML terms) randomly as susceptible node.
- Calculate the Euclidean distance of all sensor nodes from the susceptible node.
- Sorted the distance values and determined Kth nearest neighbor’s distance.
- Determine K-nearest neighbors.
- Extract the values of these k-nearest neighbors for the time interval of query time+/-30 seconds and query time-60 seconds.
- Determine arithmetic mean of these values which is the predicted of the query instance.
The value of k is going to determine the impact of the neighbors on the query node. Higher the K, nodes at the greater distance will impact the query node which is an undesirable one, where as when K=1, noise of the immediate neighbor will influence the prediction a lot. In order to get the optimum value of K, accuracy of the test data is found for various value of K with various number of instances. The results showed that accuracy is maximum when K=2 for maximum number of instances. This is illustrated in Figure 2.

![Figure 2. Accuracy Vs K value](image)

When the individual instance is chosen, the result was:
- Time: 3990 and Node ID: 41
- Output when k is: 2
  18.334650, 39.704775, 326.600000, 2.675345
- Output when k is: 3
  18.109250, 40.452563, 370.760000, 2.678372
- Correct value is:
  18.1852, 39.6878, 382.72, 2.67532

Output when K=3 is higher than the output when k=2. The inference from the above result is that optimum value of K largely depends on the test instance. As K=2 gave maximum accuracy of 92% to lot of instances it was taken as default value for all the experiments.

### 5.3 Past Correlation method

The algorithm implemented for Past correlation method is as follows:
- Initialize weight vector to zero
- Generate a test instance randomly.
- Find the neighbors of the test node using K-NN algorithm with K=2.
- Each time test node is encountered in the record i.e, in the dataset, find the difference between the test node’s value and average value of its neighbors at that time and update the weight vector.
- Find the average of the neighbor’s data at query time and add it with the updated weight vector, test node’s values at query time is predicted.
  Data of query node= Average of neighbors+ updated weight vector.

### 6 Observations

1. K-NN method was tested with 500 instances to determine the accuracy.

Observed output is as follows:
- For K= 1 System gave output for 84.000000 % instances with accuracy 89.549441
- For K= 2 System gave output for 84.000000 % instances with accuracy 91.648655
- For K= 3 System gave output for 84.000000 % instances with accuracy 86.670694
- For K= 4 System gave output for 84.000000 % instances with accuracy 85.723917
- For K= 5 System gave output for 84.000000 % instances with accuracy 85.866055
- For K= 6 System gave output for 84.000000 % instances with accuracy 85.624496
- For K= 7 System gave output for 84.000000 % instances with accuracy 87.459689
- For K= 8 System gave output for 84.000000 % instances with accuracy 86.854976
- For K= 9 System gave output for 84.000000 % instances with accuracy 85.020204
- For K= 10 System gave output for 84.000000 % instances with accuracy 85.611902

2. Past Correlation method was tested with 500 instances to determine the accuracy.

Observed output is:
- System gave output for 68% instances with the accuracy of 71.06%

By comparing the outputs of both the methods, we can conclude that K-NN based method with K=2 performs better than past correlation method.
8 Future Directions

In practice wireless sensor networks have following setup.

- Base station sends request to the sensors in the network.
- Parameter values generated by the sensor node are sent to the base station.
- Base station may acknowledge the sensor about the data it received.

If we have K-NN based method implemented in the base station, then after receiving every data it can be checked for integrity and validity of the sensors can be ensured. This increases the decision accuracy. It needs following improvements in the present method.

1. Automatic detection of the query node: In the current implementation, the query node is generated randomly. In order to make the system reliable, in real time we need to find the instances having the missing /incorrect values. This can be done by simple checking the integrity of the data received at a particular time from a node. If it is having abnormal values then that node will become the query node.

2. Detection of failed nodes: If the same node acts as the query node for more than 3 times that node can be concluded as dead node and can be excluded in the next iterations.

9 Conclusions

The experiments for the generation of the missing values for the sensors in the wireless network will improve the accuracy of the parameter determined out of it. This is beneficial because in the adhoc sensor network, as any sensor node can fail due to battery power consumption and various other reasons. So getting the relevant values at that particular time can be achieved by our method. This will also validate the data received from each sensor.

10 References