AD HOC SENSOR NETWORK LOCALIZATION USING DISTRIBUTED KERNEL REGRESSION ALGORITHMS

*Chaopin Zhu, Anthony Kuh**

Dept. of Electrical Engineering
University of Hawaii
Honolulu, HI 96822
{czhu,kuh}@spectra.eng.hawaii.edu

ABSTRACT

In this paper we apply distributed kernel regression methods to perform sensor network localization. This follows up on earlier work where a centralized kernel regression algorithm was considered to perform localization. Here we examine the tradeoffs between using distributed algorithms versus centralized algorithms in terms of communication costs, computational costs, and performance of the estimate. Simulation results demonstrate that distributed methods work well with comparable performance to centralized algorithms with less communication costs.

Index Terms— distributed learning, kernel methods, sensor network localization

1. INTRODUCTION

This paper extends previous work on ad hoc sensor network localization problem using signal strength information by using distributed algorithms with kernel regression methods. The earlier work used signal strength information to form information vectors for each sensor, represent this information in a kernel matrix, and then use either classification or regression methods [1, 2, 9]. to perform localization. These algorithms give good performance, but may be impractical as centralized algorithms require extensive signal strength information from all sensors and this may not be feasible because of energy and communication constraints imposed on the sensors. Signal strength between two sensors depends on the distance between the sensors and we formulate a distributed algorithm where sensors are first partitioned into local regions, a kernel regression problem is solved for each region, and then localization is performed by resolving the local algorithms. Simulation results show that the distributed algorithms can achieve good performance and the tradeoffs between communication, computation, and error performance.

Ad hoc sensor networks are being deployed in a variety of applications from environmental sensing to security and intrusion detection to medical monitoring [3]. We will consider the model developed in [1] consisting of *base nodes* where the locations are known and simple nodes called *motes* where the locations of the motes are unknown. In this model wireless radio is used where the received signal power / strength is proportional to a power function of the distance between a transmitter and a receiver [5], i.e.

$$s \propto P d^{-\eta}, \tag{1}$$

where s is the received signal strength, $\eta > 2$ is a constant, P is the transmitting power, and d is the distance between the transmitter and the receiver. Rich scattering in the real world makes the received signal strength quite noisy.

Using the model of [1] an ad hoc network of size n is deployed in a connected two-dimension geographical area $\mathcal{T}$. An integer from 1 to n represents each node as its ID. Denote the set of all nodes in the network by $\mathcal{N} = \{1, \cdots, n\}$. The location of node $i \in \mathcal{N}$ is denoted by P_i (a complex number with east-west coordinate represented by the real part and north-south coordinate represented by the imaginary part). Assume that the first m nodes are base nodes, which have more computational capabilities than motes and the locations of the base nodes are known with $P_1 = p_1, \cdots, P_m = p_m$ where $p_i \in \mathcal{C}$. The base nodes are also able to get signal strength information from the other base nodes. We need to estimate the location of the other $n - m$ motes which could be mobile (i.e. to find $P_{m+1}, \cdots, P_n$).

As mentioned above signal strength information between two nodes depends on the distance between the two nodes, scattering, and additive noise. The localization problem consists of a system with a set of parameters that are learned. The input to the system is the signal strength information between base nodes and the outputs are estimates of base node locations. The kernel methods used in [1, 2, 9] are based on first forming information vectors from signal strength and then using the information vectors to form kernels. Then Support Vector Machine, [6] or least squares kernel methods [8] are used to find the parameters (i.e. support vector weights α and threshold value b).

*Portions of this work were supported by RTI.

This paper looks to improve previous kernel localization algorithms by exploiting signal strength's relationship to distance and the location of the base nodes to create distributed kernel localization algorithms. These algorithms have similar performance to centralized algorithms, but can take advantage of parallel processing and reduced communication and energy costs. The coarse localization / fine localization algorithms discussed in [1, 2] are distributed algorithms, but don't take advantage of base node locations and information about signal strength. A kernel regression algorithm was formulated in [10] for sensor networks as the algorithm considered is based on Gaussian elimination and a message passing algorithm to converge to a centralized regression problem. Distributed learning is also considered for sensor networks in [11] using alternate projections. These methods are different from those considered here as the distributed algorithms are based on signal strength information and base node locations, the distributed algorithms are different than centralized algorithms, and the resolution of the distributed algorithms is based on weighting of local estimates.

The paper is organized as follows. Section 2 reviews the centralized base node regression algorithm. Section 3 discusses training and testing of the distributed learning algorithm. Simulations are conducted in Section 4 comparing distributed algorithms to the base node regression algorithm and Section 5 summarizes the main contributions of this paper and discusses some future research directions.

2. BASE NODE REGRESSION

The base node regression model is discussed in detail in [9] and uses a least squares kernel subspace regression model [7]. For this model base nodes are stationary and motes can move.

These methods assume that there is a base node or a central processor to collect all the signal strength readings between base nodes and to do the training using least squares kernel methods discussed in [7]. Motes are given the support vector information vectors, support vector values α, and threshold b. The motes then use equation (2) to estimate their location.

The first step is to choose an information vector from the signal strength readings of the base nodes. Corresponding to each base node i, $1 \leq i \leq m$ will be an information vector of length m where the jth component describes the signal strength, $s(p_j, p_i)$ from node j to node i. The information vector is described by

$$x_i = \begin{bmatrix} s(p_1, p_i) \\ s(p_2, p_i) \\ \vdots \\ s(p_m, p_i) \end{bmatrix}.$$

The m information vectors correspond to m training examples. The output, p_i for each training example is the location of the base node. We then use m_S of these information vectors as support vectors. We can use a selection criteria discussed in [7] or an unsupervised algorithm such as the k-Means algorithm (based on locations) to choose support vectors. The least squares kernel subspace regression algorithm is then applied to get the parameters α and b.

For mote j, $m < j \leq n$ we then form an information vector from signal strengths to get

$$x_j = \begin{bmatrix} s(p_1, p_j) \\ s(p_2, p_j) \\ \vdots \\ s(p_m, p_j) \end{bmatrix}.$$

The mote then uses the following equation to estimate its own location at time.

$$f(x_j) = \sum_{i=1}^{m} \alpha_i K(x_j, x_i) + b), \quad m < j \leq n \qquad (2)$$

3. DISTRIBUTED LEARNING ALGORITHM

This section examines a distributed kernel learning algorithm that considers base node locations and signal strength information.

3.1. Training base nodes

The distributed algorithm consists of two steps. We first partition the base nodes into different regions. Each region then comes up with an estimate for location of a sensor based on parameters α and threshold value b. The location estimate will be more accurate in the region where base nodes are located, then outside the region.

We partition base nodes into r different sets. Set $R(i)$, $1 \leq i \leq r$ will also be called region $R(i)$ which consists of base nodes $j_1, \ldots, j_{r(i)}$ where $r(i) \geq 1$ is the number of base nodes in the ith region. Since estimation algorithm is based on signal strength which is directly dependent on distance, base nodes within a region should for the most part be closer in distance than when compared to base nodes outside the region. Different algorithms can be used to partition the base nodes into the different regions with a goal of having a computationally efficient algorithms that can take advantage of parallel processing such as a distributed k-Means algorithm discussed in [12]. Base nodes in each region will also have a tag identifying region.

After partitioning into regions, each region will use the subspace kernel regression algorithm to come up with a set of parameters α that are associated with base nodes that are support vectors and threshold value b. Let $S(i)$ denote the set of all nodes that will be used in the regression algorithm for the ith region with $R(i) \subset S(i)$. Each node in $S(i)$ will have signal strength with at least one node of $R(i)$ that is greater than some specified threshold value s_0. For region i information vector for sensor i' will be given by

$$x(i')[i] = (s(j,i'), j \in S(i))^T$$

and the estimate for the location of sensor i' is given by

$$\hat{p}(i')[i] = \sum_{j \in S(i)} \alpha_j[i] K(x(i')[i], x(j)[i]) + b[i]. \quad (3)$$

When the subspace algorithm is used a subset of base nodes in $S(i)$ will be support vectors and only these nodes will have nonzero α values. Each region will have its own estimate of location so there will be a total of r different estimates for location of sensor i'. Region i will have an estimate with parameters given by $\alpha[i]$ and $b[i]$ with each base node in i storing these set of parameters.

Remark 1: The algorithm's most critical parameter is r the number of regions. If $r = 1$, then this corresponds to centralized base node regression where one subspace kernel regression is performed on the m base nodes. If $r = m$, then every base node performs a subspace kernel regression algorithm and this is similar to the k nearest neighbors algorithm. Good values for r depend on the number of base nodes m, their physical deployment, and signal strength. Let $m_{S(i)}$ denote the number of support vectors for region i and m_i be the cardinality of $S(i)$. Note that $m_{S(i)}$ will depend on the signal strength of nodes in $S(i)$ and the desired accuracy we would like. Finding $\alpha[i]$ and $b[i]$ for least squares kernel regression involves matrix multiplication and inversion operations. This will take $\mathcal{O}({m_{S(i)}}^3 + {m_{S(i)}}^2 m(i))$ operations.

Remark 2: For most ad hoc sensor networks, the sensors will be deployed over a wide geographical range with only a few base node sensors having strong signal strength with a given mote. This will result in only a few of the regions having significant nonzero information vectors for a given mote. In general, only a few regions will be part of the location estimate of a given mote.

3.2. Mote Localization

Each mote will store information about regions that are in close proximity to the mote. These are base nodes in regions that have signal strength greater than a certain threshold value s_0. For each region i that is of close proximity, the mote will store $\alpha[i]$ and $b[i]$, and information vectors for base nodes in region i. The mote i' will get its own information vector $x(i')[i]$ and then compute its location estimate from region i, $\hat{p}(i')[i]$. The mote i' will compute a location estimate for each of the regions it is in close proximity to. It will then compute a location estimate that is the weighted average of the location estimate of each region that is in close proximity to the mote. The weighting will depend on the number of base nodes in i that are close to mote i' and their signal strengths. Let $N(i')$ denote the set of regions that are in close proximity to mote i', then the location estimate is given by

$$\hat{p}(i') = \sum_{i \in N(i')} w(i)\hat{p}(i')[i] \quad (4)$$

where the weights $w(i)$ are positive numbers that sum to 1. Weights are given by

$$w(i) = \frac{\sum_{j \in R(i)} g(s(j,i'))}{\sum_{j=1}^{m} g(s(j,i'))} \quad (5)$$

with possible values for weighting functions $g()$ including $g(z) = z$ and $g(z) = \mathbf{1}(z > s_0)$ with $\mathbf{1}()$ denoting an indicator function.

If the motes are mobile, then the regions that are in close proximity to the mote will change. The mote will have to update the regions that are in close proximity to the mote and get parameters and information vectors as needed.

Remark 3: The computational and communication costs to estimate a mote's location will depend on the number of regions and signal strengths. Again, the key parameter is r, the number of regions chosen. The number of regions should be chosen to balance the error rate, computational, and communication costs.

4. SIMULATIONS

Several simulations were conducted to test the performance of the centralized and distributed kernel regression localization algorithm. Sensors were placed on a 10 by 10 grid. There were $m = k^2$ base sensors placed on a grid and then the locations were perturbed by additive Gaussian noise. There were 400 motes placed randomly on the grid. The same signal strength models as [1] were used with additive noise that had deviation $\tau = 0.2$. For centralized base node regression the subspace algorithm was used with $3k$ support vectors. Fig. 1 shows how the mean localization error varies as the number of base sensors increases and Σ (signal strength parameter) is varied. Simulations were conducted 100 times for each setting with average curves and standard deviations shown. Fig. 2 shows similar plots for the distributed algorithm. Here the base nodes were partitioned using the k-Means algorithm with k regions and 7 support vectors used for each region. The error performance is very close to the centralized base node performance and actually give slightly better results when Σ is small. Computationally, distributed algorithm requires slightly more computations than the centralized method, but large savings can be realized if the base nodes perform computations in parallel. The distributed algorithm saves in communication costs as motes need to only get signal strength information from base nodes that are close to motes.

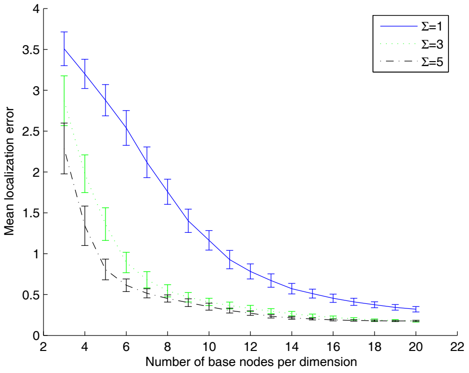

Fig. 1. Centralized base node regression with Gaussian kernels

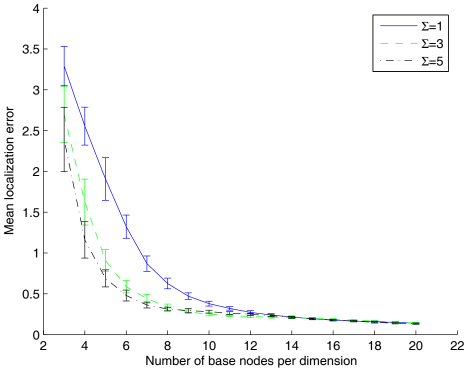

Fig. 2. Distributed regression with Gaussian kernels, $r = k$, 7 SV

5. SUMMARY AND FURTHER DIRECTIONS

This paper discusses a distributed kernel regression algorithm to perform sensor localization. The distributed algorithm can achieve similar performance to centralized algorithms while reducing communication costs and taking advantage of parallel processing capabilities of the sensor network. There are many further directions for this work including analyzing performance and determining optimal number of regions r. The distributed algorithms can also be used to get other information from the sensors and to track mobile sensors.

6. REFERENCES

[1] X. Nguyen, M. I. Jordan, and B. Sinopoli, "A kernel-based learning approach to ad hoc sensor network localization," *ACM Transactions on Sensor Networks*, vol. 1, no. 1, pp. 134–152, Aug. 2005.

[2] C. Zhu and A. Kuh, "Sensor network localization using pattern recognition and least squares kernel methods," in *Proceedings of 2005 Hawaii, IEICE and SITA Joint Conference on Information Theory, (HISC 2005)*,pp. 13–17, Honolulu, HI, USA, May, 2005.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[4] I. Stojmenovic, "Position-based routing in ad hoc networks," *IEEE Communications Magazine*, vol. 40, no. 7, pp. 128–134, July, 2002.

[5] S. Y. Seidel and T. S. Rappaport, "914Mhz path loss prediction models for indoor wireless communications in multifloored buildings," *IEEE Transactions on Antennas and Propagation*, vol. 40, no. 2, pp. 207–217, Feb. 1992.

[6] V. N. Vapnik, *Statistical learning theory*. John Wiley, New York City, NY, USA, 1998.

[7] A. Kuh, "Intelligent recursive kernel subspace estimation algorithms," in *Proceedings of the 39th annual Conference on Information Sciences and Systems (CISS 2005)*, pp. 216-221, Baltimore, MD, USA, Mar., 2005.

[8] J. A. Suykens, T. V. Gestel, J. D. Brabanter, B. D. Moor, and J. Vandewalle, *Least squares support vector machines*. World Scientific Publishing Co., Singapore, 2002.

[9] C. Zhu and A. Kuh, "Dynamic ad hoc network localization using online least squares kernel subspace methods," in *Proceedings of the 2006 IEEE International Symposium on Information Theory (ISIT 2006)*, pp. 630–634, Seattle, WA, USA, July, 2006.

[10] C. Guestrin, P. Bodi, R. Thibau, M. Paski, and S. Madde, "Distributed regression: an efficient framework for modeling sensor network data," in *IPSN '04: Proceedings of the third international symposium on Information processing in sensor networks*, ACM Press, pp. 1-10, Berkeley, CA, Apr., 2004.

[11] J. B. Predd, S.R. Kulkarni, H. V. Poor. Distributed Regression in Sensor Networks: Training Distributively with Alternating Projections, *Proceedings of the SPIE Conference and Advanced Signal Processing Algorithms, Architectures, and Implementations XV*, San Diego, CA, Aug., 2005.

[12] G. Jagannathan and R. Wright. "Privacy-Preserving Distributed k-Means Clustering over Arbitrarily Partitioned Data," *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2005.