Poster: Local Algorithms for Sensor Selection

Simon Shamoun
City University of New York
srshamoun@yahoo.com

Tianyi Tu
City University of New York
tty920723@gmail.com

Amotz Bar-Noy
City University of New York
amotz@sci.brooklyn.cuny.edu

Tarek F. Abdelzahe
University of Illinois,
Urbana-Champaign
zaher@illinois.edu

Abstract

In sensor networks, the sensor selection task is to activate only a subset, possibly a small subset, of the sensors while gaining as much utility as possible from the sensors. Most solutions are centralized algorithms with full knowledge of the network. We explore local algorithms for sensor selection. In these algorithms, each sensor independently decides if it should be included in the selection based on knowledge of its neighborhood alone. We design algorithms for increasing levels of knowledge in terms of the neighborhood size and demonstrate on randomly generated graphs representing sensor networks the improvement possible with more knowledge.

1 Introduction

The prospect of large scale sensor networks has driven research in efficient data collection. One way is to select a subset of sensors to predict the data of all other sensors [1, 2, 3]. Most prior work focuses on centralized algorithms that use information about the entire network. Our contribution is the design and study of local algorithms for sensor selection, in which each node independently decides if it should be part of the selected set only using information about nodes in its neighborhood. The goal is to maximize the prediction quality while keeping to the budget requirements. The advantage of this approach is that there is no overhead of centralized control. We study how the overall prediction quality can be improved by increasing the size of the neighborhood.

2 Model and Solutions

We model the sensor network as a graph of the *predictability* relationships between sensors. Each edge is assigned a weight from 0 to 1 indicating how well one node predicts another. A node is said to be *covered* by those predicting it. The level of coverage is defined by an aggregate function of the weights of the edges from the predicting nodes. A selected node fully covers itself. Let the weight of the edge from node v_i to v_j be w_{ij} and S be a set of nodes covering node v_j. We consider three commonly found aggregate functions [4]:

1. Maximum edge weight (MAX): $\max_{v_i \in S} w_{ij}$
2. Sum of all incoming edge weights (SUM), truncated at 1: $\min(1, \sum_{v_i \in S} w_{ij})$
3. Total probability (PROB): $1 - \prod_{v_i \in S}(1 - w_{ij})$. If edge weights represent the probability that two sensors detect the same event, then PROB is the probability that an event detected by a sensor is detected by at least one of the predicting sensors.

The optimization goal is to select k of n sensors with maximum coverage of all sensors in the network.

A naive solution when the entire predictability graph is known would be to order the nodes by the sum of their outgoing edge weights and select the top k nodes. The best possible approximation of the optimal solution is to select the nodes one by one, each time selecting the node that increases coverage by those already selected the most [1]. We refer to these as the "static" and "dynamic" greedy algorithms, respectively, and use them as lower and upper bounds on a good solution.

Local algorithms make decisions based on limited information about the predictability graph. We define the following hierarchy of knowledge available to sensors when making their decisions:

- **Local 0:** A sensor has no information about its neighbors.
- **Local 1.0:** The only information a sensor has is how it predicts its immediate neighbors.
- **Local 2.0:** As implied by the previous cases, Local 2.0 means a sensor knows how it predicts its neighbors, and how its neighbors predict their neighbors.

For the local algorithms, we assume that the weights of all edges are defined by some probability distribution, and that this distribution, the total number of sensors n, and the budget k, are known to all sensors. The general design of our algorithms is for a node to compute some combined value of the all the edge weights it knows. If the combined value is above some threshold, it randomly decides if it should be included in the selection. The thresholds and probabilities of being selected are set such that the expected number of nodes

International Conference on Embedded Wireless
Systems and Networks (EWSN) 2018
14–16 February, Madrid, Spain

ISBN: 978-0-9949886-2-1

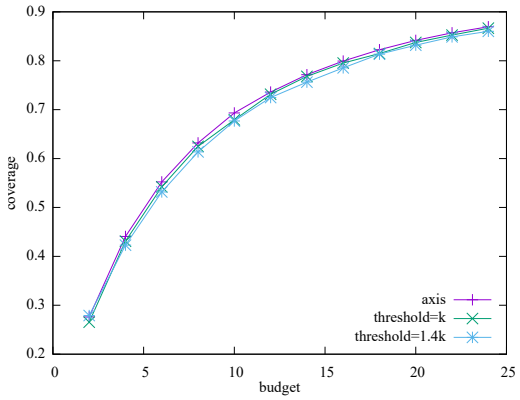

Figure 1. Coverage using various thresholds for the local1sum

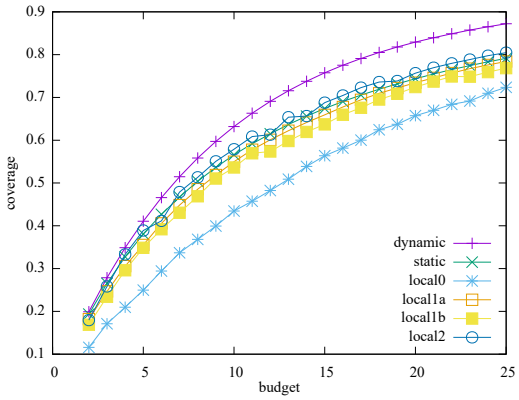

Figure 2. Coverage by local and global algorithms

to be selected equals the budget. We designed the following algorithms:

2.1 Local 0 (local random)

Since there is no information that can be used to make a decision, each node decides to be selected with probability k/n.

2.2 Local 1.0

For this level, we designed algorithms that are meant to favor nodes with higher aggregate functions of their outgoing edge weights, like the static greedy algorithm.

local1a Each node sums all of its outgoing edge weights. If the sum is greater than some threshold t, it adds itself to the selection with probability p.

local1b Each node counts the number of outgoing edges whose weight is above some pivot value s. If this number is greater than some threshold t, then select it with probability p.

2.3 Local 2.0

Each node ranks itself amongst its neighbors according to the sum of each nodes outgoing edge weights. It then calculates the ratio $r = b/l$ of how many of its neighbors it defeated, where b is the count of neighbors beaten and l is the total number of neighbor nodes. Then, as in the other algorithms, if r is above some threshold t, select it with probability t.

3 Evaluation

We analyzed these algorithms on Edros-Renyi and Barabasi-Albert graphs with 100 nodes whose edge weights were randomly assigned from either a uniform or decreasing or increasing Zipf distribution. We first determined that for all cases and algorithms, the best threshold to use is the one for which the expected number of selected nodes is as close to the budget k as possible. Figure 1 shows the coverage of a Erdos-Renyi graph by the local1 algorithm when the threshold is such that 1, 1.4, and 1.8 sensors are within that threshold and then randomly selected to meet the budget. It shows that 1 is the best value. Figure 2 compares the coverage by the different algorithms for the same graph. This shows that while local0, random selection, is far worse than static, the local1 algorithms are very close. Local2 is slightly better than static, showing that with more information and better algorithm design, it is possible to approach near-optimal solutions.

4 Conclusions

We defined a hierarchy of local algorithms for sensor selection and designed several algorithms for the first two levels. We demonstrated in simulations on random graphs that the random selection is not beneficial when unnecessary and that increasing the level can give better results. The goal is to see how much knowledge is necessary to achieve results as good as the global solution.

5 Acknowledgments

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

6 References

[1] C. C. Aggarwal, A. Bar-Noy, and S. Shamoun. On sensor selection in linked information networks. *Computer Networks*, 126:100–113, 2017.

[2] S. Joshi and S. Boyd. Sensor selection via convex optimization. *IEEE Trans. Signal Processing*, 57(2):451–462, 2009.

[3] A. Krause, A. P. Singh, and C. Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284, 2008.

[4] Y. Zou and K. Chakrabarty. Target localization based on energy considerations in distributed sensor networks. *Ad Hoc Networks*, 1(2-3):261–272, 2003.