Distributed estimation of dynamic functions

Shreevatsa R and Devavrat Shah

Laboratory for Information and Decision Systems

Massachusetts Institute of Technology

Applications/Motivation

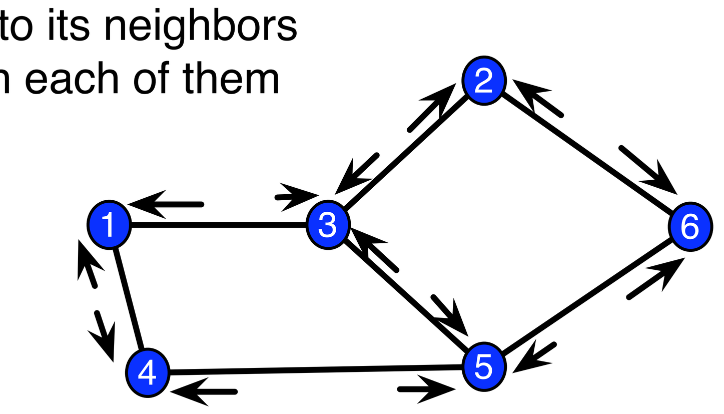
In some problems of control in wireless networks, sensor networks, etc., we want distributed algorithms that can maintain local estimates of a global function on the network, while the network itself is dynamically changing with time.

Problem statement

- Network with values at each node
- Values changing with time: $X_v(t)$
- Want distributed algorithm for maintaining estimates of (separable) functions of values, e.g. $\min_v(X_v(t))$, $\mathrm{sum}_v(X_v(t))$, …

Model

- Time is discrete
- Each node at each time talks to its neighbors
- Exchanges a real number with each of them



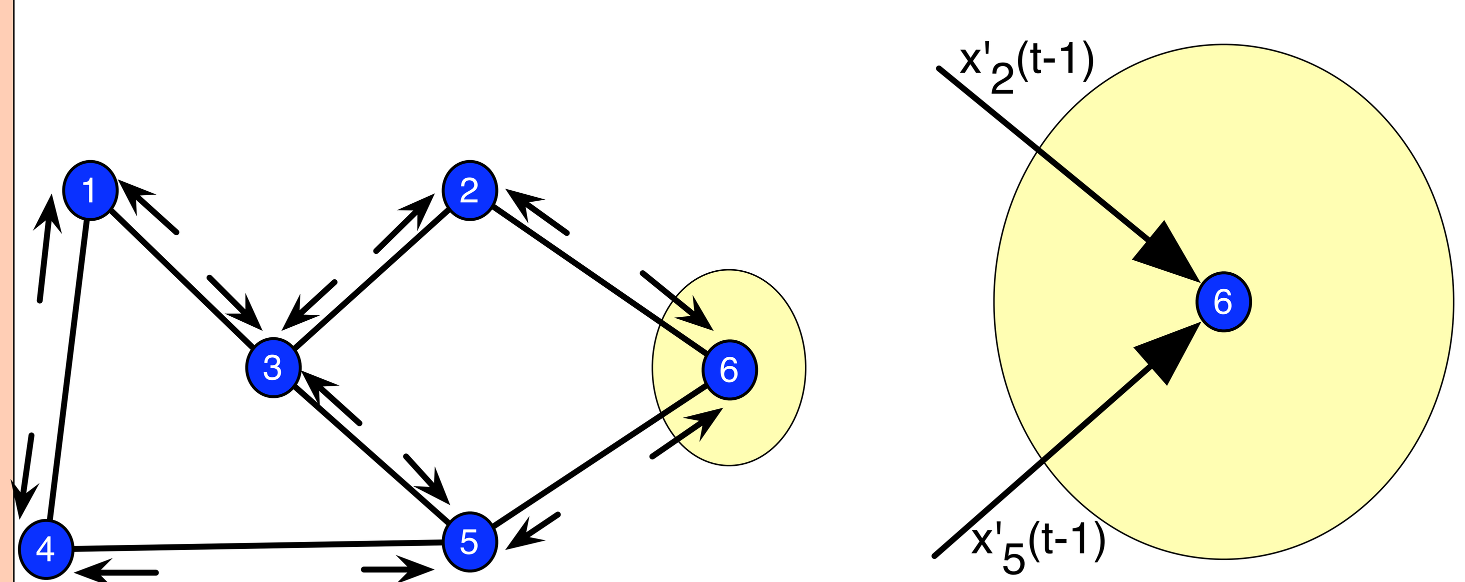
Minimum

Assuming $|X_v(t)-X_v(t-1)| \le \delta$

Algorithm

1. Each node exchanges its estimate $x'_v(t-1)$ with its neighbors
2. Each node sets its new estimate to be

$$x'_v(t) := \min(X_v(t),\ \delta + \min_u x'_u(t-1))$$



$x'_6(t) = \min(X_6(t), \delta + x'_2(t-1), \delta + x'_5(t-1))$

Theorem

$$0 \le x'_v(t) - \text{actual } \min(t) \le 2d\delta$$

where d is the diameter of the graph.

Similar algorithms and results with max instead of min, or multiplication instead of addition.

Sum

Assuming $1-\delta \le X_v(t)/X_v(t-1) \le 1+\delta$

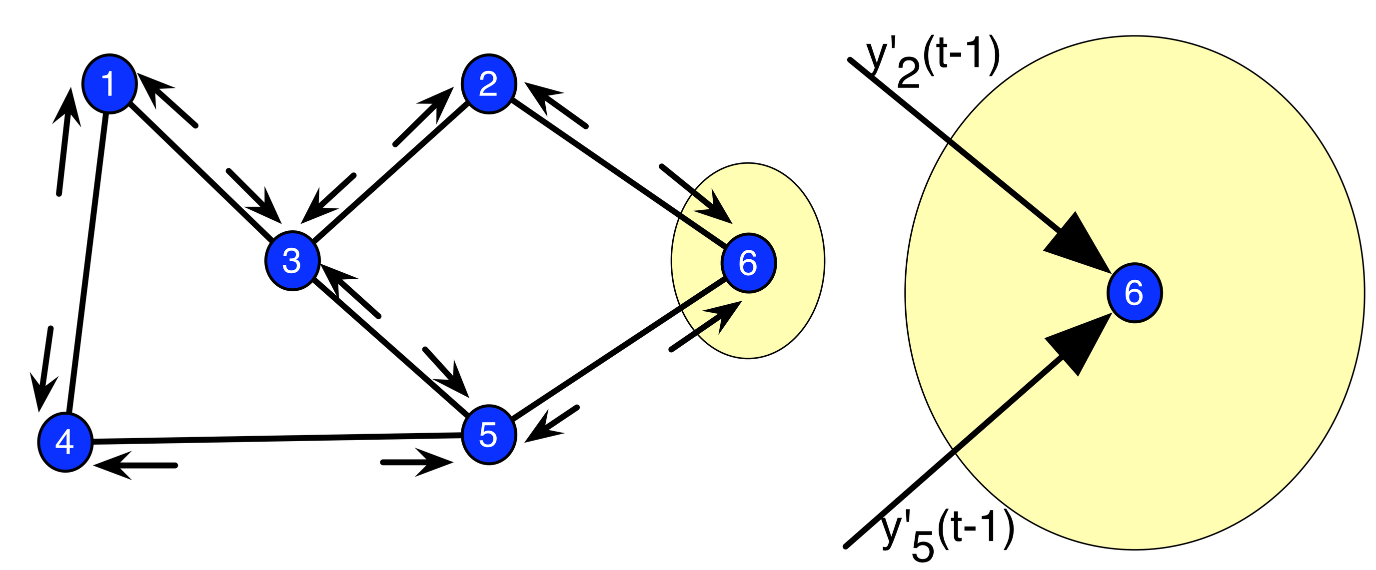
Algorithm

1. Each node v starts with random variables $Y_v(0)$
2. Random variables are (locally) updated with time as

$$Y_v(t+1) = Y_v(t)\, X_v(t) / X_v(t+1)$$

3. Use previous algorithm to maintain $\min Y_v(t)$

To deal with randomness (to get a bound w.h.p), run m copies of the above "in parallel", e.g. by taking turns.



$y'_6(t) = \min(Y_6(t), (1+\delta)y'_2(t-1), (1+\delta)y'_5(t-1))$

Theorem

- $Y_v(t)$ is distributed exponentially with mean $X_v(t)$
- (Estimate got using above algorithm)/(actual sum)
- is in $(1-2\varepsilon dm\delta, 1+2\varepsilon dm\delta)$ with probability 1-p, where $m = O(\ln(1/p) / \varepsilon^2)$

Sum, contd.

Lower bound

There is a matching lower bound up to the factor of $m\varepsilon$:
Any distributed algorithm cannot guarantee better than $d\delta$

Probabilistic communication

The results can be extended to the model where each node contacts one another node with some probability distribution. In this case, the diameter is replaced by the percolation time in the network.

Changing network

For the case where the network is also changing with time, i.e. nodes may enter and leave, matching lower bounds to show that this algorithm is close to optimal.

Conclusion

Gave simple distributed algorithms for maintaining
* min, max under additive, multiplicative errors
* sum under multiplicative errors

Future work

Other separable functions, tight lower bounds.