Breathe before Speaking: Efficient Information Dissemination despite Noisy, Limited and Anonymous Communication

Amos Korman

In collaboration with:
Bernhard Haeupler, Microsoft
Ofer Feinerman, Weizmann Institute

Inspiration

A recent experiment on recruitment by ants.
[Razin et al. Journal of the Royal Society Interface. 2013].

Inspiration

A recent experiment on recruitment by ants.
[Razin et al. Journal of the Royal Society Interface. 2013].

Some mysteries from the experiment

- Why is the recruiting process so slow and seemingly inefficient?

Inspiration

A recent experiment on recruitment by ants.
[Razin et al. Journal of the Royal Society Interface. 2013].

Some mysteries from the experiment

- Why is the recruiting process so slow and seemingly inefficient?
- Why is it only the recruiting ant that is doing all the work?

Possible difficulties that ants may face

- Communication is very limited in its vocabulary
  (what do you mean when you bump into me?)

Possible difficulties that ants may face

- Communication is very limited in its vocabulary
  (what do you mean when you bump into me?)

- Communication is noisy
  (are you really bumping into me or are you just happy to see me?)

Possible difficulties that ants may face

- Communication is very limited in its vocabulary
  (what do you mean when you bump into me?)
- Communication is noisy
  (are you really bumping into me or are you just happy to see me?)
- Stochastic and anonymous meetings
  (you look familiar, did we meet before?)

Possible difficulties that ants may face

- Communication is very limited in its vocabulary
  (what do you mean when you bump into me?)
- Communication is noisy
  (are you really bumping into me or are you just happy to see me?)
- Stochastic and anonymous meetings
  (you look familiar, did we meet before?)

Want to study:

Limited, Noisy and Stochastic communication

The typical assumption: messages are NOT distorted

Researchers in DC study node-failures (crash or Byzantine) and message-crashes, but noise in messages is typically ignored.

Distributed computing and Noise in communication

The typical assumption: messages are NOT distorted

Researchers in DC study node-failures (crash or Byzantine) and message-crashes, but noise in messages is typically ignored.

Why?

When bandwidth is not a big issue, employ error correction.

Error correction is problematic in restricted and stochastic communication

Error correction is problematic in restricted and stochastic communication

- When message size is restricted, redundancy comes at a price of limiting vocabulary.

Error correction is problematic in restricted and stochastic communication

- When message size is restricted, redundancy comes at a price of limiting vocabulary.
- Repeatedly talking to the same person is difficult in stochastic and anonymous meeting patterns.

Rumor spreading in Computer science: a classic setting

Initially

A complete network with n nodes. One source node s has a message m to be delivered to all nodes.

Rumor spreading in Computer science: a classic setting

Initially

A complete network with n nodes. One source node s has a message m to be delivered to all nodes.

The push model

- Synchronous model
- At each round, each node with the message m contacts another node, chosen uniformly at random, and delivers it the message.

Rumor spreading in Computer science: a classic setting

Initially

A complete network with n nodes. One source node s has a message m to be delivered to all nodes.

The push model

- Synchronous model
- At each round, each node with the message m contacts another node, chosen uniformly at random, and delivers it the message.

Complexities

- Time: $\Theta(\log n)$ rounds
- Total number of messages sent: $\Theta(n \log n)$
- Good against crash faults.

The noisy rumor spreading problem

The problem

A source node s has a bit $B \in \{0, 1\}$ that needs to be delivered to all nodes with high probability.

The Flip model of communication

- At each round, each agent u contacts another agent v, chosen uniformly at random, and chooses whether or not to deliver it a bit message $b \in \{0, 1\}$.
- With probability at most $1/2 - \epsilon$, the bit b is flipped and v receives $\bar{b}$.

Synchronization assumptions

- Each agent can count rounds.
- Global clock: all agents start with their clock set to zero (assumption can be removed with some price).

Some basic strategies: what to do when you receive a message?

Strategy 1. Wait to receive another message before sending

Some basic strategies: what to do when you receive a message?

Strategy 1. Wait to receive another message before sending

The problem: The time until the first agent hears two messages (from the source) is $\Omega(\sqrt{n})$ (birthday paradox).

Some basic strategies: what to do when you receive a message?

Strategy 1. Wait to receive another message before sending

The problem: The time until the first agent hears two messages (from the source) is $\Omega(\sqrt{n})$ (birthday paradox).

Strategy 2. Immediately send your opinion

Some basic strategies: what to do when you receive a message?

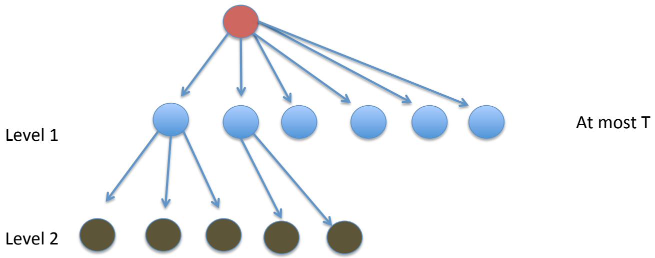
Strategy 1. Wait to receive another message before sending

The problem: The time until the first agent hears two messages (from the source) is $\Omega(\sqrt{n})$ (birthday paradox).

Strategy 2. Immediately send your opinion

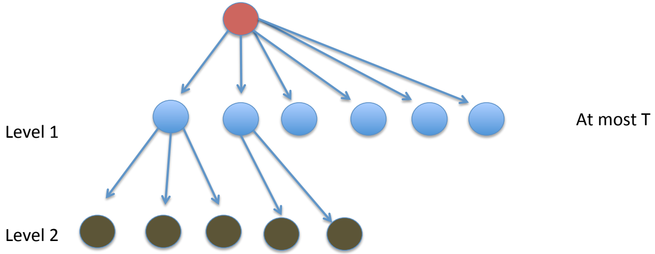
The problem: The quality of messages quickly deteriorates

A closer look



- In time T at most T agents heard directly from the source.

A closer look



- In time T at most T agents heard directly from the source.
- Most agents received a second hand rumor (at least).
  Hence the agents on level 2 will dominate the spreading.

What happens at level 2?

Probability of correct

$$(1/2+\epsilon)(1/2+\epsilon)+(1/2-\epsilon)(1/2-\epsilon)=1/2+2\epsilon^2.$$

What happens at level 2?

Probability of correct

$$(1/2+\epsilon)(1/2+\epsilon)+(1/2-\epsilon)(1/2-\epsilon)=1/2+2\epsilon^2.$$

For level *i*

Probability of correct is roughly $1/2+\epsilon^i$.

So quality of messages quickly deteriorates

Observation

The exists a simple protocol that runs in $O(\log n)$ *rounds no matter how small is* ϵ*.*

Our results

Theorem
$\exists$ a simple symmetric protocol running in $O(\frac{1}{\epsilon^2}\log n)$ rounds using $O(\frac{1}{\epsilon^2}n\log n)$ messages in total.

Our results

Theorem

$\exists$ a simple symmetric protocol running in $O(\frac{1}{\epsilon^2}\log n)$ rounds using $O(\frac{1}{\epsilon^2} n \log n)$ messages in total.

Observe: Each agent should receive $\Omega(\frac{1}{\epsilon^2}\log n)$ messages to be convinced even if these messages come directly from the source.

Our results

Theorem

∃ a simple symmetric protocol running in $O(\frac{1}{\epsilon^2}\log n)$ *rounds using* $O(\frac{1}{\epsilon^2} n \log n)$ *messages in total.*

Observe: Each agent should receive $\Omega(\frac{1}{\epsilon^2}\log n)$ messages to be convinced even if these messages come directly from the source. Hence:

- $\Omega(\frac{1}{\epsilon^2}\log n)$ rounds are required even to convince 1 agent, directly informed from the source.
- $\Omega(\frac{1}{\epsilon^2} n \log n)$ messages in total are required.

Phase 1: Spreading the information

Goal: Inform all agents, such that the fraction of agents with the correct opinion is at least $1/2 + 1/\sqrt{n}$.

Phase 1: Spreading the information

Goal: Inform all agents, such that the fraction of agents with the correct opinion is at least $1/2 + 1/\sqrt{n}$.

We want a good balance between:

- Slow deterioration of messages (short depth of tree), and
- Fast rumor spread (high depth of tree).

A first idea: delay messaging to control synchronization of levels

We want to have: **Level *i* agents do not spread their opinion before sufficiently many level *i* agents were informed.**

A first idea: delay messaging to control synchronization of levels

We want to have: **Level *i* agents do not spread their opinion before sufficiently many level *i* agents were informed.**

- Divide the time into *phases*. Phase i takes time $[T_i, T_i + \beta_i)$.
- If you receive a message (for the first time) in Phase i, wait until Phase $i + 1$ starts and only then start sending your opinion repeatedly.

A first idea: delay messaging to control synchronization of levels

We want to have: **Level i agents do not spread their opinion before sufficiently many level i agents were informed.**

- Divide the time into *phases*. Phase i takes time $[T_i, T_i + \beta_i)$.
- If you receive a message (for the first time) in Phase i, wait until Phase $i + 1$ starts and only then start sending your opinion repeatedly.

Property

If we have L_i agents awake when Phase i starts then we have $\beta_i \cdot L_i$ agents awake when Phase $i + 1$ starts.

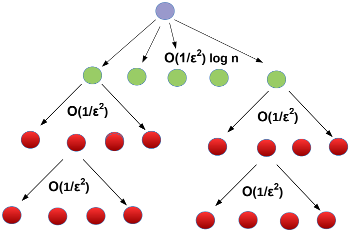
Setting β_i

Level 1: We want at least $O(\frac{1}{\epsilon^2}\log n)$ agents of level 1, to make sure that w.h.p, the majority of those have the correct opinion. So let $\beta_1 \approx \frac{1}{\epsilon^2}\log n$.

Setting β_i

Level 1: We want at least $O(\frac{1}{\epsilon^2}\log n)$ agents of level 1, to make sure that w.h.p, the majority of those have the correct opinion. So let $\beta_1 \approx \frac{1}{\epsilon^2}\log n$.

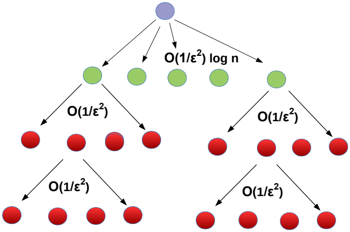
Level i, $i > 1$: Recall, if $1/2 + \delta_i$ fraction is correct on level i then $1/2 + \delta_i \cdot \epsilon$ fraction is correct on level $i + 1$. Set $\beta_i = \beta = O(\frac{1}{\epsilon^2})$ (degree $\approx \frac{1}{\epsilon^2} \gg$ inverse of the deterioration factor $\approx \epsilon$).



Setting β_i

Level 1: We want at least $O(\frac{1}{\epsilon^2} \log n)$ agents of level 1, to make sure that w.h.p, the majority of those have the correct opinion. So let $\beta_1 \approx \frac{1}{\epsilon^2} \log n$.

Level i, $i > 1$: Recall, if $1/2 + \delta_i$ fraction is correct on level i then $1/2 + \delta_i \cdot \epsilon$ fraction is correct on level $i + 1$. Set $\beta_i = \beta = O(\frac{1}{\epsilon^2})$ (degree $\approx \frac{1}{\epsilon^2} \gg$ inverse of the deterioration factor $\approx \epsilon$).



Time complexity: Total number of phases is $O(\log_\beta n)$. So total # rounds is $\beta_1 + \beta \log_\beta n = O(\frac{1}{\epsilon^2} \log n)$.

A (slow) deterioration of opinions

At phase i fraction of correct agents is at least $\approx 1/2 + \epsilon^i$.

A (slow) deterioration of opinions

At phase i fraction of correct agents is at least $\approx 1/2 + \epsilon^i$.

phases is $m \leq \log_\beta n = \log_{\frac{1}{\epsilon^2}} n = \log_\epsilon(1/\sqrt{n})$,

so the final fraction of correct agents is:

$$\geq 1/2 + \epsilon^m \geq 1/2 + \epsilon^{\log_\epsilon(1/\sqrt{n})} = 1/2 + 1/\sqrt{n},$$

as desired.

Second stage: boosting the faction of correct agents

Note: we start with a very small bias towards the correct opinion: $1/2 + 1/\sqrt{n}$.

In such a case, even without noise, the task of boosting the majority opinion is non-trivial. E.g., # of samples each agent should get from such a population should be higher than n.

Second stage: boosting the faction of correct agents

Note: we start with a very small bias towards the correct opinion: $1/2 + 1/\sqrt{n}$.

In such a case, even without noise, the task of boosting the majority opinion is non-trivial. E.g., # of samples each agent should get from such a population should be higher than n.

An $O(\log n)$ time majority boosting algorithm exists [Angluin, Aspnes, and Eisenstat, DISC 2007]. However, this algorithm uses messages of size 2 bits (rather than 1) and does NOT account for noise in messages.

[Doerr et al SPAA 2011] show that a method based on gradual boosting the majority can achieve consensus in $O(\log n)$ time. We show that a similar approach works, also in the presence of noise.

The idea: gradual boosting

- Divide the time into phases. Up to the last phase, all phases consist of $\gamma = O(1/\epsilon^2)$ rounds. In each phase:

The idea: gradual boosting

- Divide the time into phases. Up to the last phase, all phases consist of $\gamma = O(1/\epsilon^2)$ rounds. In each phase:
  - Send γ times your current opinion,
  - Receive γ opinions. Set your opinion to the majority opinion among those γ opinions.

The idea: gradual boosting

- Divide the time into phases. Up to the last phase, all phases consist of $\gamma = O(1/\epsilon^2)$ rounds. In each phase:
  - Send γ times your current opinion,
  - Receive γ opinions. Set your opinion to the majority opinion among those γ opinions.

After $O(\log n)$ of such phases, fraction of correct agents is $1/2 + \textit{constant}$

The idea: gradual boosting

- Divide the time into phases. Up to the last phase, all phases consist of $\gamma = O(1/\epsilon^2)$ rounds. In each phase:
  - Send γ times your current opinion,
  - Receive γ opinions. Set your opinion to the majority opinion among those γ opinions.

After $O(\log n)$ of such phases, fraction of correct agents is $1/2 + \textit{constant}$

Then, one last phase of length $O(\frac{1}{\epsilon^2} \cdot \log n)$ where each agent is sending its opinion in each round, and at the end taking majority guarantees that all agents have the correct opinion with high probability.

Gradual boosting- a closer look

Let δ_i be such that $1/2 + \delta_i$ is the fraction of correct agents at phase i. (Note $\delta_1 > 1/\sqrt{n}$).

Theorem

- *As long as δ_i is smaller than some constant c_1, we have $\delta_{i+1} \geq 2\delta_i$.*
- *If $\delta_i > c_1$, then δ_{i+1} is greater than another constant $c_2 < c_1$.*

Note that since δ_i maybe very small we cannot use Chernoff directly to obtain the theorem!

Gradual boosting- a closer look

Let δ_i be such that $1/2 + \delta_i$ is the fraction of correct agents at phase i. (Note $\delta_1 > 1/\sqrt{n}$).

Theorem

- *As long as* δ_i *is smaller than some constant* c_1*, we have* $\delta_{i+1} \geq 2\delta_i$*.*
- *If* $\delta_i > c_1$*, then* δ_{i+1} *is greater than another constant* $c_2 < c_1$*.*

Note that since δ_i maybe very small we cannot use Chernoff directly to obtain the theorem!

Corollary

After $O(\log n)$ *phases (which is* $O(\frac{1}{\epsilon^2} \log n)$ *time), the fraction of correct agents is at least* $1/2 + c_2$*.*

Removing the global clock assumption

So far we assumed all agents wake up at time 0. What about if agents do not have the same starting time?

Removing the global clock assumption

So far we assumed all agents wake up at time 0. What about if agents do not have the same starting time?

First note, if all clocks are initially in the range $[0, D]$,
We can use the synchronized push model to synchronize agents:

Conclusion

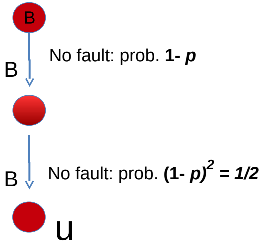
Delaying propagation of messages, relying on synchronizing, and taking majority of samples, allows to overcome highly stochastic, anonymous, and noisy settings.

Open problems

- What about if the synchronization is very bad?
- Our time complexity is polylogarithmic. In case an adversary controls the content of the faulty message, can we prove a polynomial lower bound?
- Different graph families...

Adversary model: What happens at level 2?

Assume that an **adversary** controls the content of faulty messages. Assume $p = 1 - \frac{1}{\sqrt{2}} \approx 0.3$.



With prob 1/2 the first message at u is ``clean" (u receives B)

With prob 1/2 at least 1 fault. Adversary makes u receive $\overline{B}$

Messages received at level 2 nodes are uniformly spread between 0 and 1

Noise in communication: A new dimension to fault tolerance in distributed computing

Distortion in communication is interesting to study especially when message size is limited, and one does not have full control over its interaction (dynamic/stochastic meeting patterns).

Noise in communication: A new dimension to fault tolerance in distributed computing

Distortion in communication is interesting to study especially when message size is limited, and one does not have full control over its interaction (dynamic/stochastic meeting patterns).

We encourage you to study such noisy models:

Noise in communication: A new dimension to fault tolerance in distributed computing

Distortion in communication is interesting to study especially when message size is limited, and one does not have full control over its interaction (dynamic/stochastic meeting patterns).

We encourage you to study such noisy models:

- Discrete noise: E.g., the flip model of communication.

Noise in communication: A new dimension to fault tolerance in distributed computing

Distortion in communication is interesting to study especially when message size is limited, and one does not have full control over its interaction (dynamic/stochastic meeting patterns).

We encourage you to study such noisy models:

- Discrete noise: E.g., the flip model of communication.
- Continuous distortion: A message is a real number. If a message is sent as x then the received message is $x + n$, where n is sampled from some continuous noise distribution.

Thank you!