Gossip and Knowledge

Hans van Ditmarsch
Open University of the Netherlands

Malvin Gattinger
University of Amsterdam

- Gossip terminology around calls
- Epistemic, symmetric, and distributed gossip protocols

presentation for ESSLLI Galway

August 2022

Gossip terminology — call

A is the set of agents or callers.

A gossip graph G is a triple (A, N, S) where $N \subseteq \mathcal{P}(A \times A)$ is the neighbour relation and $S \subseteq \mathcal{P}(A \times A)$ is the secret relation.

If $N = A \times A$ (all agents can call each other), the gossip graph is *complete*. We also call the gossip graph a secret distribution. The *initial secret distribution* $\boldsymbol{I}$ is the triple (A, A^2, I).

A call or telephone call is a pair from $A \times A$. For $(a, b) \in A \times A$ we write ab, and we require $a \neq b$. We say that a and b are *involved* in call ab, that a is the *caller*, and b the *callee*.

Write S_a for $\{b \in A \mid (a, b) \in S\}$. If $S_a = A$, agent a is an expert.

Gossip terminology — semantics of a call

Given gossip graph (A, N, S).

- **pushpull:** The result of applying call ab is the gossip graph (A, N, S^{ab}), where $S^{ab} = S \cup (\{(a, b), (b, a)\} \circ S)$.
  a and b learn each other's secrets

Variants (not varying the notation)

- **push:** The result of making call ab is the gossip graph (A, N, S^{ab}), where $S^{ab} = S \cup (\{(b, a)\} \circ S)$.
  b learns the secrets of a
- **pull:** The result of making call ab is the gossip graph (A, N, S^{ab}), where $S^{ab} = S \cup (\{(a, b)\} \circ S)$.
  a learns the secrets of b
- **dynamic pushpull:** The result of call ab is the gossip graph (A, N^{ab}, S^{ab}), where $N^{ab} = N \cup (\{(a, b), (b, a)\} \circ N)$ and $S^{ab} = S \cup (\{(a, b), (b, a)\} \circ S)$.
  a and b learn each other's secrets and neighbours

Gossip terminology — call sequence

A call sequence is inductively defined as: ϵ is a call sequence, if σ is a call sequence and ab is a call, then $\sigma.ab$ is a call sequence.

We write (all with obvious inductive definitions) :

- $|\sigma|$ to denote the length of a call sequence
- $\sigma[i]$ for the ith call of the sequence
- $\sigma|i$ for the first i calls of the sequence
- σ_a for the *view* of σ by a (subsequence of calls involving a)
- $\widetilde{\sigma}_a$ for the *asynchronous full view* of σ by a, defined as:

$$\begin{aligned}
\widetilde{\epsilon}_a &:= \epsilon \\
\widetilde{\sigma.bc}_a &:= \widetilde{\sigma}_a \\
\widetilde{\sigma.ab}_a &:= (\widetilde{\sigma}_a, \widetilde{\sigma}_b).ab \\
\widetilde{\sigma.ba}_a &:= (\widetilde{\sigma}_b, \widetilde{\sigma}_a).ba
\end{aligned}$$

synchronous f.v.: $\widetilde{\sigma.bc}_a := \widetilde{\widetilde{\sigma}}_a.\bullet$

Applying σ to a secret relation S: $S^\epsilon = S$; and $S^{\sigma;ab} = (S^\sigma)^{ab}$. Same for N. By G^σ, where $G = (A, N, S)$, we mean (A, N, S^σ).

Gossip terminology — observation relation

Given $a \in A$ and gossip graphs $G = (A, N, S)$, $H = (A, O, T)$.
The *asynchronous observation relation* $\sim_a$ is the smallest equivalence relation such that:

- $(G, \epsilon) \sim_a (H, \epsilon)$ if $N_a = O_a$ and $S_a = T_a$
- $(G, \sigma.bc) \sim_a (H, \tau)$ if $(G, \sigma) \sim_a (H, \tau)$ and $a \notin \{b, c\}$
- $(G, \sigma.ab) \sim_a (H, \tau.ab)$ and $(G, \sigma.ba) \sim_a (H, \tau.ba)$ if $(G, \sigma) \sim_a (H, \tau)$ and $S^\sigma_b = T^\tau_b$

The *synchronous observation relation* $\approx_a$ is the smallest equivalence relation such that:

- $(G, \epsilon) \approx_a (H, \epsilon)$ if $N_a = O_a$ and $S_a = T_a$
- $(G, \sigma.bc) \approx_a (H, \tau.de)$ if $(G, \sigma) \approx_a (H, \tau)$ and $a \notin \{b, c, d, e\}$
- $(G, \sigma.ab) \approx_a (H, \tau.ab)$ and $(G, \sigma.ba) \approx_a (H, \tau.ba)$ if $(G, \sigma) \approx_a (H, \tau)$ and $S^\sigma_b = T^\tau_b$

$(G, \sigma) \sim_a (H, \tau)$ implies $S^\sigma_a = T^\tau_a$; $(G, \sigma) \approx_a (H, \tau)$ implies $S^\sigma_a = T^\tau_a$.
The observation relation determines what an agent knows.

Gossip terminology — other observation relations

Recalling synchronous and asynchronous observation relation

- $(G, \sigma.bc) \sim_a (H, \tau)$ if $(G, \sigma) \sim_a (H, \tau)$ and $a \notin \{b, c\}$
- $(G, \sigma.bc) \approx_a (H, \tau.de)$ if $(G, \sigma) \approx_a (H, \tau)$ and $a \notin \{b, c, d, e\}$

Other observation relations

Agents observe all calls (as in the cup telephone illustration)

- $(G, \sigma.bc) =_a (H, \tau.bc)$ if $(G, \sigma) =_a (H, \tau)$, for any $b, c \in A$

Merge and inspect (agents see the output but not the input)

- $(G, \sigma.ab) \sim_a (H, \tau.ab)$ if $(G, \sigma) \sim_a (H, \tau)$ and $S^\sigma_b \cup S^\sigma_a = S^\tau_b \cup S^\tau_a$

Asymmetric observation (a sees b calling, but not the adressee)

- $(G, \sigma.bc) \approx_a (H, \tau.bd)$ if $(G, \sigma) \approx_a (H, \tau)$ and $a \notin \{b, c, d\}$

All you know (full-information protocol in distributed computing)

- $(G, \sigma) \sim_a^{\text{full}} (H, \tau)$ iff $N_a = O_a$, $S_a = T_a$, and $\widetilde{\sigma}_a = \widetilde{\tau}_a$

For $N_a = O_a$ and $S_a = T_a$ we can write $G \sim_a H$.

Epistemic and distributed protocols — gossip protocol

Let a gossip graph $G = (\mathrm{A}, N, S)$ and a call sequence σ be given.

- protocol condition for a call ab: a property P_{ab} (intentionally vague!) that can be determined with respect to G and σ, and that is *local* (a.k.a. epistemic) and *uniform* (a.k.a. symmetric).
- local (or epistemic): for all $\tau \sim_a \sigma$ (or $\tau \approx_a \sigma$), whenever P_{ab} holds for σ, then P_{ab} holds for τ. a knows P_{ab}
- uniform (or symmetric): whenever we simultaneously replace all a, b in P_{ab} by c, d, then we obtain P_{cd}. all are equal

A gossip protocol P is a non-deterministic algorithm:
While not all agents are experts and there are $a, b \in A$ with $a \neq b$ such that b is a neighbour of a and P_{ab}, choose $a, b \in A$ with $a \neq b$ such that b is a neighbour of a and P_{ab}, and execute call ab.

The second 'choose' means gossip protocols are also distributed.

Epistemic and distributed protocols — observation model

Given gossip graph G and call sequence σ, ab is P-permitted if P_{ab} (holds); ϵ is P-permitted on G; $\sigma.ab$ is P-permitted on G iff σ is P-permitted on G and ab is P-permitted on G after σ.

The extension $\mathrm{P}(G)$ of protocol P on G is the set of P-permitted call sequences on G. We write $\mathrm{P}(\mathcal{G}) \subseteq \mathrm{P}'(\mathcal{G})$ iff $\mathrm{P}(G) \subseteq \mathrm{P}'(G)$ for all $G \in \mathcal{G}$, and $\mathrm{P} \subseteq \mathrm{P}'$ if $\mathrm{P}(G) \subseteq \mathrm{P}'(G)$ holds for all G.

Most common: $\mathcal{G} = \{(A, A \times A, I)\}$ (the initial secret distribution).

The observation model consists of all pairs (G, σ) such that $G \in \mathcal{G}$ and σ is P-permitted, and with relations $(G, \sigma) \sim_a (H, \tau)$ $(\approx_a)$ and $(G, \sigma) \to_{ab} (G, \sigma.ab)$.

We informally also allow infinite call sequences.

Epistemic and distributed protocols — termination

Given G, a P-permitted call sequence σ is terminal iff no $\sigma.ab$ is P-permitted on G. A terminal call sequence σ is successful if after σ all the agents are experts.

- P is successful on G iff all terminal $\sigma \in \mathrm{P}(G)$ are successful.
- P is weakly successful on G iff there is a successful $\sigma \in \mathrm{P}(G)$.

P is (weakly) successful iff P is (weakly) successful on every G.

A P-permitted infinite call sequence σ^ω is *fair* if for all $a \neq b \in A$, if for all i there is $j > i$ such that call ab is P-permitted on G after $\sigma^\omega|j$, then for all i there is $j > i$ such that $\sigma^\omega[j] = ab$.

P is *fair* if all P-permitted infinite call sequences are unfair.

unsuccessful = not weakly successful
unfair = not fair

Distributed protocols — different formulations

While not all agents are experts and there are $a, b \in A$ with $a \neq b$ such that b is a neighbour of a and P_{ab}, choose $a, b \in A$ with $a \neq b$ such that b is a neighbour of a and P_{ab}, and execute call ab.

While the termination condition is not satisfied, choose $a, b \in A$ with $a \neq b$ such that the execution condition is satisfied, and execute call ab.

In distributed computing we do not want a central scheduler / environment. The distributed nature appears if we describe it as:

Each $a \in A$ runs the following ***a*-program**: choose $b \in A$ with $a \neq b$ such that the execution condition is satisfied, and execute call ab (or else fail). The environment ϵ runs the following **ϵ-program**: until the termination condition is satisfied, choose $a \in A$ and execute ***a*-program**.

Omit ϵ red part and establish stabilization instead of termination.

Example gossip protocols

$$
\begin{array}{lcl}
\mathrm{ANY}_{ab} & = & \top \\
\mathrm{CMO}_{ab} & = & ab, ba \notin \sigma \\
\mathrm{wCMO}_{ab} & = & ab \notin \sigma \\
\mathrm{LNS}_{ab} & = & (a,b) \notin S^{\sigma} \\
\mathrm{PIG}_{ab} & = & \exists \tau \sim_a \sigma, \exists c, (a,c) \in S_a^{\tau} \setminus S_b^{\tau} \text{ or } (a,c) \in S_b^{\tau} \setminus S_a^{\tau} \\
\mathrm{KIG}_{ab} & = & \forall \tau \sim_a \sigma, \exists c, (a,c) \in S_a^{\tau} \setminus S_b^{\tau} \text{ or } (a,c) \in S_b^{\tau} \setminus S_a^{\tau} \\
\mathrm{SPI}_{ab} & = & \text{spider: if } a \text{ calls } b, a \text{ gets the token (if any) from } b \\
\mathrm{TOK}_{ab} & = & \text{token: if } a \text{ calls } b, a \text{ hands her token to } b
\end{array}
$$

$$
\begin{array}{lcl}
\mathrm{ANY} & = & \text{any call is permitted} \\
\mathrm{CMO} & = & \text{after call } ab, a \text{ and } b \text{ may not call each other} \\
\mathrm{wCMO} & = & \text{after call } ab, a \text{ may not call } b \\
\mathrm{LNS} & = & a \text{ does not know ('hold') the secret of } b \\
\mathrm{PIG} & = & a \text{ considers possible that } a \text{ or } b \text{ will learn a secret} \\
\mathrm{KIG} & = & a \text{ knows that } a \text{ or } b \text{ will learn a secret} \\
\mathrm{SPI} & = & \text{token holders may make a call, and then keep their token} \\
\mathrm{TOK} & = & \text{token holders may make a call, and then lose their token}
\end{array}
$$

Gossip protocol hierarchy