Trust in Multiagent Systems

Pınar Yolum
Email: p.yolum@uu.nl

Department of Information and Computing Sciences
Utrecht University

Referral systems

- Each agent looking for a service provider for a service need
- Agents differ in what they provide in terms of service
- Some agents are more knowledgeable in what others can provide and thus can give advice

Agent components

- Each agent has an interest, expertise, and a sociability.
- Interest: What kind of information it is looking for
- Expertise: What kind of queries it can answer (e.g., know-how)
- Model of others
  - How it perceives others in terms of expertise
  - How likely it is to know others with the right answers (*sociability*)
  - Trust is Expertise with Sociability. The higher it perceives expertise and sociability of another agent, the more it trusts.

Agent actions

An agent can

- Generate a query (based on its interest) and ank others for an answer.
- Answer a query (if it has *sufficient* expertise)
- Provide a *referral* to a set of other agents who are likely to answer the query

The agent who receives the referrals is free to follow them.

Main agent cycle

```
Generate query
Compute a list of matching neighbors
Send query to matching neighbors
while (!timeout) do
    Receive message
    if (message.type == referral) then
        Send query to referred agent
    else
        Add answer to answerset
    end if
end while
for $i = 1$ to $|answerset|$ do
    Evaluate answer($i$)
    Update agent models
end for
```

Representation

- Query, Interest, Expertise: Represented as vectors
  - Inspired from Vector Space Model, which is mainly used for text documents
  - Each dimension corresponds to a different term/domain
  - Each dimension between 0 and 1
- Sociability: Scalar

Example

- Three domains: CS, Math, Law
- Expertise of Agent A is $[0.4, 0.8, 0.9]$, where expertise in CS is 0.4, expertise in Math is 0.8, and expertise in Law is 0.9
- Interest of Agent A is $[0.9, 0.6, 0.1]$, where interest in CS is 0.4, interest in Math is 0.8, and interest in Law is 0.9
- Agent B might model this differently based on its interaction with A

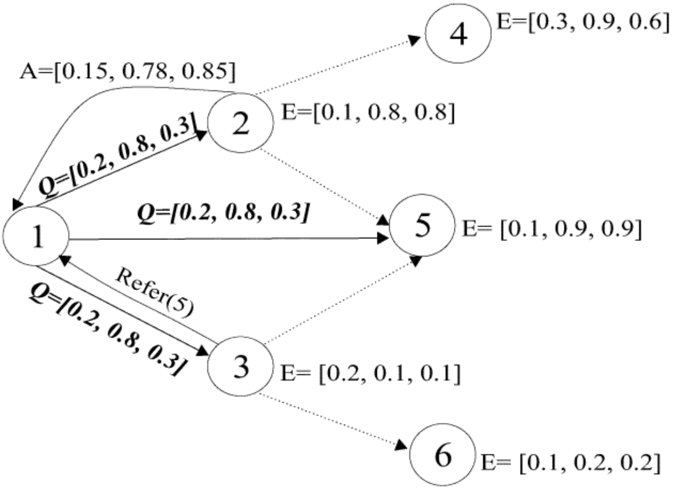
Network

- Each agent is neighbors with a subset of the network
- For each neighbor, the agent keeps a *model* of expertise and sociability
- Compute matching neighbors: Directs service requests based on how *capable* the agent is to answer the queries

$$Q \otimes E = \frac{\sum_{t=1}^{n} (q_t e_t)}{\sqrt{n \sum_{t=1}^{n} q_t{}^2}} \tag{1}$$

- The above uses the expertise in the model as the actual expertise is not known to the querying agent
- The service provider can provide an answer or give a referral to another service provider

Example (Agent 1 looking at the network)



Update models

The service received is evaluated:

- The agent that provided the service has contributed by expertise, hence the modeled expertise is updated.
- The agents who have provided referrals to find the provider have contributed with their sociability, hence their modeled sociability is updated.

Policies

- Referral policy: Under what conditions will a referral be given
- The answer or referral is evaluated by the service consumer and the models are updated.
- Neighbor selection policy: Based on the updated models, agents can choose different neighbors, though the set size is fixed.

Referral policies

- Refer all neighbors: Does not consider which neighbors would be more likely to answer.
- Refer all matching neighbors:
  - Calculate how capable each neighbor is in answering a particular query.
  - Refer those neighbors with “sufficient” expertise (those greater than a given threshold T).
- Refer best neighbor:
  - Refer the most capable neighbor.
  - Guarantees that at least one neighbor is referred.

Neighbor selection policies

- Consider weighted sociability (W) and expertise.
  - Weighted Average (W=0.25, 0.50, 0.75): Choose the best *m* based on weighing both the expertise and the sociability of the acquaintances.
  - Providers (W=0): Choose the best m agents whose expertise matches the agent's interests.
  - Sociables (W=0.90, 1): Choose the most sociable m agents

Measuring performance

- Quality: How capable are my neighbors?

$$\frac{I_i \otimes E_j}{path(i,j)} \tag{2}$$

- Clustering: How close am I to the agents that are similar?

$$I_i \oplus I_j = \frac{e^{-\|I_i - I_j\|^2} - e^{-n}}{1 - e^{-n}} \tag{3}$$

$$\gamma(i) = \frac{\sum_{(u,v)\in M_i} I_u \oplus I_v}{|V_i|(|V_i| - 1)} \tag{4}$$

Evaluation

- Methodology: Multiagent simulations
- Setup:
  - 400 agents, with 5 to 25% service providers
  - 30 queries generated based on agents' interests
  - Agents produce answers based on their expertise vector, mostly by perturbation
  - 4 to 8 neighbors per agent; initial neighbors are random
  - 10 neighbor changes

Observations from experiments

- Exchanging more referrals does not guarantee that the quality of the network will be high. The topology of the network can prevent consumers from locating some of the service provider.
- When more referrals are exchanged
  - Better providers are found
  - Some providers emerge as authorities (e.g., most people use them)
- Agents with similar interests
  - Have a tendency to be kept as neighbors.
  - But find the same providers most of the time.
  - Some providers are never reached.
- Agents emerge as authorities only if they have high expertise

Take-home messages

- Referrals are useful to propagate information about service providers
- By building models of others and updating them over time, agents obtain an accurate image of others
- Policies can be used for various activities and can be different from agent to agent
- Designing metrics is important to measure various aspects (e.g., performance, clustering)