The Advisor-POMDP

A principled approach to trust through reputation in electronic markets

Kevin Regan, Robin Cohen & Pascal Poupart
School of Computer Science
University of Waterloo

Motivation

Reputation systems are the worst way of building trust on the Internet, except for all those other ways that have been tried from time-to-time

-Paul Resnick *by way of Winston Churchill*

Electronic Markets

- Internet Buying and Selling Agents
  - Buyer requests a good
  - Potential sellers submit bids
  - Buyer selects best seller
- Assume sellers are
  - Self-interested
  - Able to vary quality of goods

Goal

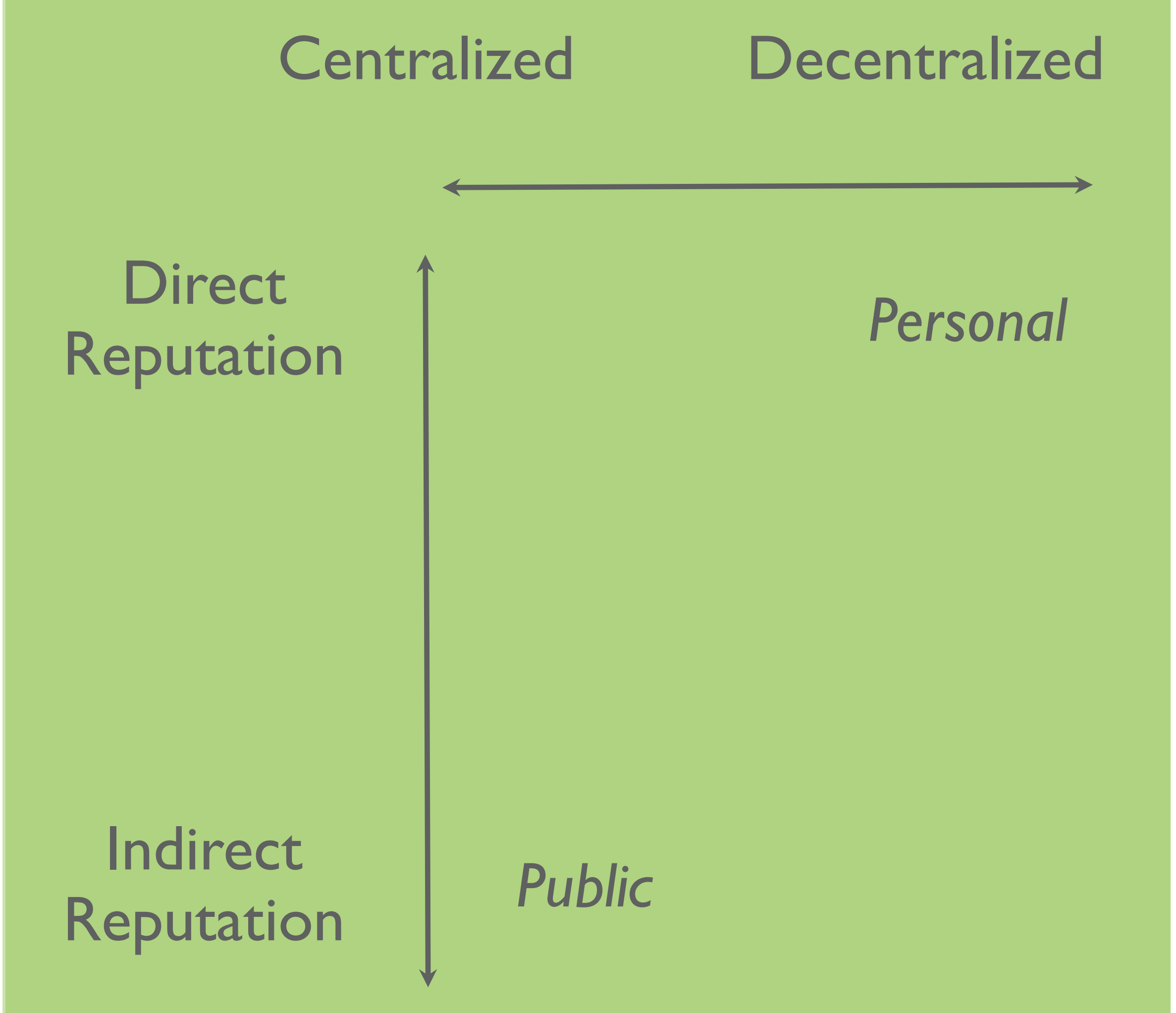
- Design an adaptive buying agent that makes effective purchase decisions
-
- Using information from other buyers to model the reputation of the seller

Outline

- Overview of Reputation Systems
- Some methods for representing reputation
- A decision theoretic framework for gathering and acting on reputation information using POMDPs
- An example illustrating an agent's beliefs about seller reputation are updated in the Advisor-POMDP
- Some conclusions and future directions

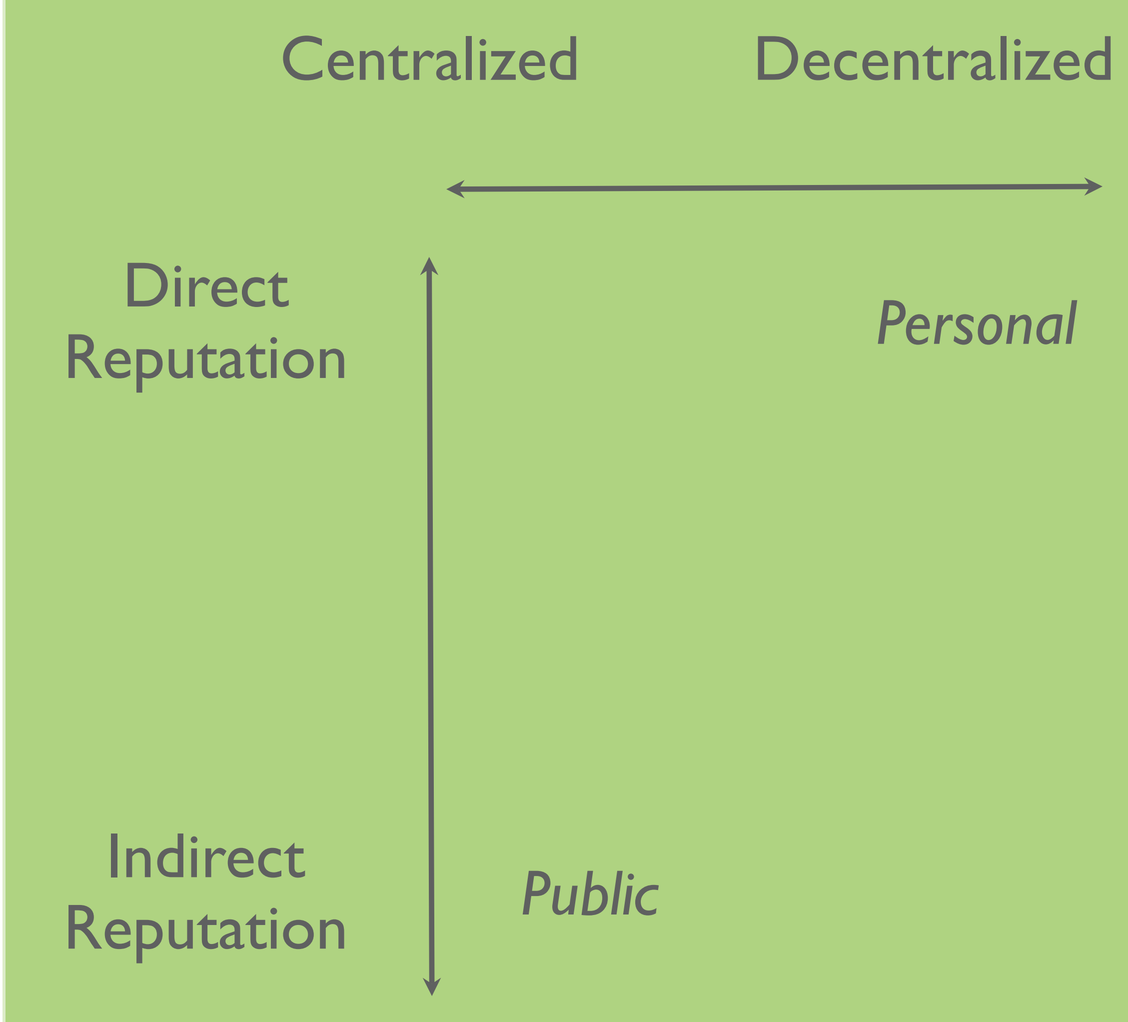
Reputation Models

- Public
  - Central service collects feedback and publishes a global seller reputation
- Personal
  - Each buyer uses only their own past interactions to build seller reputation



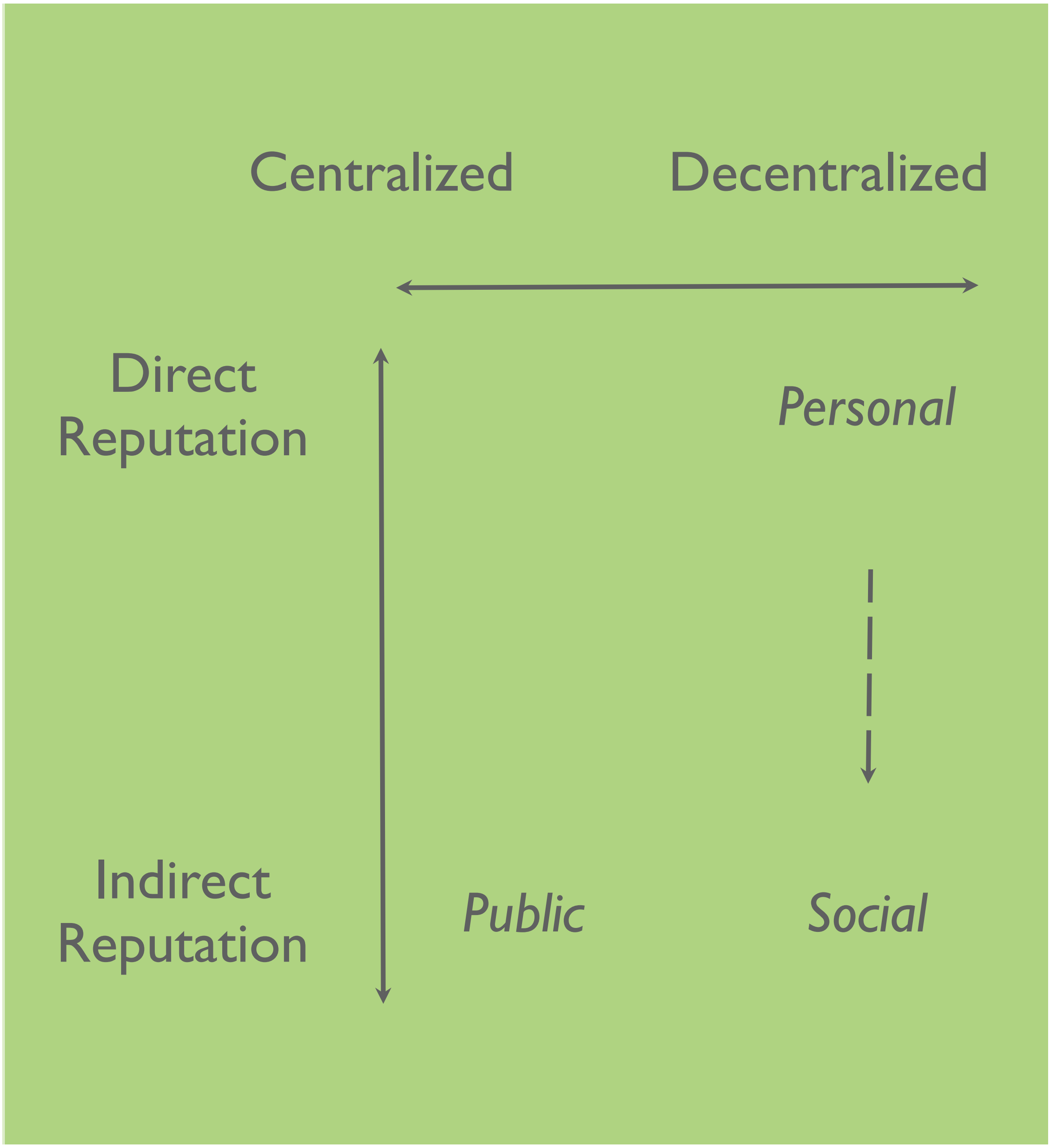
Reputation Models

- Social
  - Buyer uses past interactions *and* indirect information from other buyers to model seller reputation



Reputation Models

- Social
  - Buyer uses past interactions *and* indirect information from other buyers to model seller reputation



Challenges for Social Model

- *How to represent reputation?*
  - *Need to model our knowledge about the likelihood of being satisfied with a seller*
- *How do we gather and use reputation?*
  - *When to ask other buyers and when to decide to make purchase*

Classes of Uncertainty

- Stochastic Uncertainty
  - Due to randomness of the system
- Epistemic Uncertainty
  - Due to lack of knowledge about the randomness of the system

Purchases		Reputation [0,1]
Satisfied	Unsatisfied	
8	2	0.8

Classes of Uncertainty

- Stochastic Uncertainty
  - Due to randomness of the system
- Epistemic Uncertainty
  - Due to lack of knowledge about the randomness of the system

Epistemic Uncertainty	Purchases		Reputation [0,1]
	Satisfied	Unsatisfied	
High	8	2	0.8

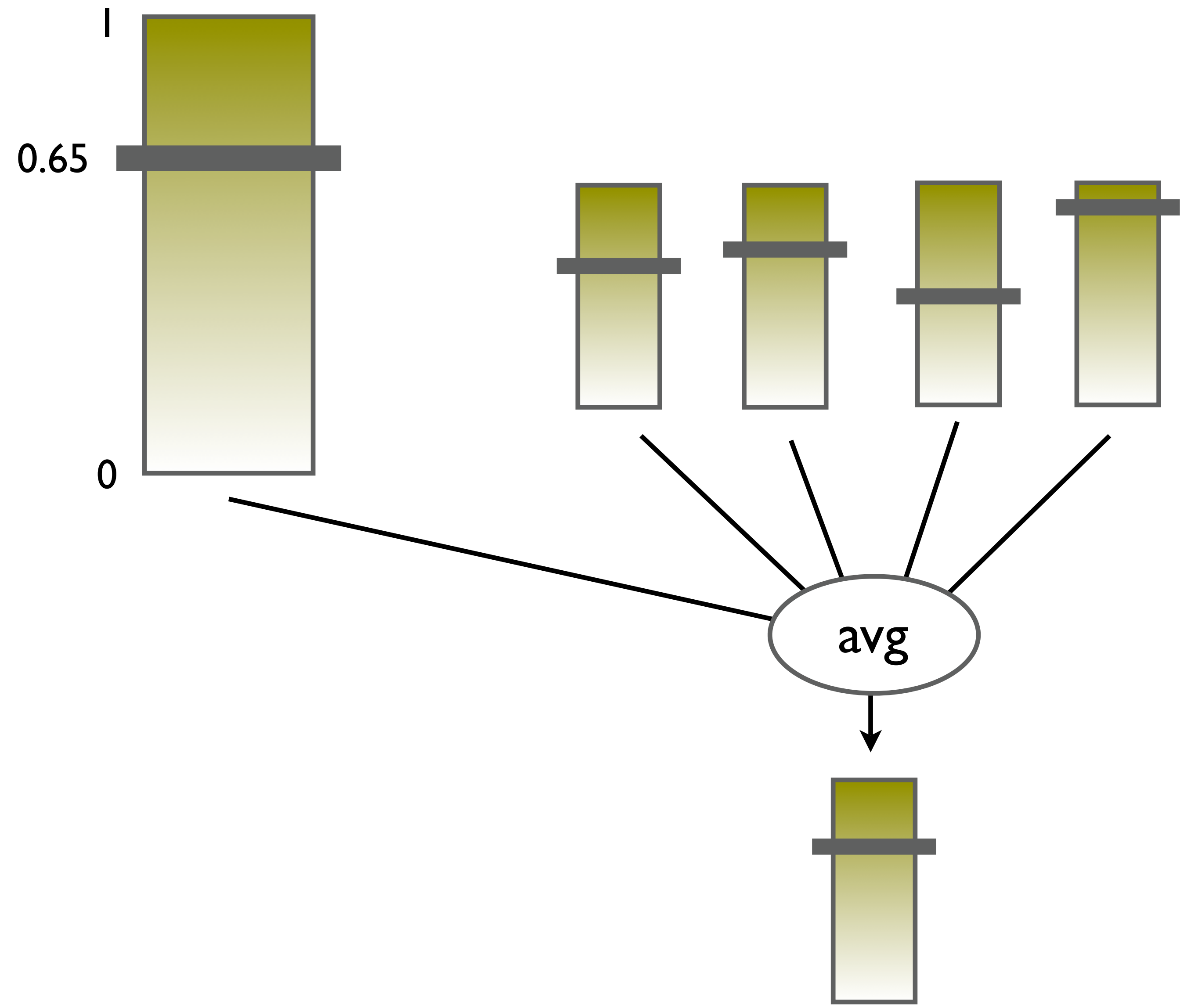
Classes of Uncertainty

- Stochastic Uncertainty
  - Due to randomness of the system
- Epistemic Uncertainty
  - Due to lack of knowledge about the randomness of the system

Epistemic Uncertainty	Purchases		Reputation [0,1]
	Satisfied	Unsatisfied	
Lower	80	20	0.8

Simple Approach

- Reputation is represented by one number
- Does not capture epistemic uncertainty
- Averaging not a principled way to combine reputation



Beta Reputation System

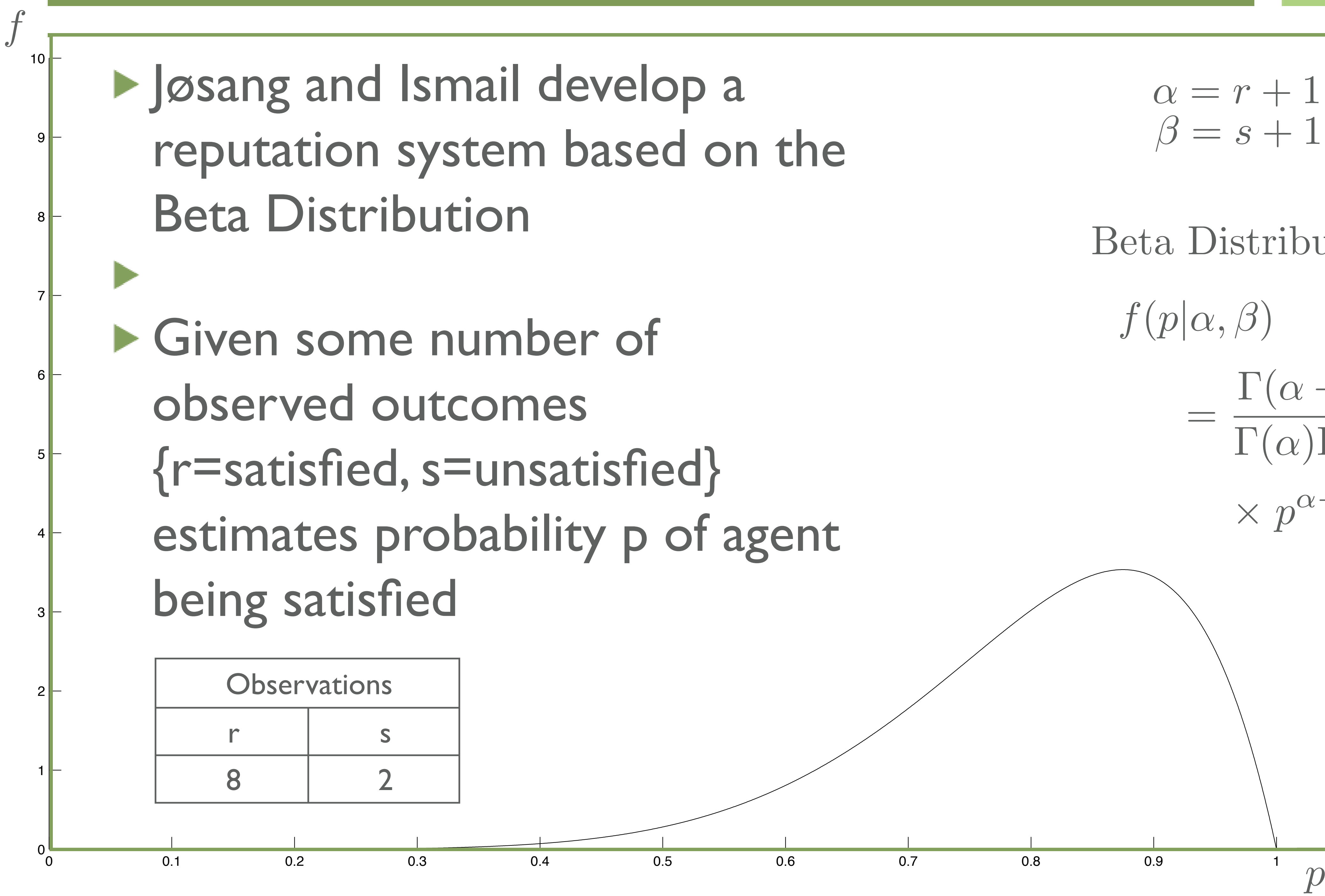
- Jøsang and Ismail develop a reputation system based on the Beta Distribution
-
- Given some number of observed outcomes {r=satisfied, s=unsatisfied} estimates probability p of agent being satisfied

Observations	
r	s
8	2

$$\alpha = r + 1$$
$$\beta = s + 1$$

Beta Distribution Function

$$f(p|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \times p^{\alpha-1} \times (1-p)^{\beta-1}$$



Beta Reputation System

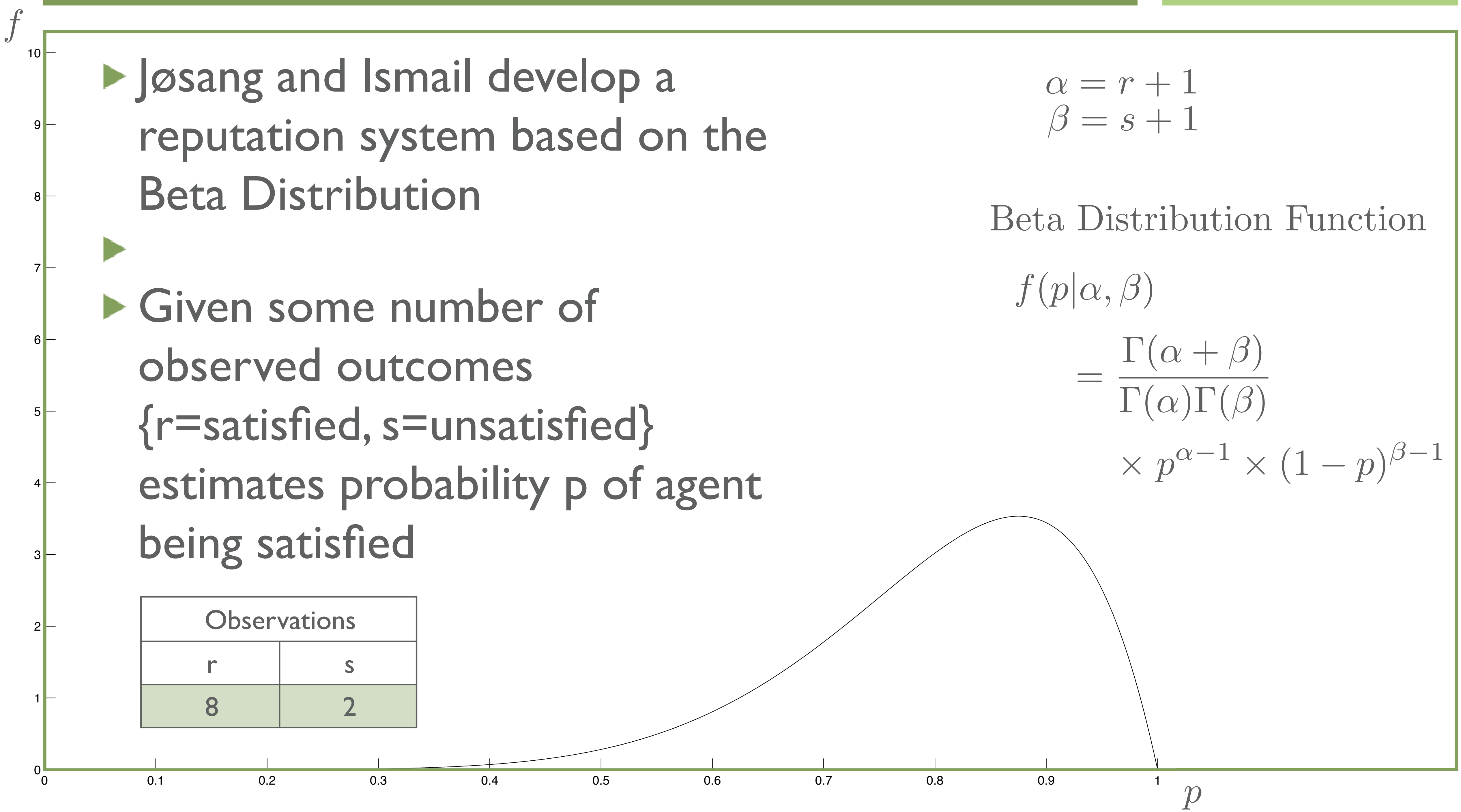
- Jøsang and Ismail develop a reputation system based on the Beta Distribution
-
- Given some number of observed outcomes {r=satisfied, s=unsatisfied} estimates probability p of agent being satisfied

$$\alpha = r + 1$$
$$\beta = s + 1$$

Beta Distribution Function

$$f(p|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \times p^{\alpha-1} \times (1-p)^{\beta-1}$$

Observations	
r	s
8	2



Beta Reputation System

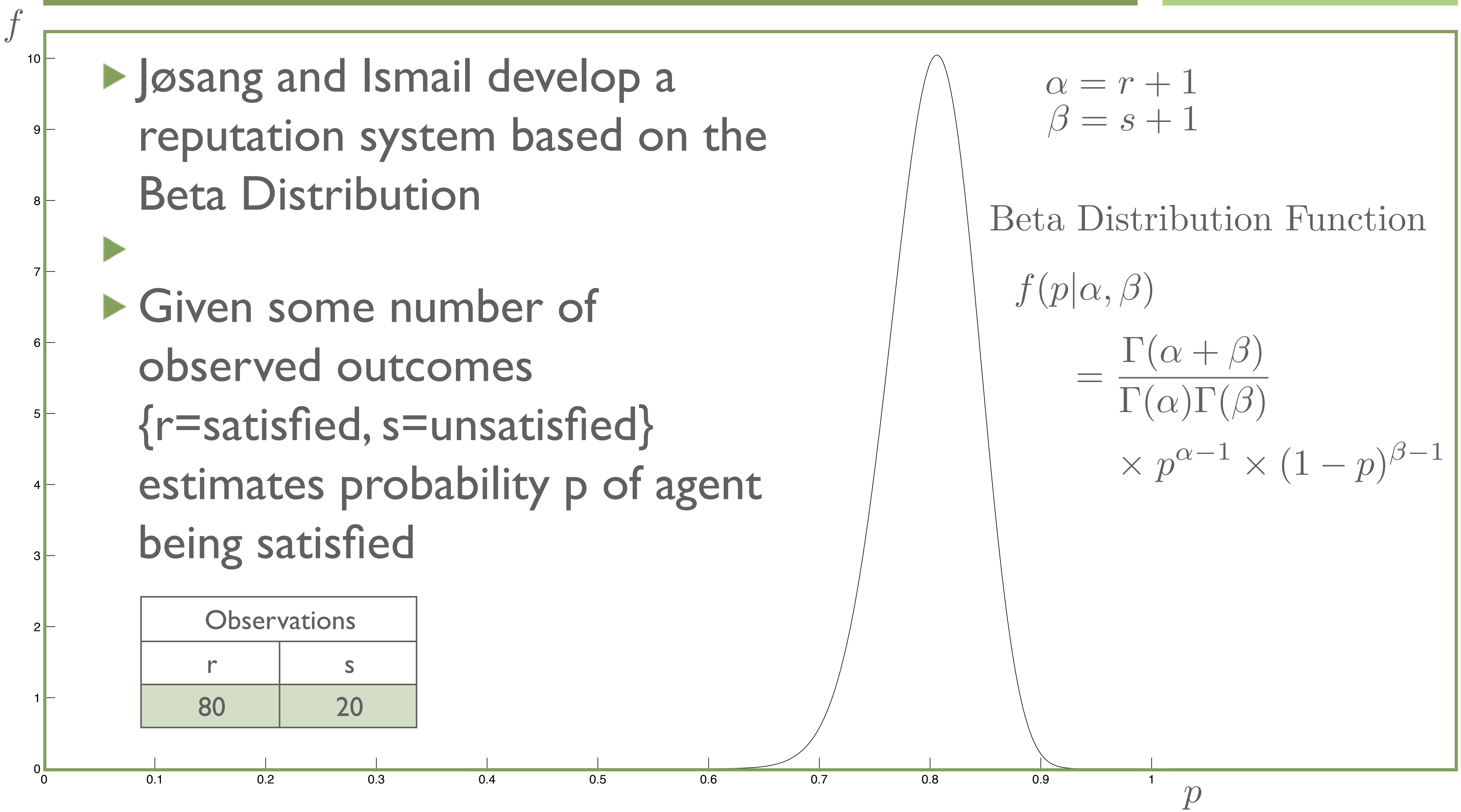
- Jøsang and Ismail develop a reputation system based on the Beta Distribution
-
- Given some number of observed outcomes {r=satisfied, s=unsatisfied} estimates probability p of agent being satisfied

Observations	
r	s
80	20

$$\alpha = r + 1$$
$$\beta = s + 1$$

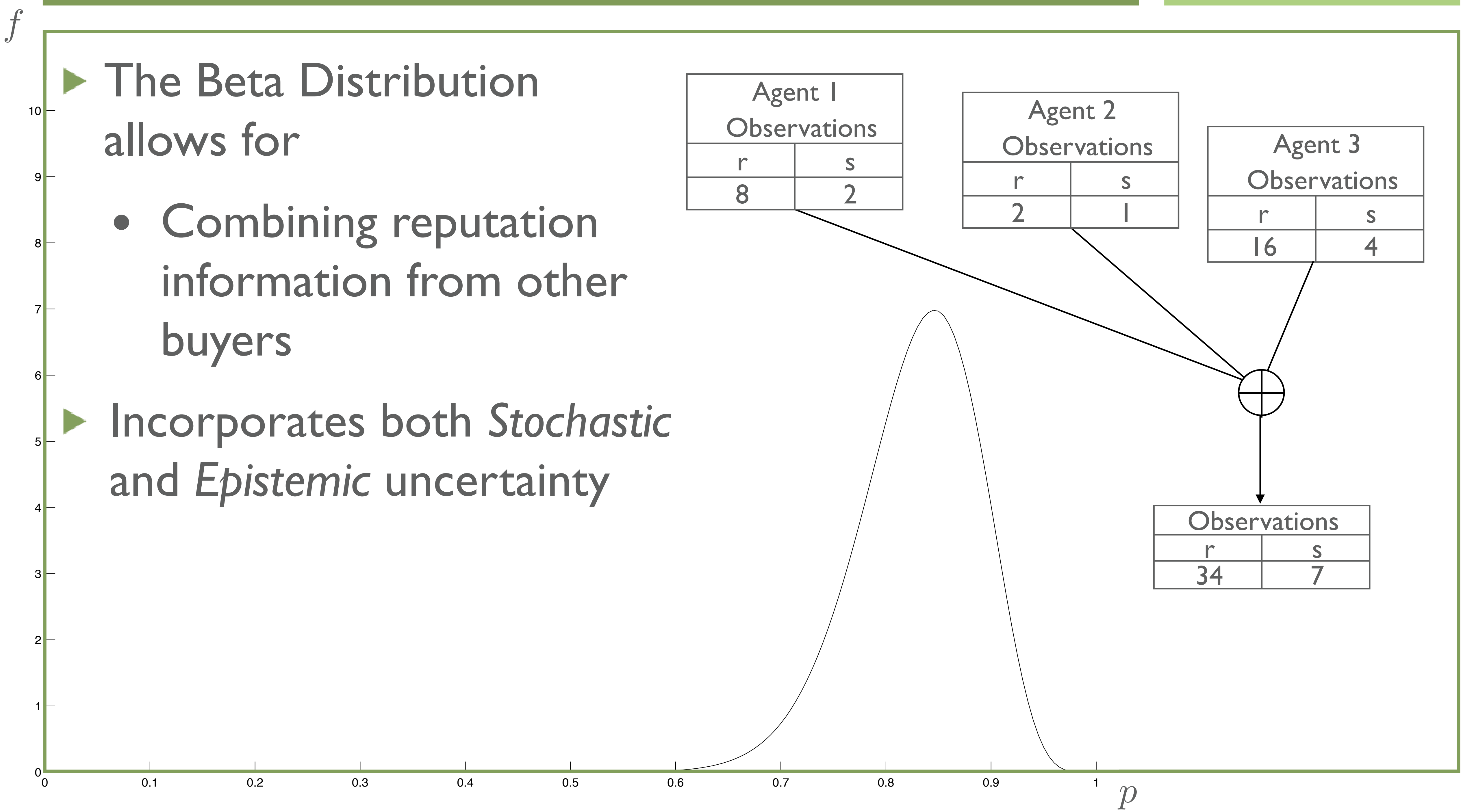
Beta Distribution Function

$$f(p|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \times p^{\alpha-1} \times (1-p)^{\beta-1}$$



Beta Reputation System

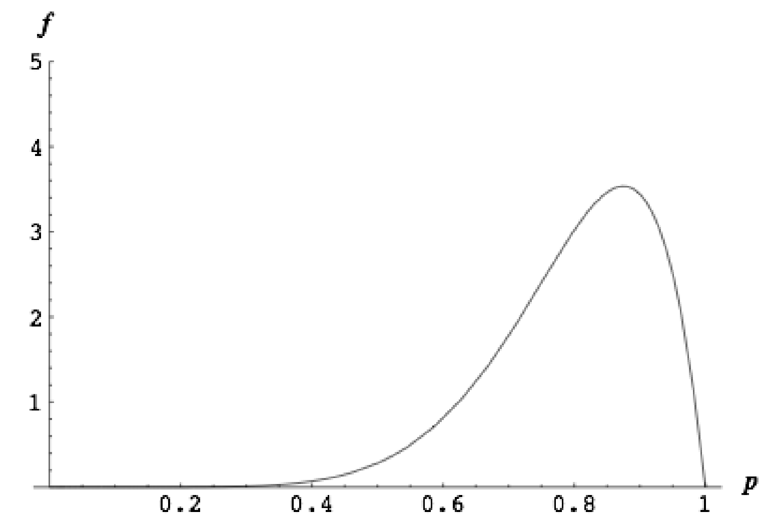
- The Beta Distribution allows for
  - Combining reputation information from other buyers
- Incorporates both *Stochastic* and *Epistemic* uncertainty



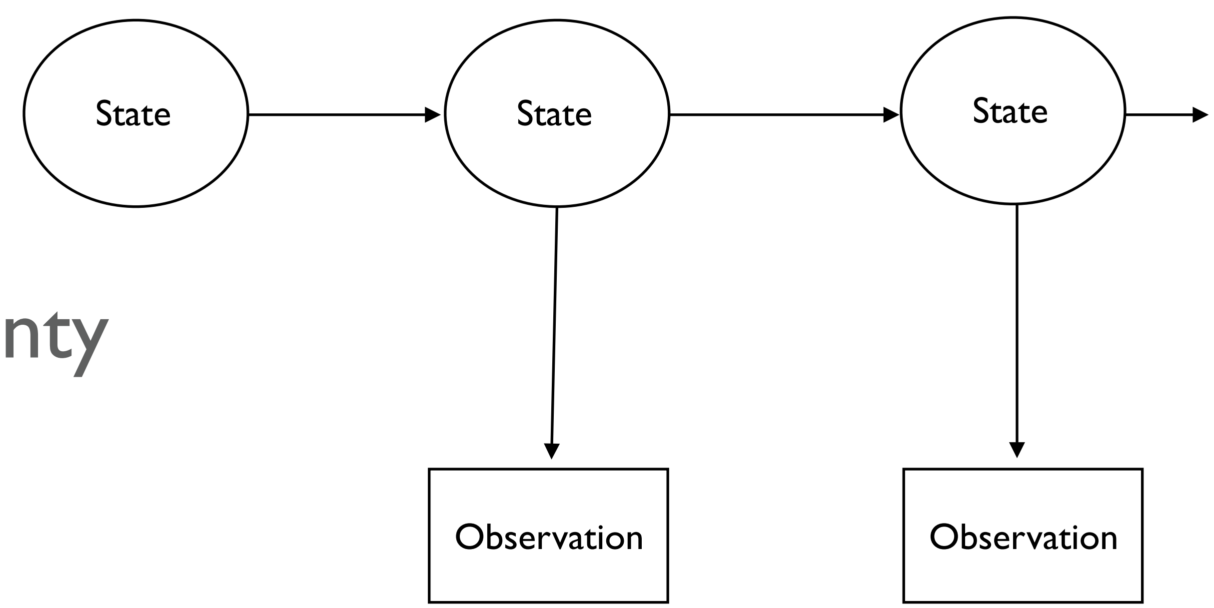
A General View

State - Represents true seller reputations

- Captures *stochastic* uncertainty
- Cannot be directly observed by buyer



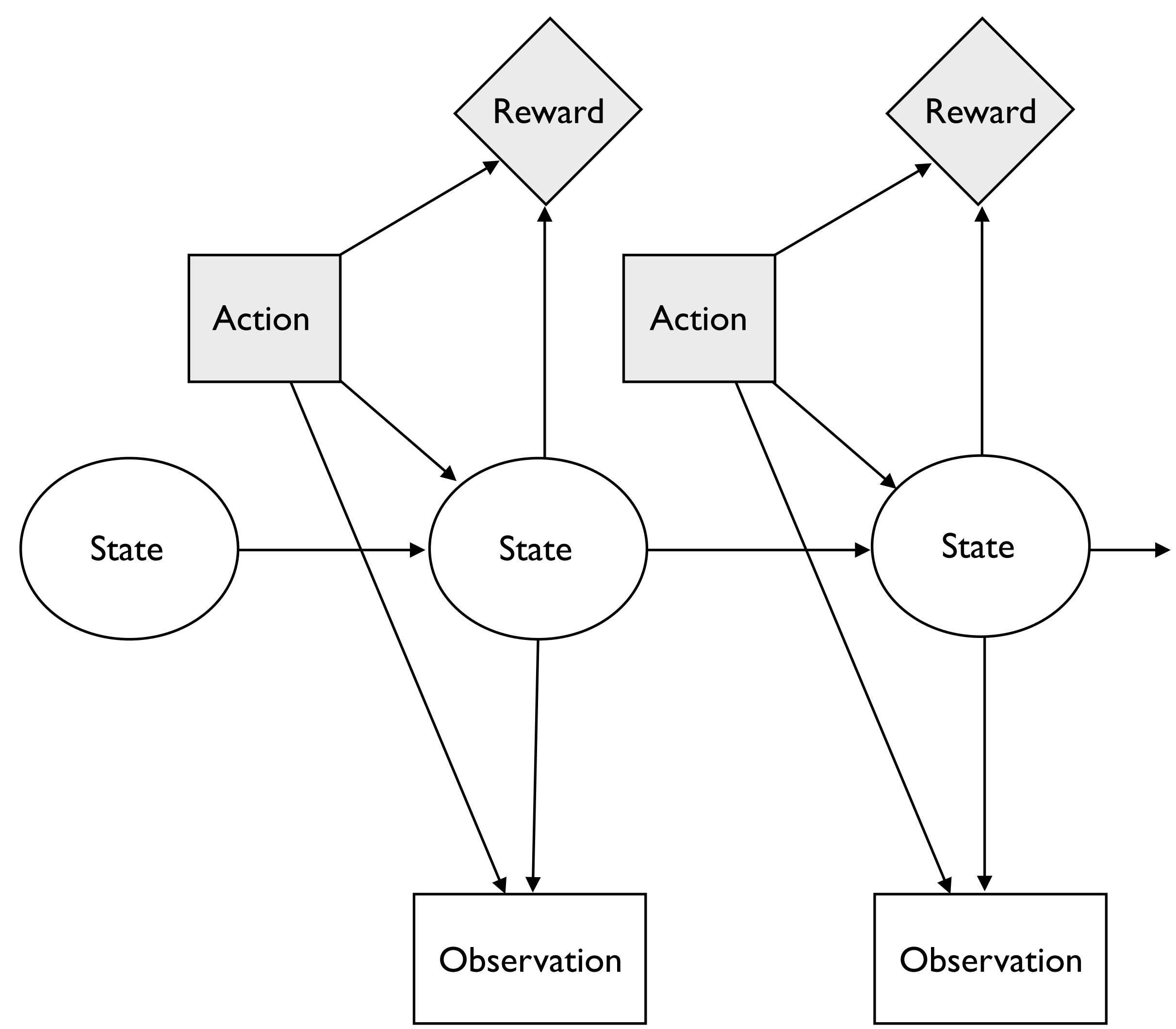
Observations - Reputation information collected from other buyers gives information about hidden state



- Decreasing *epistemic* uncertainty

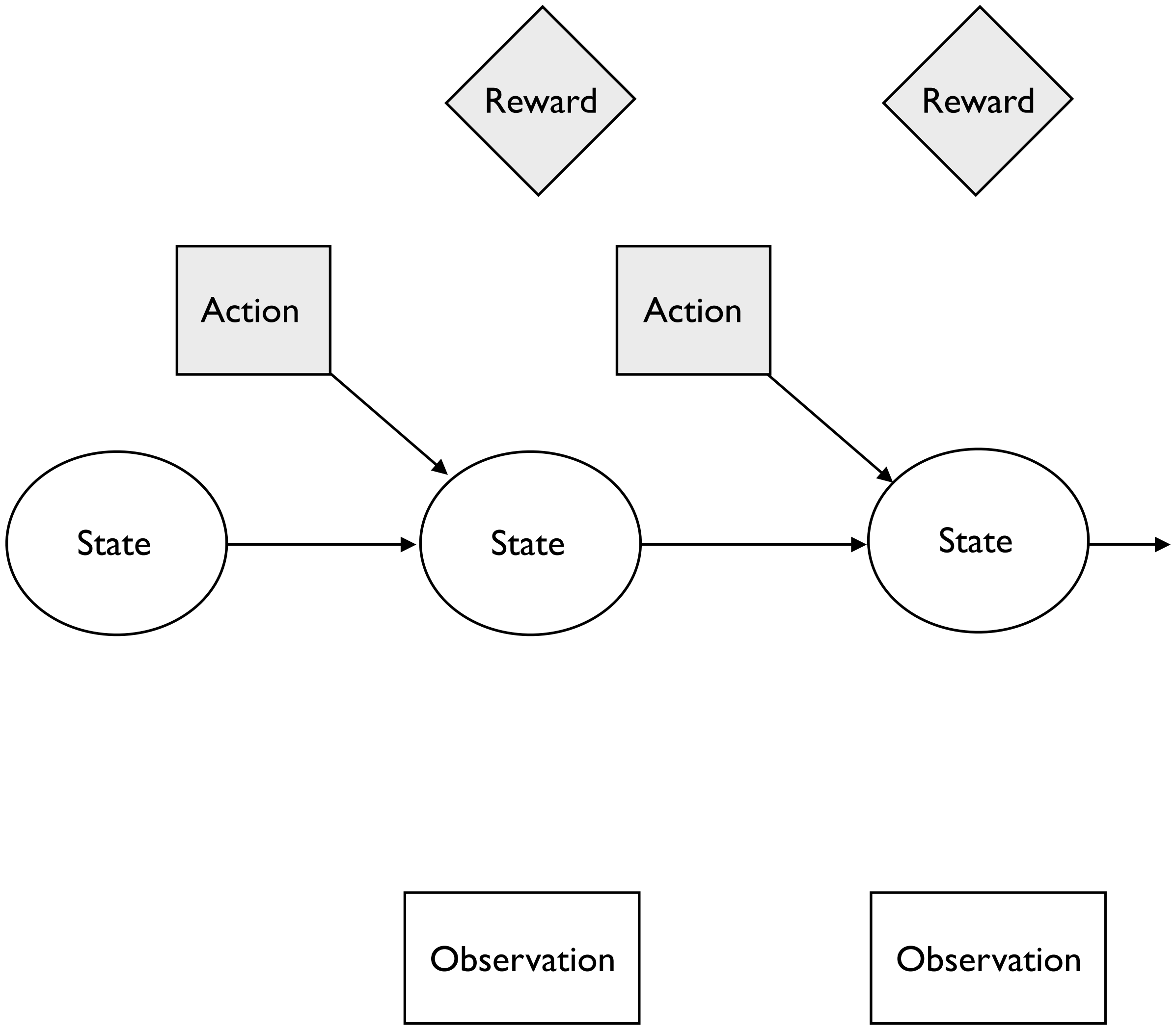
Definition of a POMDP

- POMDP defined by the tuple < S, A, R, O, T, Ω >
  - S - State
  - A - Action
  - R - Reward
  - O - Observation
  - T - Transition function
  - Ω - Observation function



Definition of a POMDP

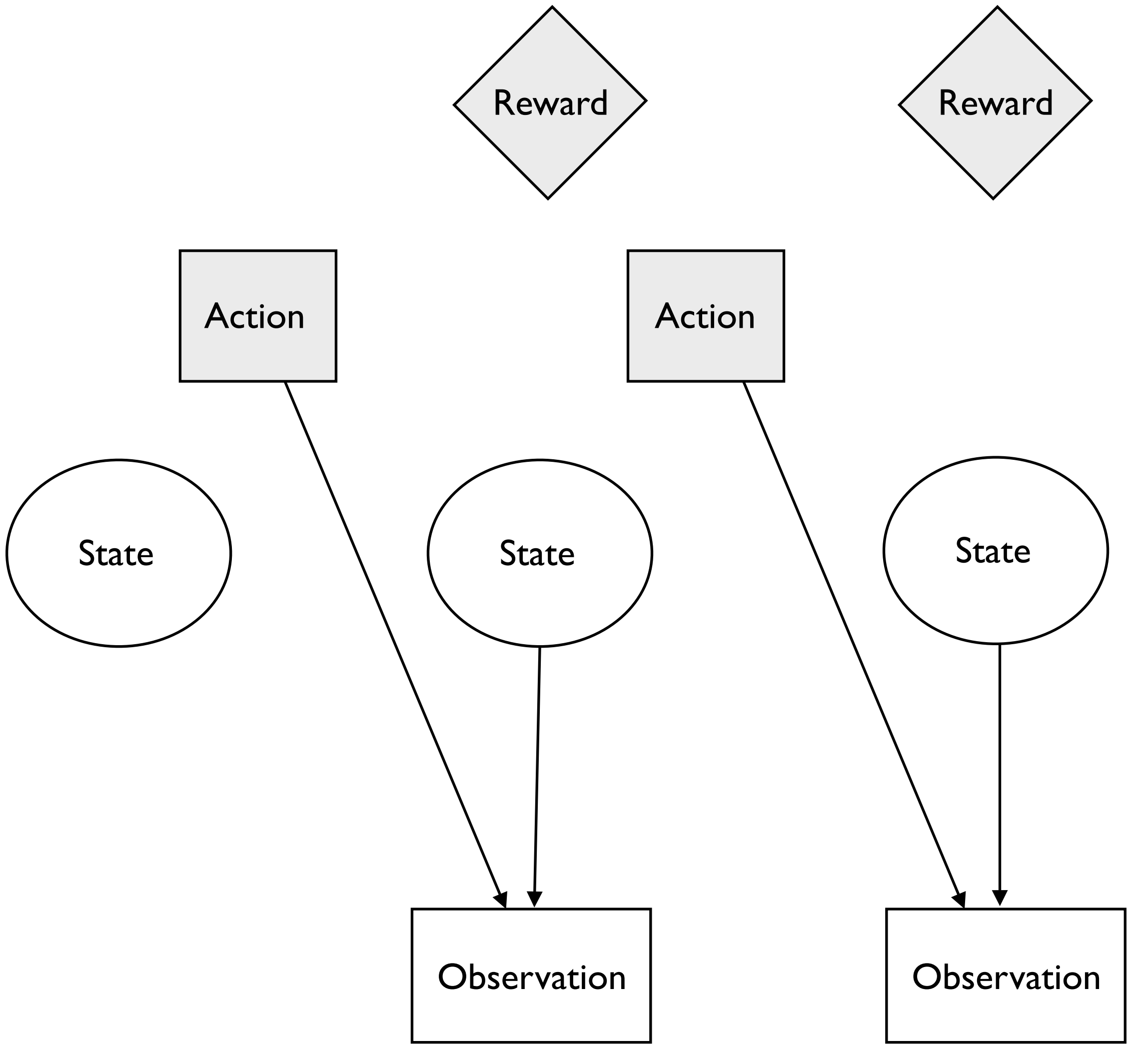
- POMDP defined by the tuple < S, A, R, O, T, Ω >
  - S - State
  - A - Action
  - R - Reward
  - O - Observation
  - T - Transition function
  - $T(s,a,s') = [0,1]$
  - Ω - Observation function



Definition of a POMDP

- POMDP defined by the tuple $< S, A, R, O, T, \Omega >$
  - S - State
  - A - Action
  - R - Reward
  - O - Observation
  - T - Transition function
  - Ω - Observation function

    $\Omega(s,a,o) = [0,1]$

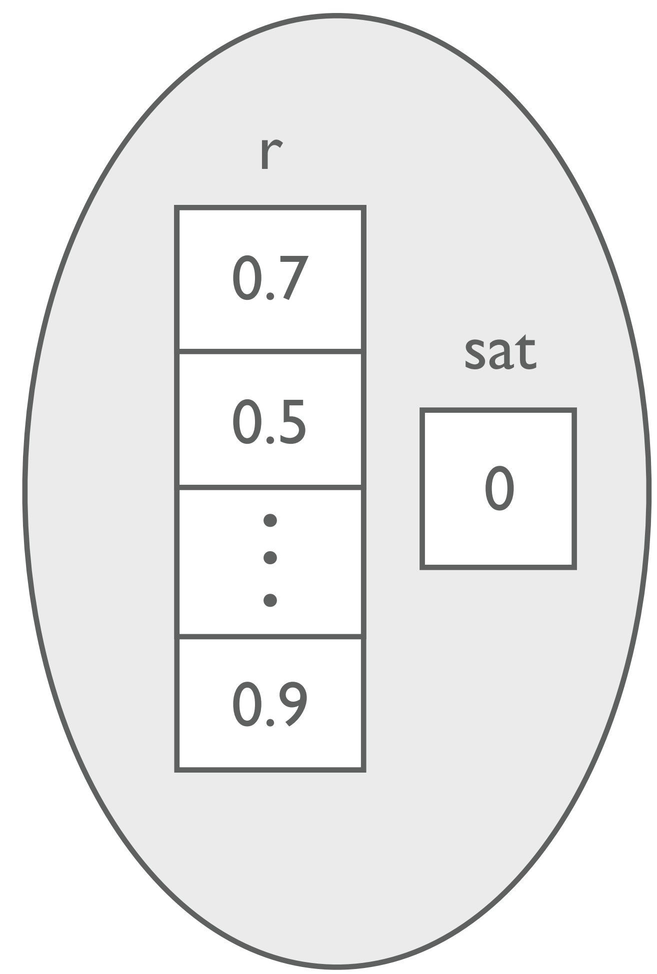


Advisor POMDP

State

- A state is the tuple < r, sat > where
  - r is a vector of real values [0,1] representing the reputation of each seller
  - sat is a scalar value of either -1, 0 or 1 representing the satisfaction resulting from a purchase

Before Purchase (advice state)		sat = 0
After Purchase	Satisfied	sat = 1
	Unsatisfied	sat = -1



Advisor POMDP

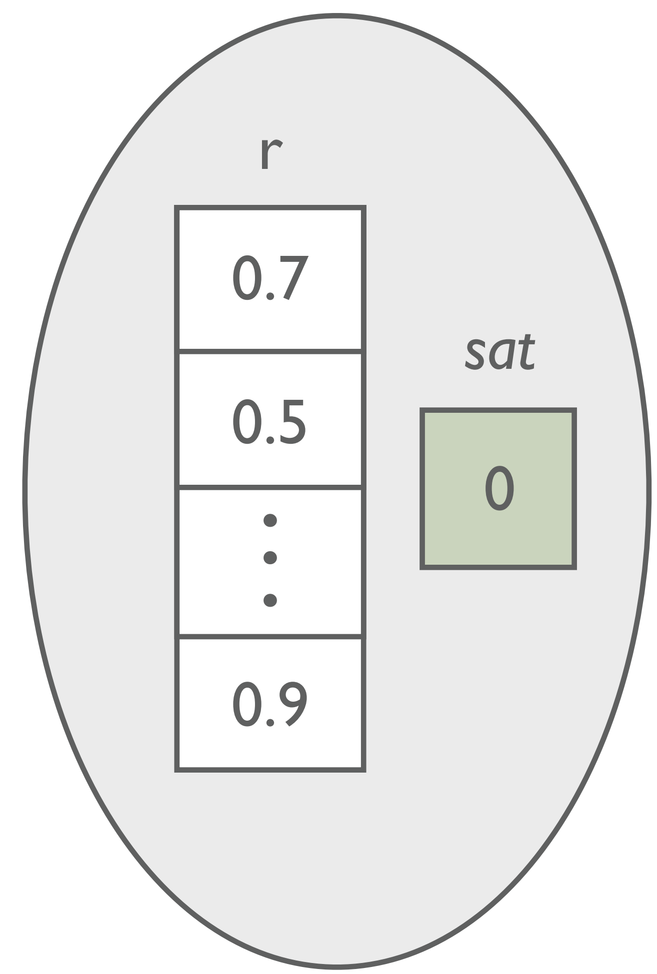
Actions

- A buying agent can choose from two sets of possible actions. It can either choose to:
  - ***Ask*** an another buyer for information about selling agent
  - ***Buy*** from a selling agent

Advisor POMDP

Reward

sat value	signifies	Reward
1	Satisfied by purchase	large & positive
-1	Unsatisfied by purchase	large & negative
0	Gathering information	small cost



Advisor POMDP

Observations

- An observation is the tuple $< rep_i, cf_i >$ where for each seller i:
  - rep_i is the seller reputation
  - cf_i is the confidence factor
    - For simplicity we use the number of transactions with seller i

Seller	rep	cf
s1	0.5	4
⋮	⋮	⋮
sn	0.9	20

Advisor POMDP

State-Transition Function

- In an advice state
  - After an *ask* action we will transition back to the same state (as the true reputation of sellers does not change) and the *belief* about this state will be updated
  - After a buy action we will transition to purchase state where *sat* value represents outcome of purchase

Advisor POMDP

Observation Function

- The observation function expresses the likelihood of receiving an observation given the current state and the action that led to this state
- Used to update our *belief* over possible states

Advisor POMDP

Policy

- Given our definition of a POMDP we can calculate a policy π which maps each belief to the action that will maximize the expected reward
- This policy will make the best tradeoff between exploring the market by asking other buyers and exploiting the information it has by making a purchase

Calculating Policies

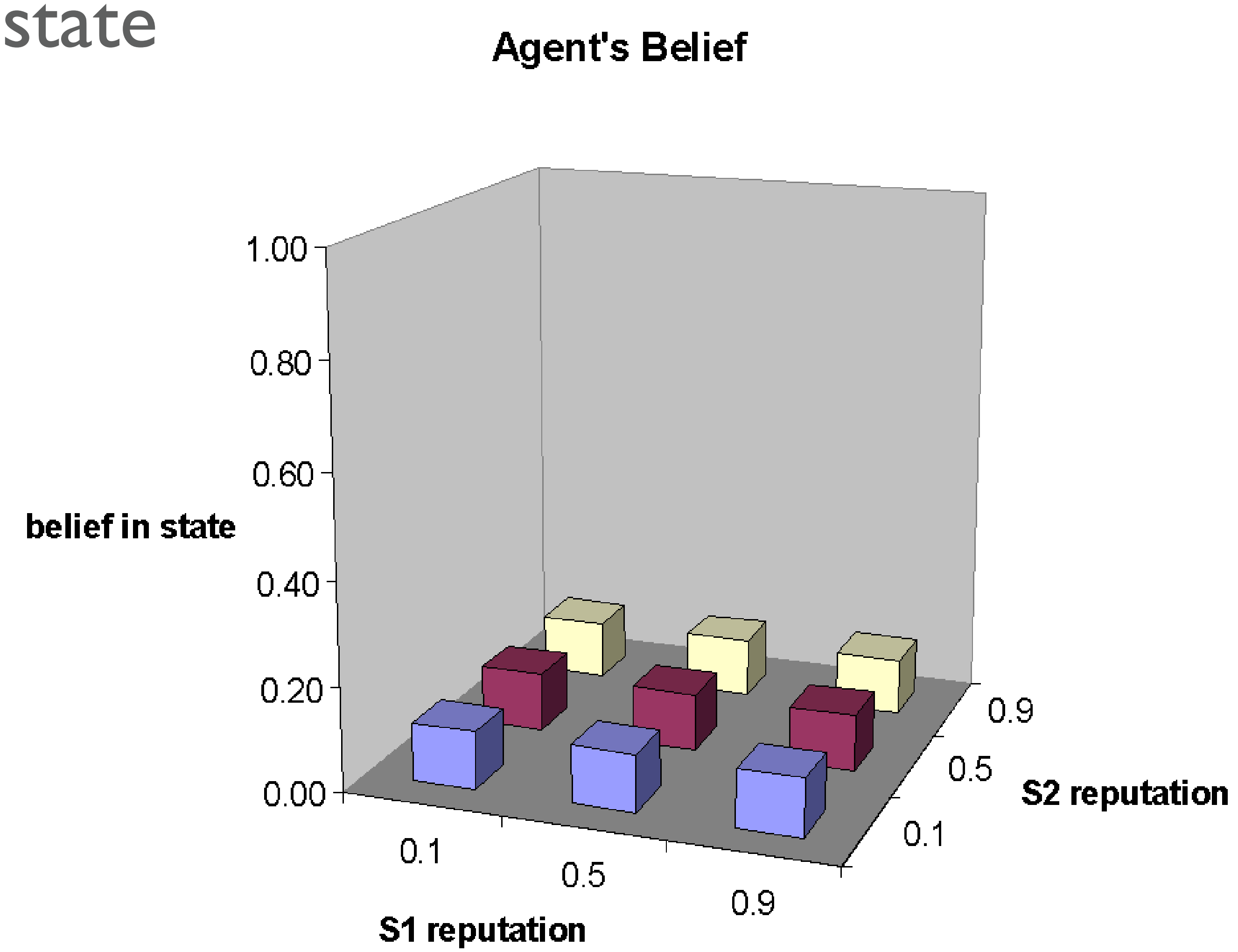
- Value Iteration
  - Uses dynamic programming to indirectly compute an optimal policy by computing an optimal value function
  - An example of this approach is point based value iteration
- Policy Search
  - Incrementally improve a policy by searching through modifications to the policy
  - An example of this approach is gradient descent

Example

- We have a buyer choosing among
  - A set of sellers s1, s2
  - Using a set of buying agents a1, a2, a3, a4 who provide seller's reputation
- Given a policy generated for the POMDP we will step through the actions taken based on the current belief state, noting the observation generated and how it influences the next belief state

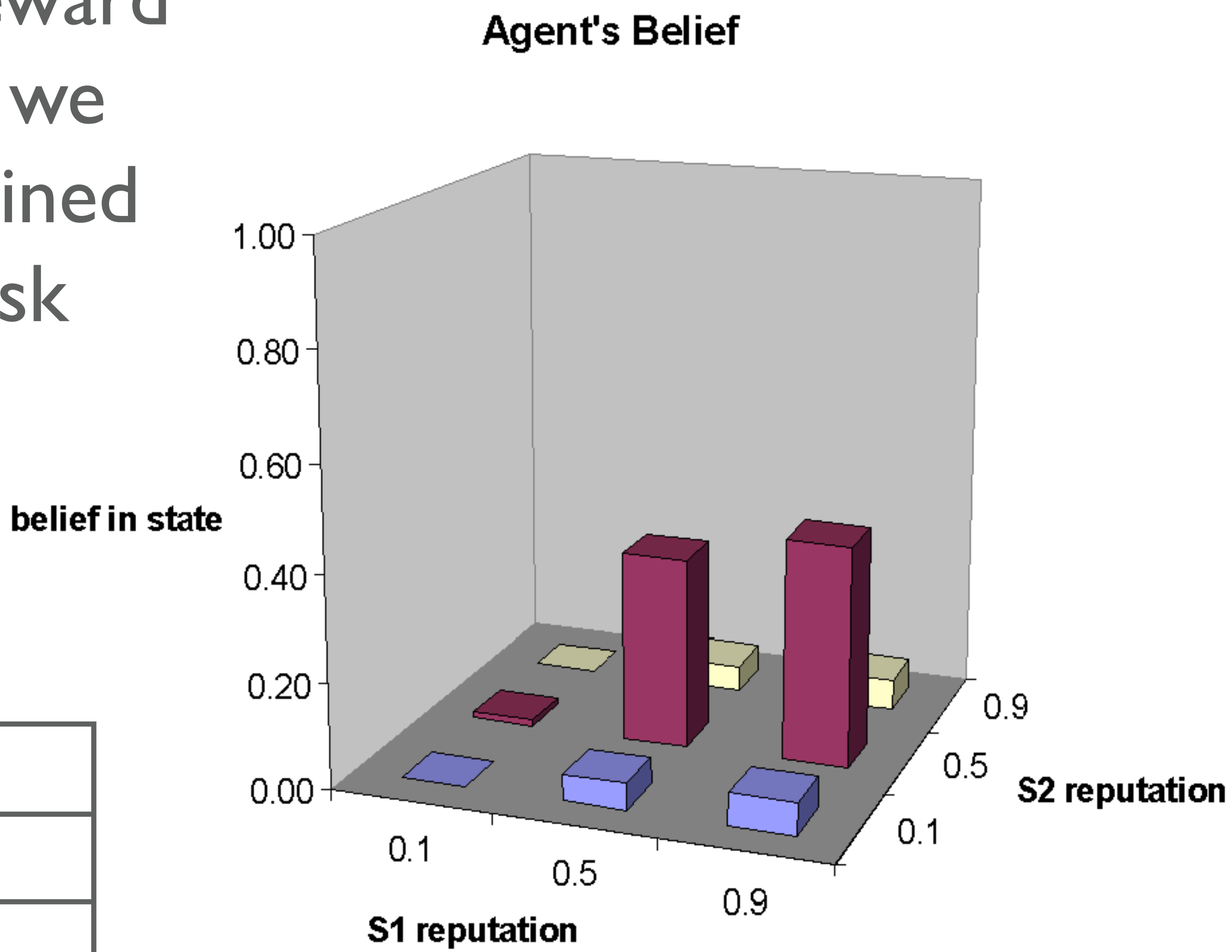
Example - Initial Belief

- Initially the agent's belief is flat giving equal weight to each possible state



Example - Action 1

- Given the flat belief state the agent will have a higher expected reward after asking an another buyer, we assume the policy has determined a1 to be the best advisor to ask
- Action - Ask a1
- Observation

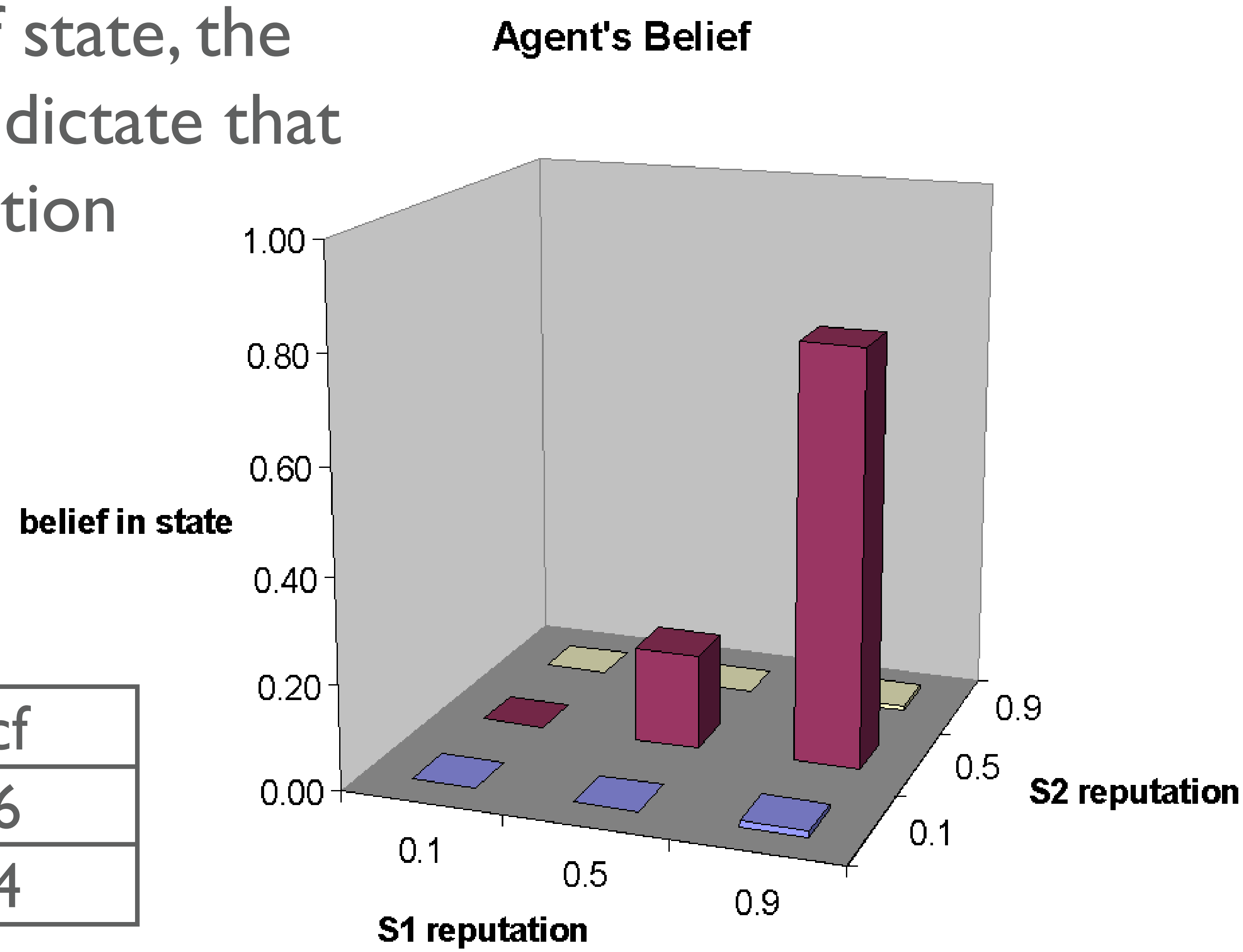


	reputation	cf
s1	0.9	20
s2	0.5	12

Example - Action II

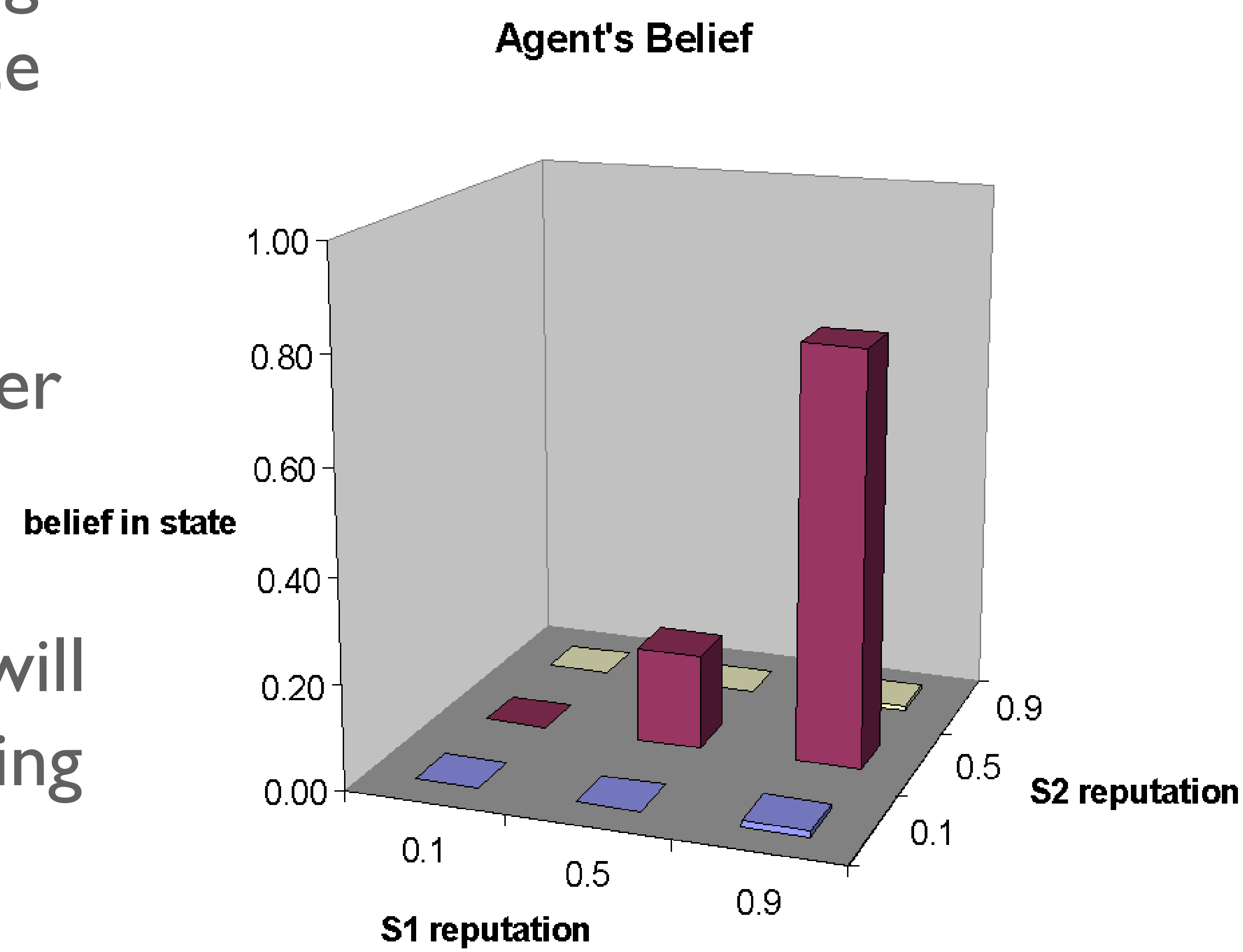
- Given the updated belief state, the policy would once again dictate that our buyer take an *ask* action
- Action - Ask a3
-
- Observation

	reputation	cf
s1	0.8	6
s2	0.5	4



Example - Action III

- At this point there is enough of a peak in the belief space that the best action is to select a seller
- Given the agent's belief over seller reputations the expected reward for purchasing from seller s1 will be far higher than purchasing from s2
- Action - buy from s1



Conclusions

- Reputation systems need to capture both stochastic and epistemic uncertainty
- The Advisor-POMDP decision theoretic framework
  - captures both kinds of uncertainty
  - making optimal trade-offs between *exploring* to gather reputation information and *exploiting* this information by making a purchase

Future Work

- Refine current model
  - Methods for extracting usable policies
  - Empirical analysis with comparison to other social reputation systems
- Extend model
  - Beyond satisfied & unsatisfied
  - Address subjectivity & deception

Questions?

Motivation

Reputation systems are the worst way of building trust on the Internet, except for all those other ways that have been tried from time-to-time

-Paul Resnick *by way of Winston Churchill*

Observation Function

$$P(o|s)$$

$$= P(o =<< rep_{s1}, cf_{s1} >, < rep_{s2}, cf_{s2} >> \mid s =< r_{s1}, r_{s2} >)$$

$$= (r_{s1})^{rep_{s1} cf_{s1}} \cdot (1 - r_{s1})^{(1-rep_{s1}) cf_{s1}}$$

$$\times (r_{s2})^{rep_{s2} cf_{s2}} \cdot (1 - r_{s2})^{(1-rep_{s2}) cf_{s2}}$$

Belief Update

$$
\begin{aligned}
b'(s) &= P(s|o) \\
&= \frac{P(s)P(o|s)}{P(o)} \\
&= k \cdot b(s)O(s,o)
\end{aligned}
$$

Calculating Policies

$$V^n(b) = \max_a R^a(b) + \sum_o P(o|b,a)V^{n-1}(b')$$

where

$$P(o|b,a) = \sum_s b(s)P(s'|s,a)P(o|s')$$

$$R^a(b) = \sum b(s)R^a(s)$$