Recipe tuning by Reinforcement Learning in the SandS ecosystem

B. Fernandez-Gauna, M. Graña*

*Computational Intelligence Group, UPV/EHU[1]

July 30, 2014

Summary

- Social and Smart (SandS) project ecosystem: household appliance users, recipes, and an intelligent social layer.
  - innovation producing new recipes for unkown user tasks,
  - and the adaptation to personalize the recipe.
- Reinforcement Learning: user feedback == system reward.
- actor-critic approach,
- providing some experimental results on synthetic datasets

Contents

Contents

Introduction

Fact

Social networks can be seen as a repository of information and knowledge that can be queried when needed to solve problems or to learn procedures.

.

Fact

In the social sciences, social networks have been useful to spread educational innovations

- in health care training
- management of product development programs,
- engagement in agricultural innovations by farmers.

Crowdsourcing

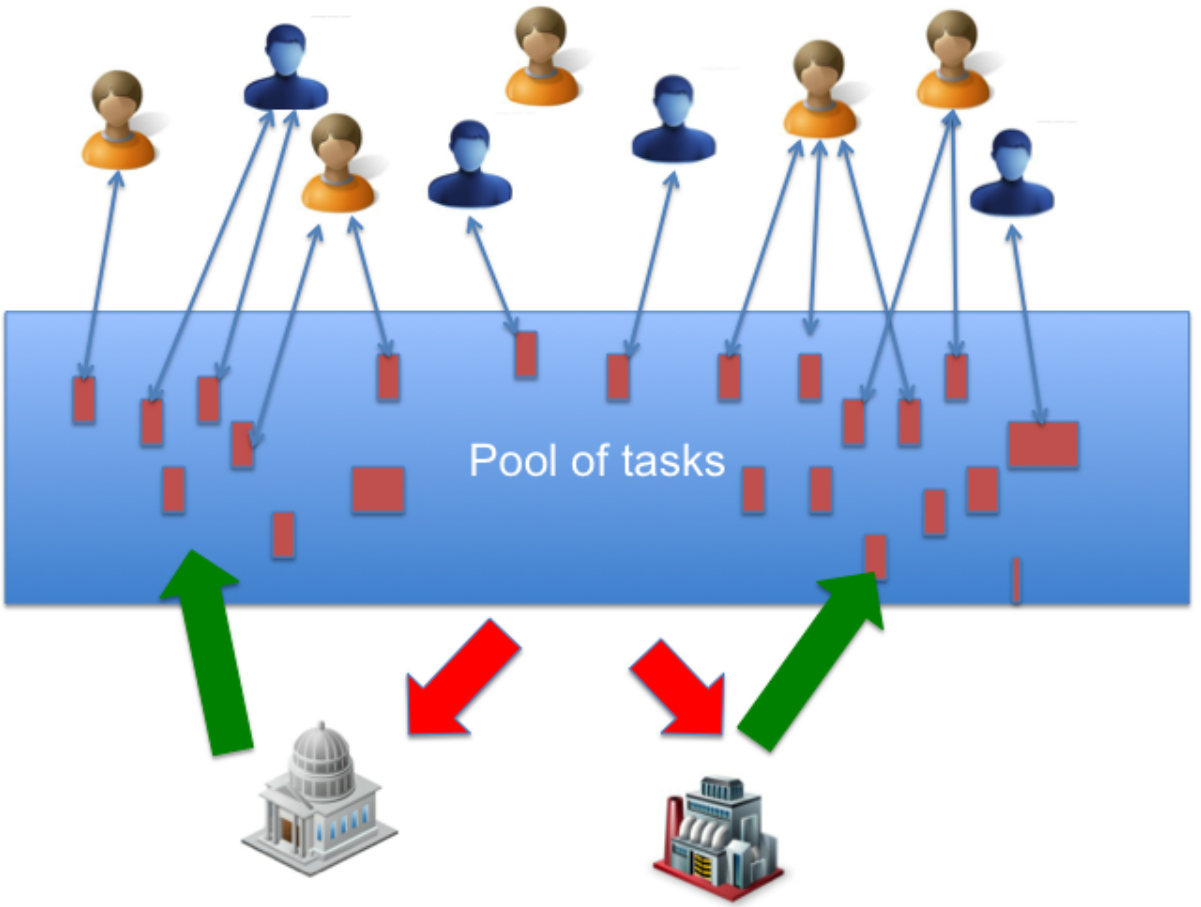


Figure: Crowdsourcing paradigm

Crowdsourcing

Crowdsourcing "enlists a crowd of users to explicitly collaborate to build a long-lasting artifact that is beneficial to the whole community"[2]

- how to recruit and retain users;
- what can users do;
- how to combine their inputs; and
- how to evaluate them

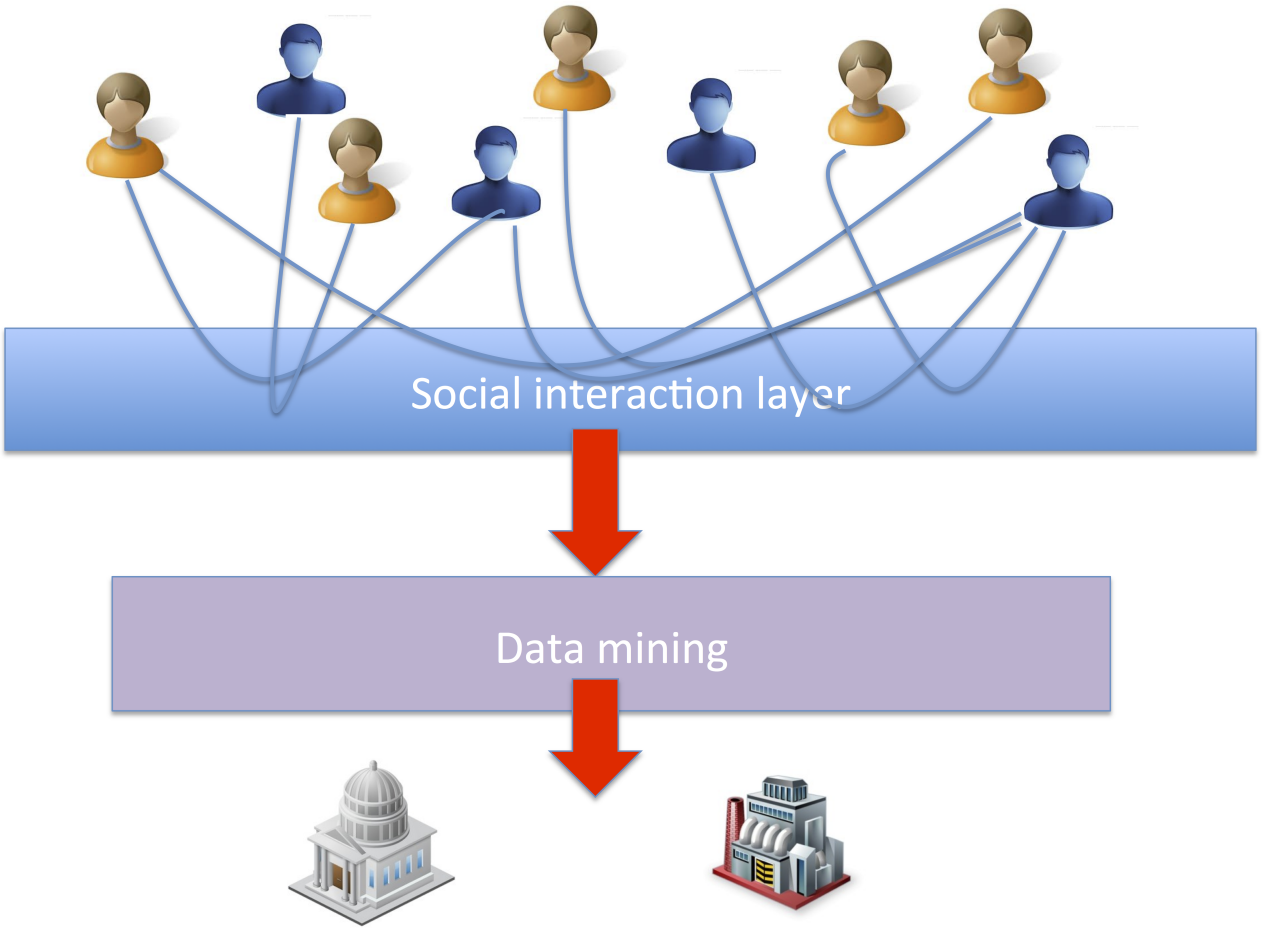
[2] Anhai Doan, Raghu Ramakrishnan, and Alon Y. Halevy, Crowdsourcing systems in the World-Wide Web, CACM, (2011) 54:86-96

Crowdsourcing efforts

- Galaxy Zoo [3]: classifying galaxy images
- FolfIt [4]: solving protein folding puzzles
- Image labeling [5]
- reCAPTCHA [6] for crowsourced OCR
- Wikipedia, sourceforge...
- Amazon Mechanical Turk

[3] http://www.galaxyzoo.org
[4] http://fold.it/portal/
[5] http://www.artigo.org/about.html
[6] http://www.google.com/recaptcha/learnmore

Computational Social Sciences

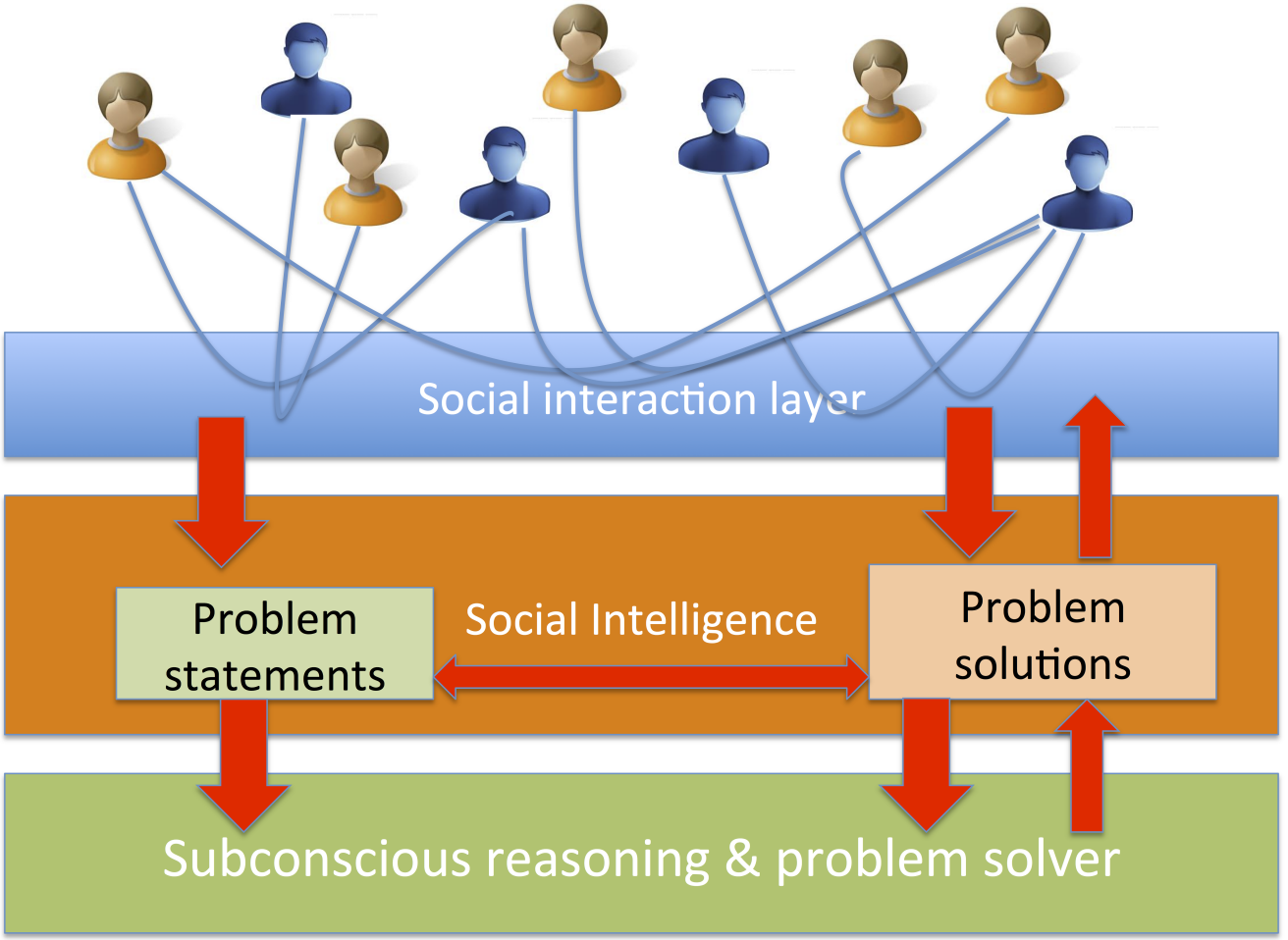


Social Computing and Computational Social Science paradigm

Computational social sciences

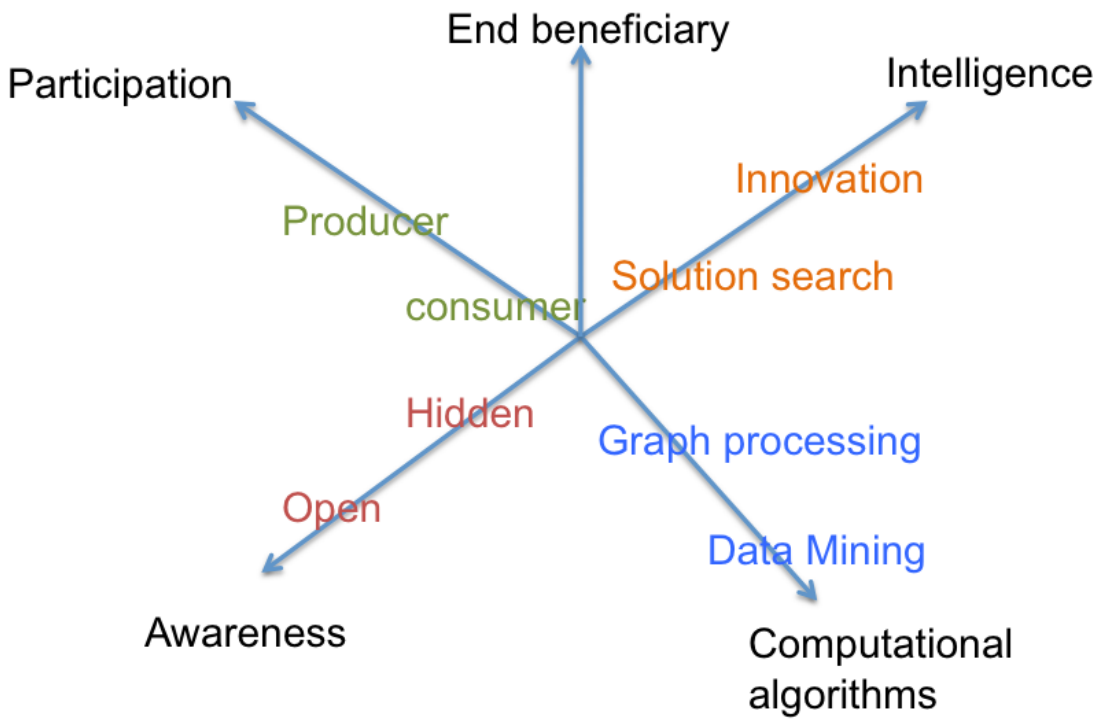
- User profiling
  - Targeted marketing
- Community discovery
  - New product development
- Security
- Sentiment Analysis
- Process mining

Subconscious social intelligence



Subconscious Social Intelligence paradigm

Axes of a Taxonomy

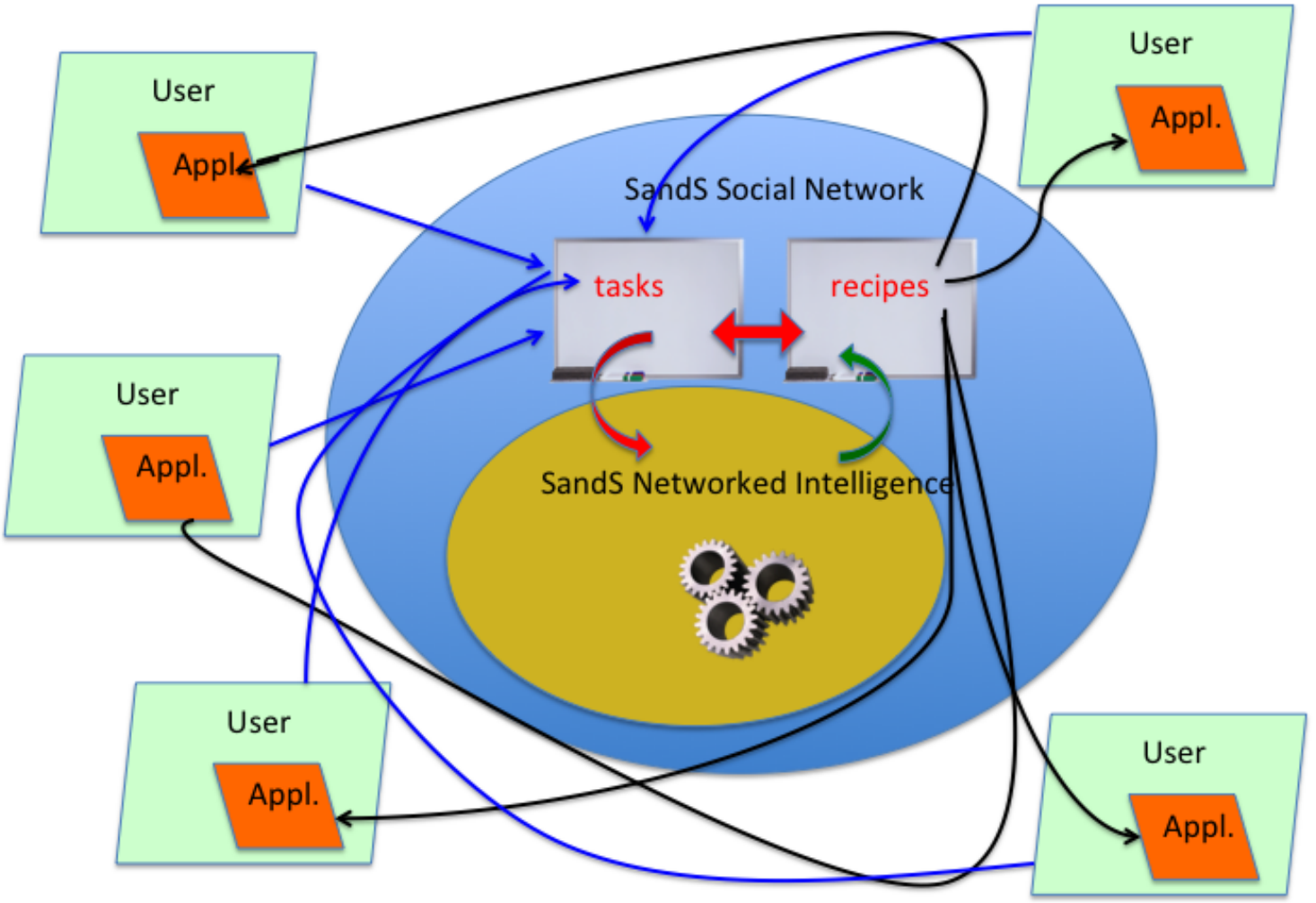


Axes of social computing taxonomy

SandS project

- The Social and Smart (SandS) project aims
  - to lay the foundations for a social network of home applicance users
  - endowed with a layer of intelligent systems
    - to produce new solutions to new problems
    - from knowledge accumulated by the social players.
- The system is not a symple recolection of tested appliance use recipes,
  - generate new recipes trying to satisfy user demands,
  - fine tuning of recipes on the basis of user satisfaction
    - by a hidden reinforcement learning process.

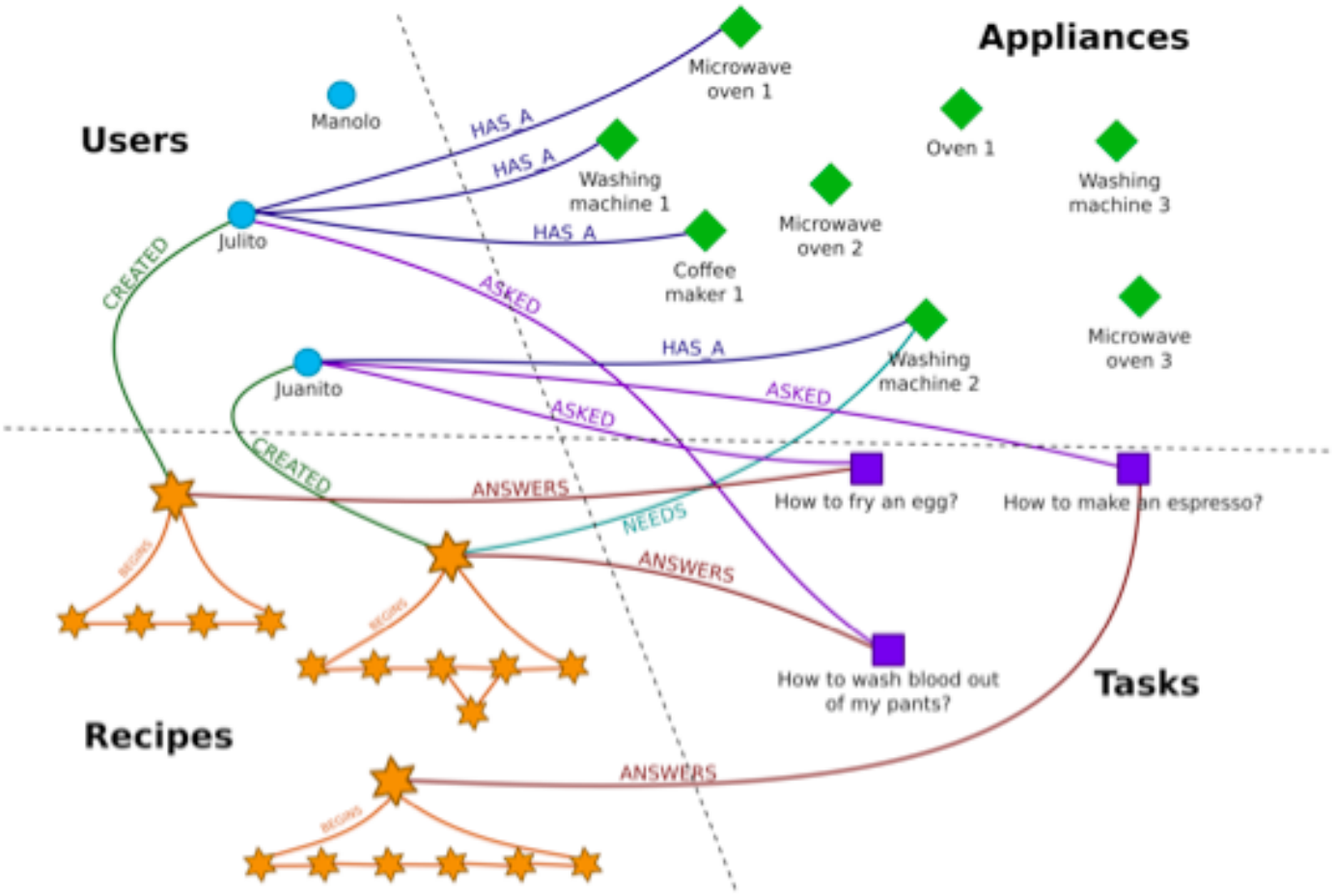
The SandS architecture



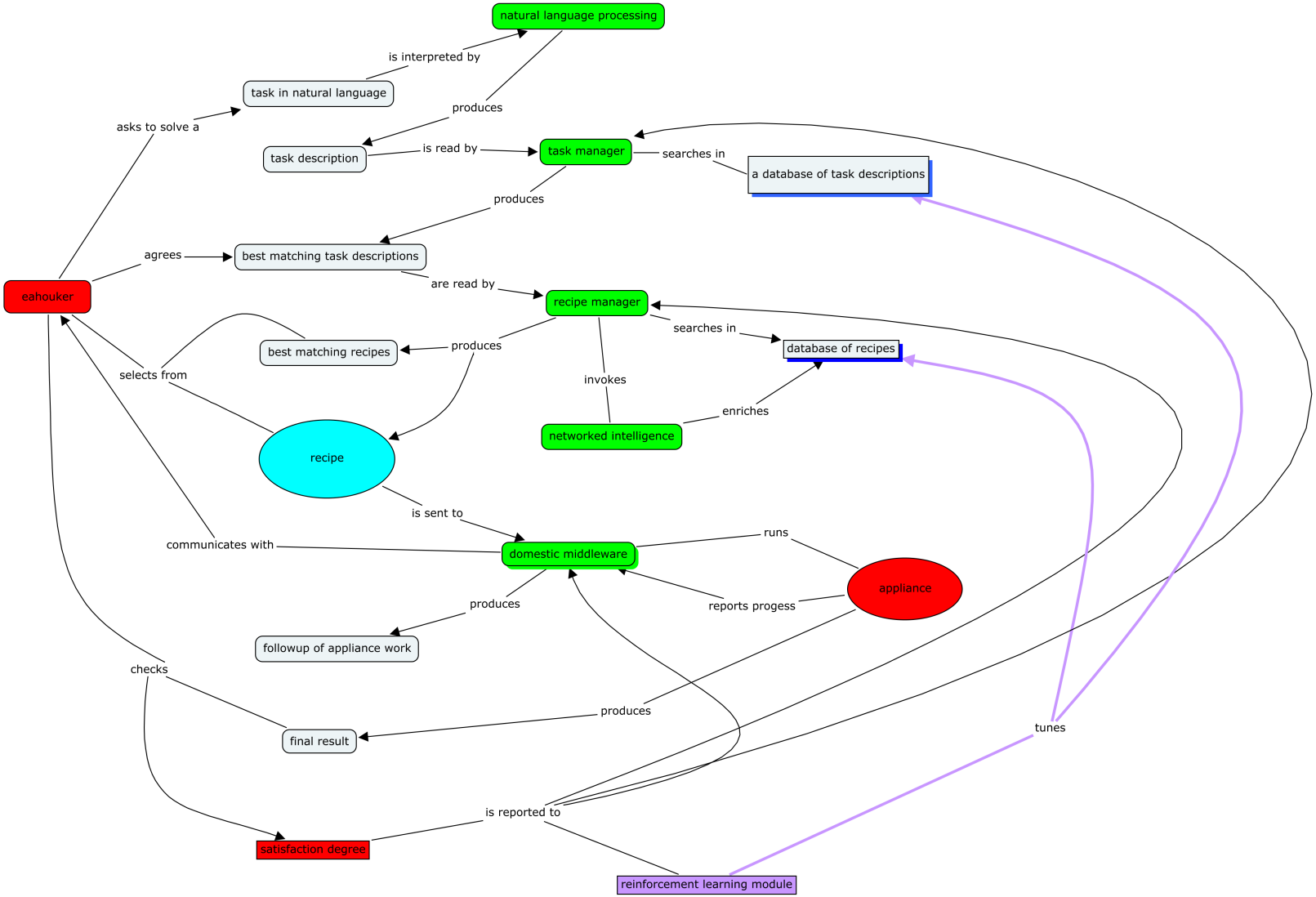
The SandS architecture

- Tasks
  - Specified by the user
- Recipes provided by
  - Appliance Manufacturer
  - User: conscious innovation
  - Networked intelligence: subconscious innovation,
- reinforcement learning for personalization

SandS knowledge representation



SandS interaction



Recipe (washing) as a process

Contents

Interaction Actor-Critic

The learning scheme adapted to the SandS project is:

1. The eahouker sets the parameters of the task he/she wants to accomplish ($t_i \in T$).
2. The actor reacts outputting the recommended recipe $r_i \in R$ according its actual policy.
3. Upon completion of the task, the user gives his/her satisfaction $s_i \in S$ and the critic updates the value δ_i of the actor's policy for task t_i accordingly
4. The value update δ_i is passed then to the actor for policy updating.

Interaction Actor-Critic

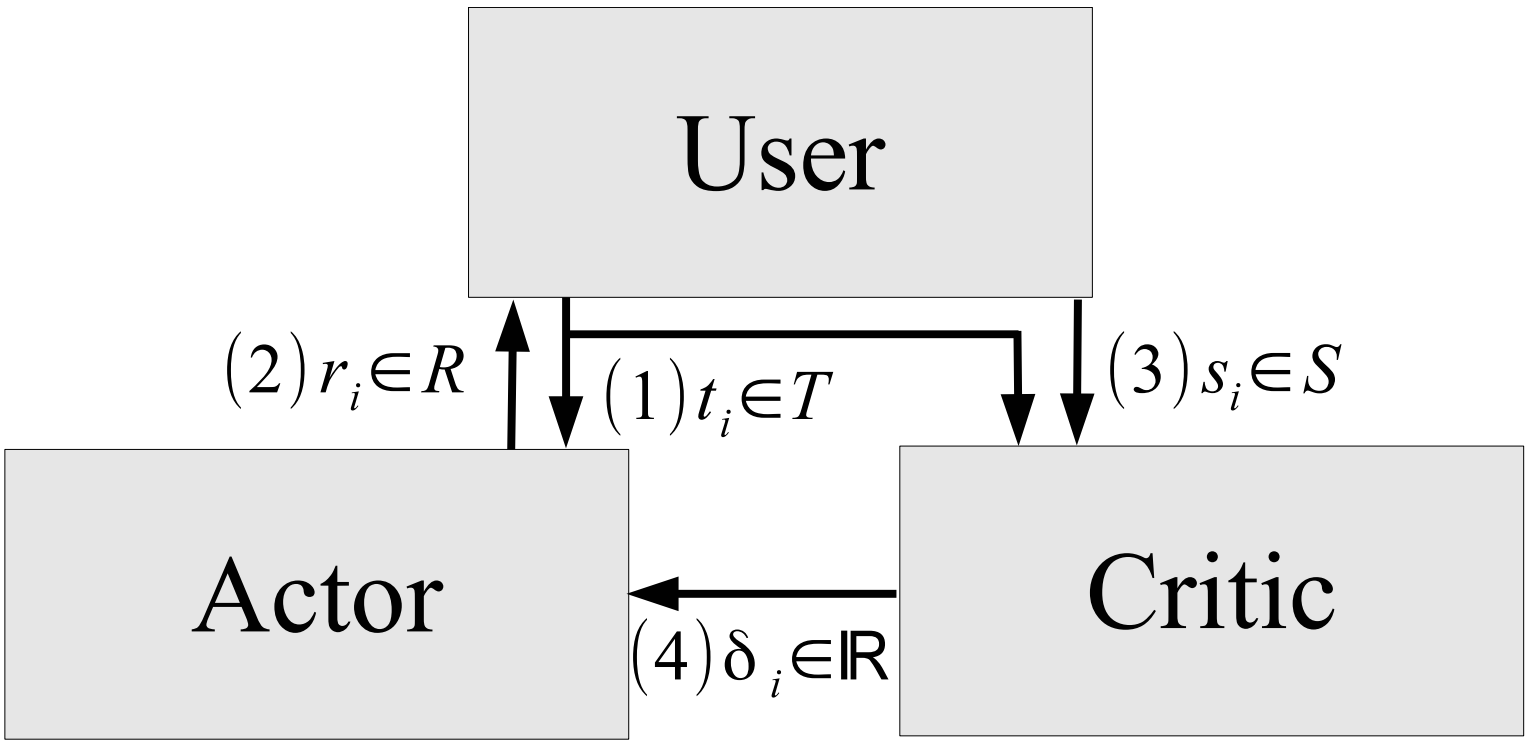


Figure: Online Actor-critic learning scheme.

Reinforcement learning

- Markov Decision Process (MDP) $\langle S, A, P, R \rangle$
  - S is the state space defined by state variables $X = \{X_1, X_2 \dots X_n\}$,
  - A is the action space,
  - P is the state transition function $P: S \times A \times S \to [0, 1]$, and
  - R is the reward function $R: S \times A \times S \to \mathbb{R}$.
- The learning agent looks for a policy $\pi_a(s)$ maximizing the expected accumulated reward, denoted $R^{\pi}(s)$.
  - The state-action value function $Q^{\pi}(s, a)$
  - The optimal action-state value function $Q^{*}(s, a)$

Reinforcement learning

Continuous Action-Critic Learning Automaton (CACLA)

- The actor only updates its policy if the critic is positive:

$$if\ \delta_t > 0 : \theta_t^a(s) \leftarrow \theta_t^a(s) + \alpha_t \cdot (a_t - \pi_a(s)) \cdot \frac{\partial \pi_a(s_{t-1})}{\partial \theta_{t-1}^{\pi}}. \quad (1)$$

- The critic is given by a $TD(\lambda)$ value iteration algorithm: The value function $V^{\pi}(t)$ is represented as Gaussian RBFs with 6 features per dimension. The update rule is defined:

$$\theta^V \leftarrow \theta^V + \alpha \left(s - \hat{V}(t) \right) \cdot \frac{\partial \hat{V}(t)}{\partial \theta^V}, \quad (2)$$

where α is the learning gain, s is the satisfaction value observed and $\hat{V}(t)$ is the estimated value of the actor's policy for task t.

Contents

Parameter definitions

Washing machines:

- The task ($T \in \mathbb{R}^{12}$)
  - Material percentages of the load: $C1$ (synthetic), $C2$ (silk), $C3$ (bedding), $C4$ (cotton), $C5$ (wool).
  - Degree of dirtiness of the load: $C6$ (less), $C7$ (normal), $C8$ (high), $C9$ (very stained).
  - Colors: $C10$ (white), $C11$ (little colors), $C12$ (very colored).
- The recipe ($R \in \mathbb{R}^5$): water in Liters, Temperature, (RPM) while drying, Detergent ml, duration in Minutes
- The satisfaction $S \in [0, 5]$,
  - distance of a given task-recipe pair to one of the 6 hidden *optimal tasks-recipes* ($\langle T_i^*, R_i^*, 5.0 \rangle$) unknown to the learning system.
  - The smaller the distance, the higher the satisfaction value (*reward* in RL) it is given.

Setup

1. The actor was presented one of the tasks for which an optimal recipe has been defined.
2. The actor outputs its recipe
3. The system simulates the satisfaction of the user as a function of the distance to the *optimal recipe*
4. The critic observes the reward, calculates the TD-error,
5. Observed this TD-error, the actor updates its policy
6. The actor reduces the amplitude of the additive noise signal

Contents

Some results

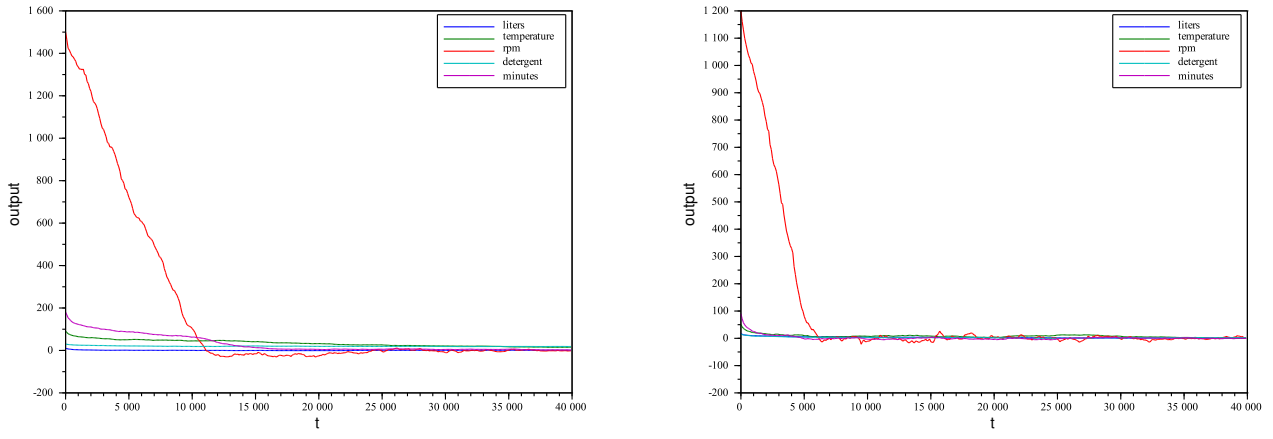


Figure: Actor: Outputs of the actor during the learning process for the original task $T_1^*$and T_2^*

Some results

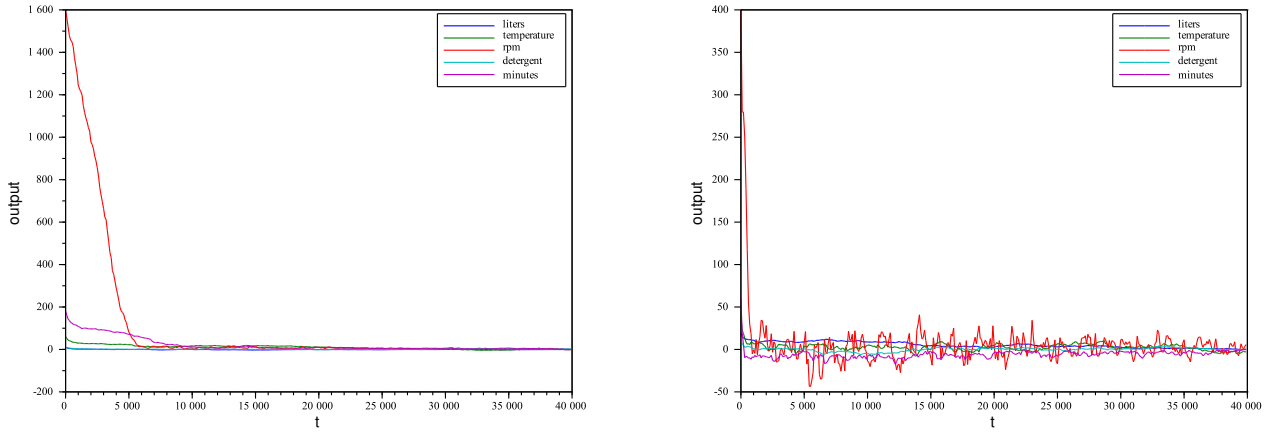


Figure: Actor: Outputs of the actor during the learning process for the original tasks T_3^* and T_4^*.

Contents

Conclusions

- computational experimental setup: washing machines
- We define 6 hidden prototype task-recipe pairs with maximum satisfaction from the user
- The reward is defined as the distance from the ideal recipe,
- so the aim of the RL algorithm is to reach zero
- The computational results are encouraging. The RL effectively converges to the hidden optimal recipes and maximum eahouker satisfaction. .

Acknowledgments

Grant agreement 317947 EU, SandS project. UFI11/07 of the UPV/EHU.