# An Instance of Social Intelligence in the Internet of Things: Bread Making Recipe Recommendation by ELM Regression

Manuel Graña and J. David Nuñez-Gonzalez(✉)

Computational Intelligence Group, UPV/EHU, Leioa, Spain
manuel.grana@ehu.eus,jdnunez001@gmail.com

**Abstract.** The Social and Smart project proposes a new framework for the interaction between users and their household appliances, where social interaction becomes an intelligent social network of users and appliances which is able to provide intelligent responses to the needs of the users. In this paper we focus on one incrasingly common appliance in the european homes: the bread-maker. There are a number of satisfaction parameters which can be specified by the user: crustiness, fragance, baking finish, and softness. A bread making recipe is composed mainly of the temperatures and times for each of the baking stages: first leavening, second leavening, precooking, cooking and browning. Although a thoroughful real life experimentation and data collection is being carried out by project partners, there are no data available for training/testing yet. Thus, in order to test out ideas we must resort to synthetic data generated using a very abstract model of the satisfaction parameters resulting from a given recipe. The recommendation in this context is carried by a couple of Extreme Learning Machine (ELM) regression models trained to predict the satisfaction parameters from the recipe input, and the other the inverse mapping from the desired satisfaction to the breadmaker appliance recipe. The inverse map allows to provide recommendations to the user given its preferences, while the direct map allows to evaluate a recipe predicting user satisfaction.

## 1 Introduction

There is an emerging view of social networks as information and knowledge repository at the service of the social agents to solve specific problems or to learn procedures relative to a shared domain of problems. Besides popular web service implementations, social networks have shown to be useful to spread educational innovations.

*Social computing* [1] may be defined as the result of social interaction when it is oriented towards information processing or decision making. Preliminary elaborations towards a taxonomy of social computing systems [2] include the term *subconscious intelligent social computing* [2–5] characterized by some hidden layer of intelligent processes that helps to produce innovative solutions to the problems posed by the social players. The social player asks for the solution

E. Onieva et al. (Eds.): HAIS 2015, LNAI 9121, pp. 16–25, 2015.
DOI: 10.1007/978-3-319-19644-2_2

of a problem, i.e. how to wash my laundry composed of items with some specific dirtiness and according to my preferences? The social framework provides solutions either from previous reported experiences of other social players or as innovation generated by the hidden intelligent layer.

*Intuitive description of the system.* In the framework of the Social and Smart (SandS) project[1] users are called eahoukers [6]. There are two repositories of knowledge in the SandS Social Network containing tasks to be carried on the appliances and the recipes solving them. When a user requires a task to be performed (blue dashed arrows) there are two possible situations, either the recipe solving the task is known or not. In the second case, the so called Networked Intelligence incorporating the hidden intelligent layer is in charge to produce a new recipe to solve the unknown task. In other words, it is in charge of achieving innovation (green arrow). The recipe found either way is returned to the appliance (black arrows). In the specific case of the breadmaker appliance, we do not have a proper task specification, because it is always the same. In some way it can be said that the specification of the desired satisfaction parameters, i.e. baking, crustines, softness, fragance, are the task specification. So, real life experiments give us pairs of (recipe, satisfaction) vector values, which are always the result of setting the breadmaker parameters and measuring the resulting bread. There is no way in real life to produce the data in the inverse way, setting the user satisfaction to see what is the resulting recipe. Therefore, this inverse map must be estimated from the data gathered in the direct experiments.

The SandS system simplified description introduces the fundamental questions that we are tackling in this paper by designing a prototype recommender system for a specific appliance, the breadmaker, and its validation.

– The first question is: how to build a recipe recommendation from the specification of the user satisfaction? That problem is addressed by building an Extreme Learning Machine[2] [7,8] from the experimental data that implements the inverse mapping.
– The second question is: how to decide that we need innovation? In other words, the inverse model may produce a recipe which in fact is far from solving the problem, so we need to create some new recipe outside the knowledge embedded in the mappings. How we detect that situation? The answer lies in the application of the direct mapping from recipes to satisfaction, and measuring the distance between the predicted satisfaction vector and the one specified by the user.
– The third question is: how to perform innovation? We need to build some generative process that achieves to create new recipes optimizing expected satisfaction. The solution proposed in [9] is a stochastic search process guided by the learned user satisfaction model, specifically an Evolutionary Strategy approach [17].

[1] http://www.sands-project.eu/.
[2] Source-code: http://www.ntu.edu.sg/home/egbhuang/elm_codes.html.

A critical issue is the lack of real life data supporting the design and validation of this architecture. The SandS project is currently building the framework that would allow users to experience this social interaction, but no actual data is being generated yet. So we have to resort to synthetic data.

*Paper contribution.* The contribution of this paper is a recommendation system for breadmaker recipes based on the know information about user satisfaction with past recipes tried, in the context of a social intelligence for appliance management. The system is composed of direct and inverse mappings between recipes and satisfaction evaluations, so it may produce a recipe recommendation from the specification of desired satisfaction parameter values given by the user. The direct mapping may be used for the satisfaction prediction on the recommended recipe, which may be used to decide if a random search innovation mechanism is required to produce a better suited recipe.

*Contents of the paper.* The paper is organized as follows: Sect. 2 reviews some ideas about recommender systems. Section 3 provides the precise specification of the problem and the description of the dataset synthesis, which has been used for the computational experiments. Section 4 reports results obtained on the synthetic dataset of recipes and satisfaction. Section 5 gives some conclusions of this paper.

## 2 Recommender Systems

Recommender systems [10] are taking a prominent role in the interaction with the virtual world incorporated by the miriad of webservices used on a daily basis by the common people. Early realizations included forms of collaborative filtering, however the advent of the Internet of Things will allow to use implicit, local and personal information gathered by the surrounding environment of smart objects. Recommender Systems are currently being applied in many different domains. Some example applications are: intelligent tourism [11], movie suggestions [12], electronic marketplaces [13], and university library research [14].

The State of the Art techniques involved in recommender systems deal with the problem of accurate representation and management of the user profile, requiring computational tools from many fields of Artificial Intelligence, such as Multi-agent systems, advanced optimization techniques, clustering of the users data to detect communities, and advanced knowledge representation and reasoning for the management of uncertainty.

Collaborative filtering social recommender systems [15] use social network information as additional input for improved recommendation accuracy. They define two categories of CF-based social recommender systems: matrix factorization based approaches and neighborhood based approaches, providing a comparison among algorithms.

In this paper we are concerned with the use of regression models trained with Extreme Learning Machines (ELM) to recommend a recipe from a given satisfaction, and also in the other way, to predict a satisfaction from a given recipe. Recipes are modeled as a feature vector of values from 0 to 1. Satisfaction is composed by associated models.

## 3 Problem Definition and Dataset Synthesis

In this section we give the specification of the recommendation problem that we try to solve. As there is not real life data to validate our proposal, we have had to build some models to generate a synthetic dataset with some degree of arbitrary complexity, so that if our approach succeeds on this dataset, it can be successful in real life experiments. The experiment context is the "SandS" European project (http://www.sands-project.eu/). In this project, Eahoukers (word that refers to easy house workers, in other words, users) provide a description of a problem dealing with household appliance usage to the social network. The system gives back a "recipe" that solves the proposed problem. These recipes are either proposed by the knowledge provided by other users or by the underlying intelligent layer [16]. Once that recipe is proposed, user can give the order to the system to execute it in the choosen appliance. Finally, users give a satisfaction of the recipe, this feedback is used to tune the intelligent layer and/or to personalize the system.

### 3.1 Specification of the Recommendation Problem

This paper is focused on the case of the breadmaker. It has some specific features that differentiate the way recommendations are generated. First, there is no task description per se. The user only gives the order to make the bread stating some expected satisfaction values with the result which are not stated beforehand. In other words, we only have the recipe and satisfaction pairs. The recommendation system then has two problems to solve, first it must learn the map from recipes to satisfaction, in order to predict the user satisfaction. Second, it must learn the inverse model from satisfaction to recipes in order to propose the best recipe for the user. It is also possible, once we have this inverse model, to tune the recipe to specific values of the predicted satisfaction. It may even possible to work with missing values, that is, to provide a recipe that matches some of the satisfaction parameters, when the other are left undefined. We have not touched this aspect in this paper.

*Recipes.* The baking operation consists of 5 steps carried sequentially: first leavening, second leavening, precooking, cooking and browning. Each step is specified by a pair $[Time, Temperature]$. Thus, in this case, the recipe consists in 10 variables $[r_1, ..., r_{10}]$.

*Satisfaction.* The user give a satisfaction feedback. For the breadmaker, the satisfaction consists in 4 parameters: $[fragance, softness, baking, crust]$. These parameters are represented in 4 variables $[s_1, ..., s_4]$.

*Problem specification.* The problems that we want to solve with this experiment are two:

- Direct prediction: What will be the satisfaction feedback obtained from the user for a given recipe?
- Inverse recommendation: Which is the recipe that I need to get a specific satisfaction?

Let us define:

- Let be $R$ a recipe described by bread making variables, so $[r_1, ..., r_{10}] = R$ . Thus, $R \in \mathbb{R}^{10}$ and each $r_i$ is normalized in the range [0..1]
- Let be $S$ a satisfaction described by $[s_1, .., s_4] = S$ . Thus, $S \in \mathbb{R}^4$ and each $s_j$ is a number in the set $\{0, 1, 2, 3, 4, 5\}$

To answer these questions, we define:

- A direct mapping $\phi(R) = S$ to predict the satisfaction of the user with the quality of the bread resulting from a proposed recipe (first question) $\phi : \mathbb{R}^{10} \longrightarrow \mathbb{R}^4$
- The inverse mapping $\phi^{-1}(S) = R$ that looks for the recipe that would provide the desired satisfaction parameter values (second question) $\phi^{-1} : \mathbb{R}^4 \longrightarrow \mathbb{R}^{10}$

We model the experiment with the numbers and parameters defined before but numbers and set of variables could be adapted to any other context of similar experimentation. These mappings are built by ELM because of the quick learning time which allows frequent updates when the experience of the users increase the database for learning. Notice that we only have information about experiments going in the direct prediction sense, i.e. we can try a recipe and ask the user its satisfaction. It is not possible to obtain experimental data in the other direction.

The first learning experiment is to calculate the regression of satisfaction values from given recipes. We denote this experiment as $\phi(R) \rightarrow S_j$. The second experiment is to calculate the regression o f recipe valuesfrom a given satisfaction, i.e. to create the recommendation. We denote this experiment as $\phi^{-1}(S) \rightarrow R_i$. We divided the dataset in several datasets according to the application requirements of the 10-fold cross-validation technique. Figure 1 is the pipeline which summarizes the process of the experiment.

## 3.2 Dataset Generation

We generate a dataset of 100,000 instances of recipe satisfaction pairs taking into consideration the following:

- We consider that there is a non-linear map that models the contribution from each recipe parameter to each satisfaction parameter value. For the experimental works in this paper, we have created arbitrary maps which are shown in Fig. 2. The idea is that if we can approximate these models with ELMs then we can approximate almost anything. Each entry $(i, j)$ in the table is a map randomly generated relating recipe parameter $r_i$ with each satisfaction parameter $s_j$. Thus, we have 40 models.

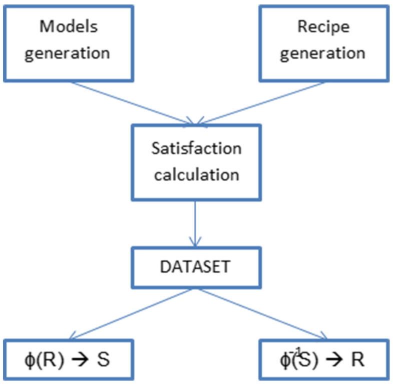

**Fig. 1.** Pipeline of experiment

– We consider that the satisfaction value is a linear combination of the contributions of the recipe parameters. If $a_{ij}$ is the satisfaction value in variable $s_j$ induce from a given value from recipe variable $r_i$, then:

$$s_j = \frac{\sum_{r=1}^{10} \alpha_i * a_{ij}}{r'}$$

using $\alpha_i$ as weighting factor of recipe variable and being $r'$ the normalizing value to obtain the weighted average. If result has decimal part, we round the number to the nearest natural one. We have choosen the value of the $\alpha_i$ arbitrarily for the experiments reported here.

Once we have models, we are able to generate a synthetic dataset according with models. To generate a database we generate randomly 100,000 instances of $[r_1, ..., r_{10}]$ then, we use the models to calculate the satisfaction.

As example, we show in Table 1 the first row of the dataset.

**Table 1.** Example of the content of Dataset

| | r1 | r2 | r3 | r4 | r5 | r6 | r7 | r8 | r9 | r10 | s1 | s2 | s3 | s4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| #1 | 0.6 | 0.3 | 0.4 | 0.2 | 1 | 0 | 0.4 | 0.8 | 0.3 | 0.4 | 2 | 2 | 2 | 2 |
| #i | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |

## 4 Experimental Results

Experiments are carried out using ELM standar code in Matlab[3]. We select Sine ('sin') activation function for executions. We test results with 1 hidden unit until

[3] Source-code: http://www.ntu.edu.sg/home/egbhuang/elm_codes.html.

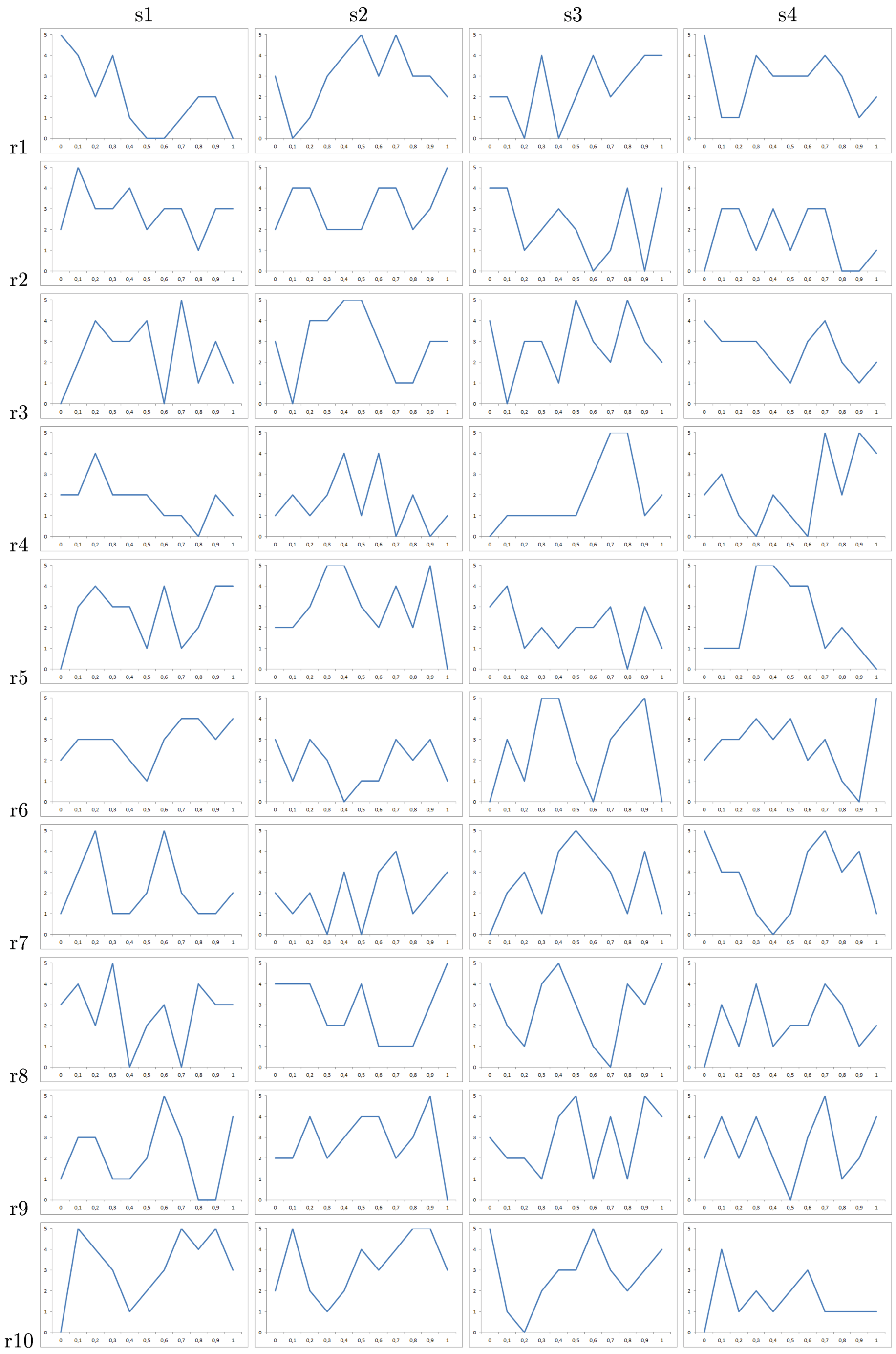

**Fig. 2.** Maps specifying the influence of the recipe parameters into the satisfaction parameters. Each entry relates a pair of recipe-satisfaction variables. Each plot has horizontal axis in the range $[0, 1]$ and the vertical axis in the set $\{0, 1, 2, 3, 4, 5\}$.

525 hidden units that are the maximum hidden units allowed without raising a memory exception. Increasing neurons, square error decreases significantly. Table 2 shows the average regression error for the direct mapping regression for each satisfaction parameter obtained in a 10-fold cross-validation experiment for the two extreme ELM sizes. The best result is equivalent to a relative error, computed dividing the regression error by the variable range which is 5 for all satisfaction values, is below 0.01. Table 3 shows the average regression error for the inverse mapping regression for each recipe parameter obtained in a 10-fold cross-validation experiment for the two extreme ELM sizes. The best relative error, computed dividing the regression error by the variable range which is 1for all satisfaction values, is below 0.2, still too high for the practical purposes of this paper.

**Table 2.** Average cross-validation error results of satisfaction prediction for given recipes: $\phi(R) \rightarrow S_j$

| | 1 hidden unit | 525 hidden units |
|---|---|---|
| s1 | 1.4490 | 0.4972 |
| s2 | 1.7756 | 0.4790 |
| s3 | 1.6259 | 0.5639 |
| s4 | 1.1084 | 0.4832 |

**Table 3.** Average cross-validation error results of recipe recommendation for desired satisfactions: $\phi^{-1}(S) \rightarrow R_i$

| | 1 hidden unit | 525 hidden units |
|---|---|---|
| r1 | 0.4382 | 0.2816 |
| r2 | 0.3910 | 0.2887 |
| r3 | 0.4298 | 0.2919 |
| r4 | 0.4080 | 0.2659 |
| r5 | 0.4433 | 0.2923 |
| r6 | 0.4063 | 0.2837 |
| r7 | 0.3936 | 0.2903 |
| r8 | 0.4743 | 0.2885 |
| r9 | 0.4308 | 0.2911 |
| r10 | 0.4456 | 0.2688 |

## 5 Conclusions

We propose the application of regression ELM to build a breadmaker recommender system which is an instance of the social intelligence in the Internet of

Things framework of the SandS european project. We have proposed a dataset synthesis procedure to carry the experimental validation of the system, due to the lack of real-life data. The experimental results are quite good for the direct mapping from recipes to satisfaction evaluations, but not so good for the inverse mapping, which will require a more careful tuning for the practical application. Further work will be addressing the computational experiments on real life data, once it is available from the breadmaking experiments being carried out by other project partners. It is also possible to open the experimentation to the general public by the implementation of a social network of breadmaking “aficionados”. This implementation would be a real test of the idea of subconscious social intelligence, which in this setting will encompass the application of both direct an inverse mappings.

**Acknowledgements.** This research has been partially funded by EU through SandS project, grant agreement no 317947. The GIC has been supported by grant IT874-13 as university research group category A.

## References

1. Vannoy, S.A., Palvia, P.: The social influence model of technology adoption. Commun. ACM **53**(6), 149–153 (2010)
2. Graña, M., Marqués, I., Savio, A., Apolloni, B.: A domestic application of intelligent social computing: the sandsproject. In: Herrero, Á., et al. (eds.) International Joint Conference SOCO’13-CISIS’13-ICEUTE’13. AISC, vol. 239, pp. 221–228. Springer, Heidelberg (2013)
3. Graña, M.: Subconscious social computational intelligence. In: Krishnan, G.S.S., Anitha, R., Lekshmi, R.S., Kumar, M.S., Bonato, A., Graña, M. (eds.) Computational Intelligence, Cyber Security and Computational Models, Proceedings of ICC3. Advances in Intelligent Systems and Computing, vol. 246, pp. 15–21. Springer, India (2013)
4. Graña, M., et al.: Social and smart: towards an instance of subconscious social intelligence. In: Iliadis, L., Papadopoulos, H., Jayne, C. (eds.) EANN 2013, Part II. CCIS, vol. 384, pp. 302–311. Springer, Heidelberg (2013)
5. Grana, M., Rebollo, I.: Instances of subconscious social intelligent computing. In: 2013 Fifth International Conference on Computational Aspects of Social Networks (CASoN), pp. 74–78, August 2013
6. Apolloni, B., Fiasche, M., Galliani, G., Zizzo, C., Caridakis, G., Siolas, G., Kollias, S., Grana Romay, M., Barriento, F., San Jose, S.: Social things - the sands instantiation. In: IoT-SoS 2013. IEEE (2013)
7. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: theory and applications. Neurocomputing **70**, 489–501 (2006)
8. Huang, G., Zhu, Q., Siew, C.: Extreme learning machine: a new learning scheme of feedforward neural networks. In: IEEE International Conference on Neural Networks - Conference Proceedings, vol. 2, pp. 985–990 (2004). Cited By (since 1996):113
9. Marques, I., Graña, M., Kamińska-Chuchmała, A., Apolloni, B.: An experiment of subconscious intelligent social computing on household appliances. Neurocomputing (2014, in press)

10. Bobadilla, J., Ortega, F., Hernando, A., GutiÃrrez, A.: Recommender systems survey. Knowl.-Based Syst. **46**, 109–132 (2013)
11. Borras, J., Moreno, A., Valls, A.: Intelligent tourism recommender systems: a survey. Expert Syst. Appl. **41**(16), 7370–7389 (2014)
12. Briguez, C.E., Budan, M.C., Deagustini, C.A., Maguitman, A.G., Capobianco, M., Simari, G.R.: Argument-based mixed recommenders and their application to movie suggestion. Expert Syst. Appl. **41**(14), 6467–6482 (2014)
13. Christidis, K., Mentzas, G.: A topic-based recommender system for electronic marketplace platforms. Expert Syst. Appl. **40**(11), 4370–4379 (2013)
14. Tejeda-Lorente, A., Porcel, C., Peis, E., Sanz, R., Herrera-Viedma, E.: A quality based recommender system to disseminate information in a university digital library. Inf. Sci. **261**, 52–69 (2014)
15. Yang, X., Guo, Y., Liu, Y., Steck, H.: A survey of collaborative filtering based social recommender systems. Comput. Commun. **41**, 1–10 (2014)
16. Graña, M., Nuñez-Gonzalez, J.D., Apolloni, B.: A discussion on trust requirements for a social network of eahoukers. In: Pan, J.-S., Polycarpou, M.M., Woźniak, M., de Carvalho, A.C.P.L.F., Quintián, H., Corchado, E. (eds.) HAIS 2013. LNCS, vol. 8073, pp. 540–547. Springer, Heidelberg (2013)
17. Gonzalez, A.I., Graña, M., Ruiz Cabello, J., D'Anjou, A., Albizuri, F.X.: Experimental results of an evolution-based adaptation strategy for VQ image filtering. Inf. Sci. **133**(3–4), 249–266 (2001). http://dx.doi.org/10.1016/S0020-0255(01)00088-3